

**PERANCANGAN SISTEM ANTRIAN PASIEN POLIKLINIK
BERBASIS SMS GATEWAY**

SKRIPSI

Disusun Oleh :
Rizqi Budhi Prastiyo
1118902



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK INFORMATIKA S-1
2013

LEMBAR PERSETUJUAN

**PERANCANGAN SISTEM ANTRIAN PASIEN POLIKLINIK
BERBASIS SMS GATEWAY**

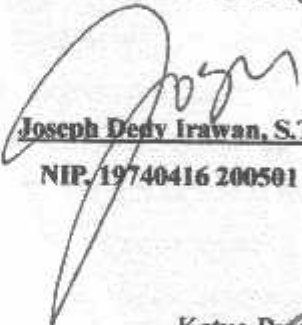
SKRIPSI

Disusun dan diajukan untuk melengkapi dan memenuhi persyaratan guna mencapai gelar
Sarjana Komputer

Disusun oleh :
Rizqi Budhi Prastiyo
11.18.902

Diperiksa dan Disetujui,

Dosen Pembimbing I


Joseph Dedy Irawan, S.T., M.T.
NIP. 19740416 200501 1 002

Dosen Pembimbing II


Suryo Adi Wibowo, S.T.
NIP.P. 1031000438

Mengetahui,

Ketua Program Studi Teknik Informatika S-1


Joseph Dedy Irawan, S.T., M.T.
NIP. 19740416 200501 1 002

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

KATA PENGANTAR

Dengan mengucap Alhamdulillah atas berkat rahmat Allah yang Maha pengasih dan pemurah, maka penulis berhasil menyelesaikan Skripsi yang berjudul “Perancangan Sistem Antrian Pasien Poliklinik Berbasis Sms Gateway”.

Skripsi ini merupakan salah satu syarat yang harus ditempuh untuk meraih gelar Sarjana di Jurusan Teknik Informatika, Fakultas Teknologi Industri, Institut Teknologi Nasional Malang.

Penulis menyadari bahwa penulisan Skripsi ini tidak akan tersusun dengan baik tanpa adanya pengarahan, bimbingan dan bantuan dari berbagai pihak. Untuk itu penulis mengucapkan banyak terima kasih kepada :

1. Bapak Ir. Suparno Djiwo, MT. selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Anang Subardi, MT. selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Joseph Dedy Irawan, S.T., M.T., selaku Pembimbing I, yang telah banyak memberi masukan, saran, dan bimbingan selama penulisan Skripsi ini.
4. Bapak Suryo Adi Wibowo, S.T., selaku Pembimbing II yang telah banyak memberi masukan, saran, dan bimbingan selama penulisan Skripsi ini.
5. Seluruh Dosen dan Laboran Teknik Informatika Institut Teknologi Nasional Malang, yang banyak memberikan masukan serta saran selama perkuliahan dan praktikum.
6. Bapak, Ibu tercinta, yang memberikan dukungan, do'a restunya serta pengorbanan yang tulus demi tercapainya cita-cita anaknya.
7. Rekan-rekan seperjuangan Teknik Informatika Institut Teknologi Nasional Malang dan semua pihak yang banyak membantu dalam penyelesaian Skripsi ini.

Penulis menyadari sepenuhnya bahwa Skripsi ini masih banyak kekurangan, oleh karena itu penulis sangat menghargai dengan sepenuh hati dan mengucapkan terima kasih jika ada saran dan kritik yang membangun guna menyempurnakan dalam penulisan laporan Skripsi.

Malang, April 2013

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
ABSTRAK	iii
KATA PENGANTAR	iv
DAFTAR ISI	v
DAFTAR GAMBAR	viii
DAFTAR TABEL	x
DAFTAR LAMPIRAN	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Perancangan.....	2
1.4 Manfaat Tugas Akhir.....	2
1.5 Ruang Lingkup dan Batasan Masalah.....	2
1.6 Metodologi Perancangan.....	2
1.7 Sistematika Penulisan.....	3
BAB II LANDASAN TEORI	4
2.1 Antrian.....	4
2.1.1 Sistem Antrian.....	5
2.1.2 Struktur Antrian.....	6
2.2 Borland Delphi 7.....	7
2.2.1 Sejarah Borland Delphi 7.0.....	8
2.2.2 Kelebihan Borland Delphi 7.....	8
2.2.3 Interbase.....	9
2.3 SMS (Short Message Service).....	10
2.4 SMS Gateway.....	11
2.5 MySQL.....	12
2.5.1 Keistimewaan MySQL.....	12
2.5.2 Perkembangan MySQL.....	14
2.5.3 Perintah Dasar MySQL.....	14
2.6 Gammu.....	15
2.7 Handphone.....	16
BAB III PERANCANGAN	17
3.1 Rancangan Proyek Akhir.....	17
3.1.1 Desain Sistem.....	17
3.1.2 Data Flow Diagram (DFD).....	17
3.1.2.1 Data Flow Diagram (DFD) Level 0.....	18
3.1.2.2 Data Flow Diagram (DFD) Level 1.....	18
3.1.2.3 Data Flow Diagram (DFD) Level 2 Pendataan Pasien.....	20
3.1.2.4 Data Flow Diagram (DFD) Level 2 Pendataan Pegawai.....	20
3.1.2.5 Data Flow Diagram (DFD) Level 2 Pendataan Dokter.....	21

3.1.2.6	Data Flow Diagram (DFD) Level 2 Pendataan Poliklinik...	21
3.1.2.7	Data Flow Diagram (DFD) Level 2 Pendataan Admin	22
3.1.2.8	Data Flow Diagram (DFD) Level 2 Pendataan Praktek	23
3.1.2.9	Data Flow Diagram (DFD) Level 2 Pendataan Rekam Medis	23
3.1.2.10	Data Flow Diagram (DFD) Level 2 Pendataan jadwal janji	24
3.1.3	Entity Relationship Diagram (ERD).....	24
3.1.4	Physical Data Model.....	26
3.1.4.1	Tabel Admin.....	26
3.1.4.2	Tabel Dokter.....	26
3.1.4.3	Tabel Praktek.....	26
3.1.4.4	Tabel Pasien.....	27
3.1.4.5	Tabel Pegawai.....	27
3.1.4.6	Tabel Rekam Medis.....	27
3.1.4.1	Tabel Pendaftaran	28
3.1.4.1	Tabel Poliklinik	28
3.1.5	Diagram Alir Program	28
3.1.5.1	Diagram Alir Menu Utama.....	28
3.1.5.2	Diagram Alir Login	30
3.1.5.3	Diagram Alir Tambah Data Pasien.....	30
3.1.5.4	Diagram Alir Tambah Data Pegawai.....	31
3.1.5.5	Diagram Alir Tambah Data Dokter	32
3.1.5.6	Diagram Alir Tambah Data Poliklinik.....	33
3.1.5.7	Diagram Alir Tambah Data Admin	34
3.1.5.8	Diagram Alir Tambah Data Praktek	35
3.1.5.9	Diagram Alir Tambah Data Rekam Medis	36
3.1.5.10	Diagram Alir User	37
3.1.6	Desain Rancangan	38
3.1.6.1	Struktur Menu	38
3.1.6.2	Rancangan Desain Menu Utama	39
3.1.6.3	Rancangan Desain Menu Login	40
3.1.6.4	Rancangan Desain Menu File.....	40
3.1.6.5	Rancangan Desain Menu Change Password.....	41
3.1.6.6	Rancangan Desain Menu Data	41
3.1.6.7	Rancangan Desain Menu Data Pasien	42
3.1.6.8	Rancangan Desain Menu Data Pegawai	42
3.1.6.9	Rancangan Desain Menu Data Dokter.....	43
3.1.6.10	Rancangan Desain Menu Data Poliklinik	43
3.1.6.11	Rancangan Desain Menu Data Admin.....	44
3.1.6.12	Rancangan Desain Menu Data Praktek.....	44
3.1.6.13	Rancangan Desain Menu Data Rekam Medis.....	45
3.1.6.14	Rancangan Desain Menu Data Antrian Pasien.....	45
3.1.6.15	Rancangan Desain Menu Setting Status Poliklinik	46
BAB IV	IMPLEMENTASI DAN PENGUJIAN	47
4.1	Hasil Implementasi Database.....	47
4.1.1	Implementasi Tabel Admin.....	47
4.1.2	Implementasi Tabel Dokter.....	47
4.1.3	Implementasi Tabel Pasien.....	48
4.1.4	Implementasi Tabel Pegawai.....	48
4.1.5	Implementasi Tabel Pendaftaran	48

4.1.6 Implementasi Tabel Poliklinik	49
4.1.7 Implementasi Tabel Praktek.....	49
4.1.8 Implementasi Tabel Rekam medis.....	49
4.2 Hasil Implementasi Sistem Keseluruhan	49
4.2.1 Hasil Form Utama.....	50
4.2.2 Hasil Form Login.....	50
4.2.3 Hasil Form Change Password	51
4.2.4 Hasil Form Data Pasien.....	51
4.2.5 Hasil Form Add Data Pasien.....	52
4.2.6 Hasil Form Update Data Pasien.....	52
4.2.7 Hasil Form Data Pegawai.....	53
4.2.8 Hasil Form Add Data Pegawai.....	53
4.2.9 Hasil Form Update Data Pegawai.....	54
4.2.10 Hasil Form Data Dokter	54
4.2.11 Hasil Form Add Data Dokter	55
4.2.12 Hasil Form Update Data dokter.....	55
4.2.13 Hasil Form Data Poliklinik	56
4.2.14 Hasil Form Add Data Poliklinik.....	56
4.2.15 Hasil Form Update Data Poliklinik	57
4.2.16 Hasil Form Data Admin.....	57
4.2.17 Hasil Form Add Data Admin	58
4.2.18 Hasil Form Data Praktek.....	58
4.2.19 Hasil Form Add Data Praktek	59
4.2.20 Hasil Form Rekam Data Medis	59
4.2.21 Hasil Form Add Data Rekam Data Medis Pasien	60
4.2.22 Hasil Form Update Data Rekam Data Medis.....	60
4.2.23 Hasil Form Data Antrian Pasien.....	61
4.2.24 Hasil Form Add Pendaftaran Pasien.....	61
4.2.25 Hasil Form Setting Status Poliklinik	62
4.2.26 Hasil Laporan Data Pegawai	62
4.2.27 Hasil Laporan Data Dokter.....	63
4.2.28 Hasil Laporan Data Pasien	63
4.3 Pengujian Sistem	64
BAB V PENUTUP	65
5.1 Kesimpulan	65
5.2 Saran	65
DAFTAR PUSTAKA	66
LAMPIRAN	67

DAFTAR GAMBAR

Gambar	Halaman
2.1	Single Channel – Single Phase..... 6
2.2	Single Channel – Multi Phase..... 6
2.3	Multi Channel – Single Phase..... 7
2.4	Multi Channel – Multi Phase..... 7
2.5	Cara Kerja SMS Gateway..... 12
3.1	Desain Sistem..... 17
3.2	<i>Data Flow Diagram</i> (DFD) Level 0 Perancangan Sistem Antrian Pasien Poliklinik Berbasis <i>SMS Gateway</i> 18
3.3	<i>Data Flow Diagram</i> (DFD) Level 1..... 19
3.4	<i>Data Flow Diagram</i> (DFD) Level 2 Pendataan Pasien..... 20
3.5	<i>Data Flow Diagram</i> (DFD) Level 2 Pendataan Pegawai..... 20
3.6	<i>Data Flow Diagram</i> (DFD) Level 2 Pendataan Dokter..... 21
3.7	<i>Data Flow Diagram</i> (DFD) Level 2 Pendataan Poliklinik..... 22
3.8	<i>Data Flow Diagram</i> (DFD) Level 2 Pendataan Admin..... 22
3.9	<i>Data Flow Diagram</i> (DFD) Level 2 Pendataan Praktek..... 23
3.10	<i>Data Flow Diagram</i> (DFD) Level 2 Pendataan Rekam Medis..... 23
3.11	<i>Data Flow Diagram</i> (DFD) Level 2 Jadwal Janji..... 24
3.12	<i>Entity Relationship Diagram</i> (ERD)..... 25
3.13	Diagram Alir Menu Utama..... 29
3.14	Diagram Alir <i>Login</i> 30
3.15	Diagram Alir Tambah Data Pasien..... 31
3.16	Diagram Alir Tambah Data Pegawai..... 32
3.17	Diagram Alir Tambah Data Dokter..... 33
3.18	Diagram Alir Tambah Data Poliklinik..... 34
3.19	Diagram Alir Tambah Data Admin..... 35
3.20	Diagram Alir Tambah Data Praktek..... 36
3.21	Diagram Alir Tambah Rekam Medis..... 37
3.22	Diagram Alir User 38
3.23	Struktur Menu 39
3.24	Rancangan Desain Menu Utama..... 40
3.23	Rancangan Desain Menu <i>Login</i> 40
3.24	Rancangan Desain Menu <i>File</i> 41
3.25	Rancangan Desain Menu <i>Change Password</i> 41
3.26	Rancangan Desain Menu Data..... 42
3.27	Rancangan Desain Menu Data Pasien..... 42
3.28	Rancangan Desain Menu Data Pegawai..... 43
3.29	Rancangan Desain Menu Data Dokter..... 43
3.30	Rancangan Desain Menu Data Poliklinik..... 44
3.31	Rancangan Desain Menu Data Admin..... 44
3.32	Rancangan Desain Menu Data Praktek..... 45
3.33	Rancangan Desain Menu Rekam Medis..... 45
3.34	Rancangan Desain Menu Data Antrian Pasien..... 46
3.35	Rancangan Desain Menu Setting Status Poliklinik..... 46
4.1	Data Pengujian Tabel Admin..... 47
4.2	Data Pengujian Tabel Dokter..... 47
4.3	Data Pengujian Tabel Pasien..... 48

4.4	Data Pengujian Tabel Pegawai.....	48
4.5	Data Pengujian Tabel Pendaftaran.....	48
4.6	Data Pengujian Tabel Poliklinik.....	49
4.7	Data Pengujian Tabel Praktek.....	49
4.8	Data Pengujian Tabel Rekam Medis.....	49
4.9	<i>Form</i> Utama.....	50
4.10	<i>Form Login</i>	50
4.11	<i>Form Change Password</i>	51
4.12	Form Data Pasien.....	51
4.13	<i>Form Add</i> Data Pasien.....	52
4.14	<i>Form Update</i> Data Pasien.....	52
4.15	<i>Form</i> Data Pegawai.....	53
4.16	<i>Form Add</i> Data Pegawai.....	53
4.17	<i>Form Update</i> Data Pegawai.....	54
4.18	Form Data Dokter.....	54
4.19	<i>Form Add</i> Data Dokter.....	55
4.20	<i>Form Update</i> Data Dokter.....	55
4.21	<i>Form</i> Data Poliklinik.....	56
4.22	<i>Form Add</i> Data Poliklinik.....	56
4.23	Form Update Data Poliklinik.....	57
4.24	Form Data Admin.....	57
4.25	<i>Form Add</i> Data Admin.....	58
4.26	<i>Form</i> Data Praktek.....	58
4.27	<i>Form Add</i> Data Praktek.....	59
4.28	<i>Form</i> Rekam Data Medis.....	59
4.29	<i>Form Add</i> Data Rekam Medis Pasien.....	60
4.30	<i>Form Update</i> Data Rekam Medis.....	60
4.31	<i>Form</i> Data Antrian Pasien.....	61
4.32	<i>Form Add</i> Pendaftaran Pasien.....	61
4.33	<i>Form Setting</i> Status Poliklinik.....	62
4.34	Laporan Data Pegawai.....	62
4.35	Laporan Data Dokter.....	63
4.36	<i>Form</i> Laporan Data Pasien.....	63

DAFTAR TABEL

	Tabel	Hal
3.1	Tabel Admin	26
3.2	Tabel Dokter	26
3.3	Tabel Praktek	26
3.4	Tabel Pasien	27
3.5	Tabel Pegawai	27
3.6	Tabel Rekam Medis	27
3.7	Tabel Pendaftaran	28
3.8	Tabel Poliklinik	28
4.1	Tabel Pengujian Sistem	64

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Surat Pernyataan	A-1
Lampiran 2. Formulir Bimbingan Skripsi.....	B-2
Lampiran 3. Berita Acara Ujian Skripsi	C-3
Lampiran 4. Formulir Perbaikan Skripsi	D-4
Lampiran 5. Lampiran <i>Script</i>	E-5

BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi terutama di bidang informatika membawa perubahan besar dengan memberikan berbagai macam program yang dapat mempermudah segala macam aktifitas manusia. Penekanan pada faktor kemudahan operasional serta kepraktisan membuat peralihan teknologi dari sistem analog ke sistem digital.

SMS (Short Message Service) belakangan ini semakin marak keberadaannya dalam kehidupan sehari-hari. Hampir semua kalangan menggunakan *SMS* sebagai sarana komunikasi sehari-hari, karena harganya yang murah. Salah satu bagian yang tidak dapat dipisahkan dalam bisnis *SMS* ini adalah peranan sebuah *SMS Gateway*. Sebuah sistem yang dipergunakan oleh penyedia jasa untuk mengirim maupun menerima *SMS*. Banyak manfaat yang akan didapat dengan menggunakan *SMS Gateway* dibandingkan dengan menggunakan sistem *online* atau menggunakan internet, contohnya biaya yang akan dikeluarkan akan lebih murah dibandingkan dengan menggunakan internet, kecepatan *SMS* lebih cepat dari pada menggunakan internet, dapat diakses kapan saja dan di mana saja.

Pemanfaatan teknologi berbasis *SMS Gateway* dapat digunakan untuk meningkatkan pelayanan kesehatan yang murah. Di rumah sakit setiap hari dijumpai banyak pasien yang datang untuk memeriksakan kandungannya. Sering dijumpai banyak pasien yang sakit harus menunggu lama untuk mendapatkan giliran pemeriksaan dan tidak lepas juga permasalahan mengenai urutan pemeriksaan, sehingga untuk meningkatkan pelayanan kepada setiap pasien yang datang perlu dibuatkan sebuah sistem agar dapat membantu pihak pasien dalam mendapatkan informasi yang akurat maupun pelayanan yang terbaik dari pihak rumah sakit.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, maka muncul permasalahan bagaimana merancang dan membangun sistem antrian pasien berbasis *SMS Gateway* sehingga dapat mengatur nomor urut antrian pasien.

1.3 Tujuan

Tujuan yang hendak dicapai adalah merancang dan membangun sistem antrian pasien berbasis SMS *gateway*.

1.4 Batasan Masalah

Agar dalam pembuatan program ini tidak menyimpang dari maksud dan tujuan yang diharapkan, maka dibuat batasan-batasan. Batasan masalah dalam perancangan sistem ini adalah:

1. Aplikasi ini dibangun menggunakan bahasa pemrograman *Delphi*.
2. *Gammu* sebagai aplikasi *SMS Gateway*.
3. Pendaftaran pasien secara manual.
4. Estimasi waktu rata-rata 7 menit.

1.5 Manfaat

Manfaat perancangan sistem antrian pasien berbasis *SMS Gateway* adalah :

1. Pasien dapat memesan nomor antrian dengan *SMS* tanpa harus datang ke rumah sakit terlebih dahulu.
2. Pasien dapat mengecek jadwal praktik dokter langganannya.

1.6 Metodologi Pelaksanaan Program

Metodologi yang digunakan dalam pembuatan tugas akhir adalah sebagai berikut:

1. Studi Literatur

Pencarian literatur untuk menunjang perencanaan dan pembuatan sistem, meliputi bahasa pemrograman *Delphi 7* dan *Gammu*.

2. Perencanaan Program

Perencanaan program dimulai dengan merencanakan sistem yang ingin dibuat, mensketsa *Flowchart*, mencobanya pada *SMS Gateway*, dan melakukan analisa setiap kesalahan.

3. Perancangan Program

Perancangan sistem ini bertujuan untuk merancang konstruksi dari SMS Gateway dan menyesuaikan dengan *Gammu* sehingga dapat bekerja dengan baik.

1.7 Sistematika Penulisan Laporan

Berikut akan diuraikan mengenai sistematika atau alur pembahasan tulisan pada laporan akhir yang terbagi dalam bab-bab sebagai berikut:

BAB I : PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi penelitian dan sistematika penulisan laporan akhir.

BAB II : LANDASAN TEORI

Bab ini berisi teori-teori dasar yang relevan dengan masalah yang dibahas dalam laporan akhir, meliputi pengertian antrian, *Borland Delphi 7*, *SMS Getway*, *MySQL*, dan *Gammu*.

BAB III : PERANCANGAN

Bab ini berisi tentang perancangan program yang berupa analisa dan desain program aplikasi yang akan dibuat. Proses analisa menjelaskan diantaranya tentang deskripsi program ini, penjelasan program yang berjalan saat ini, pengguna program, dan spesifikasi kebutuhan program. Sedangkan pada tahap desain dibagi menjadi desain dasar, dan desain aplikasi secara keseluruhan.

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Berisi pengujian, analisa, dan Implementasi dari program, membandingkan dengan perencanaan awal serta teori-teori dari proyek akhir.

BAB V : PENUTUP

Berisi tentang kesimpulan dan saran.

BAB II

LANDASAN TEORI

Bab ini membahas konsep dasar dalam perancangan sistem antrian pasien poliklinik berbasis *SMS Gateway*. Pembahasan bab ini meliputi : Pengertian Antrian, Bahasa Pemrograman *Borland Delphi 7*, *SMS Gateway*, *MySQL*, *Gammu* dan *handphone*.

2.1 Antrian

Antrian adalah suatu garis tunggu dari nasabah (satuan) yang memerlukan layanan dari satu atau lebih pelayan (fasilitas layanan) (Siagian,1987). Proses antrian (*queueing process*) adalah suatu proses yang berhubungan dengan kedatangan seseorang pelanggan pada suatu fasilitas pelayanan, kemudian menunggu dalam suatu baris (antrian) jika semua pelayannya sibuk, dan akhirnya meninggalkan fasilitas tersebut.

Antrian timbul disebabkan oleh kebutuhan akan layanan melebihi kemampuan (kapasitas) pelayanan atau fasilitas layanan, sehingga pengguna fasilitas yang tiba tidak bisa segera mendapat layanan disebabkan kesibukan layanan. Seringnya antrian dapat menyebabkan hilangnya pelanggan / klien sedangkan penambahan pelayanan dan fasilitas untuk mencegah / mengurangi terjadinya antrian seringkali membutuhkan biaya yang tidak sedikit.

Penentu aturan keputusan yang menjelaskan cara melayani pengantri. Lima bentuk disiplin pelayanan yang biasa digunakan, yaitu (Siagian,1987) :

1. *First-Come First-Served (FCFS)* atau *First-In First-Out (FIFO)*
Lebih dulu datang (sampai), lebih dulu dilayani (keluar). Misalnya, antrian pada loket pembelian tiket bioskop, antrian pembelian tiket kereta api.
2. *Last-Come First-Served (LCFS)* atau *Last-In First-Out (LIFO)*
Datang terakhir yang lebih dulu keluar. Misalnya, pasien dalam kondisi kritis, walaupun dia datang paling akhir tetapi dia akan dilayani terlebih dahulu.
3. *Service In Random Order (SIRO)*
Panggilan didasarkan pada peluang secara random, tidak soal siapa yang lebih dulu tiba. Misalnya, pada arisan di mana penarikan berdasarkan nomor undian.

4. *Priority Service (PS)*

Prioritas pelayanan diberikan kepada pelanggan yang mempunyai prioritas lebih tinggi dibandingkan dengan pelanggan yang mempunyai prioritas lebih rendah, meskipun yang terakhir ini kemungkinan sudah lebih dahulu tiba dalam garis tunggu. Kejadian seperti ini kemungkinan disebabkan oleh beberapa hal, misalnya seseorang yang dalam keadaan penyakit lebih berat dibanding dengan orang lain dalam suatu tempat praktek dokter.

Dalam hal di atas telah dinyatakan bahwa entitas yang berada dalam garis tunggu tetap tinggal di sana sampai dilayani. Hal ini bisa saja tidak terjadi. Misalnya, seorang pembeli bisa menjadi tidak sabar menunggu antrian dan meninggalkan antrian. Untuk entitas yang meninggalkan antrian sebelum dilayani digunakan istilah pengingkaran (*reneging*). Pengingkaran dapat bergantung pada panjang garis tunggu atau lama waktu tunggu. Istilah penolakan (*balking*) dipakai untuk menjelaskan entitas yang menolak untuk bergabung dalam garis tunggu (Setiawan, 1991).

2.1.1 Sistem Antrian

Sebuah sistem antrian adalah suatu himpunan pelanggan, pelayan dan suatu aturan yang mengatur kedatangan pada pelanggan dan pemroses masalahnya. Pada umumnya, sistem antrian dapat diklasifikasikan menjadi sistem yang berbeda-beda dimana teori antrian dan simulasi sering diterapkan secara luas. Sistem antrian dapat diklasifikasi sebagai berikut (Hillier dan Lieberman, 1980):

1. Sistem pelayanan komersial.
Sistem pelayanan komersial merupakan aplikasi yang sangat luas dari model-model antrian, seperti restoran, kafetaria, toko-toko, salon, butik, supermarket, dan sebagainya.
 2. Sistem pelayanan bisnis-industri.
Sistem pelayanan bisnis-industri mencakup lini produksi, sistem *material-handling*, sistem pergudangan, dan sistem-sistem informasi komputer.
 3. Sistem pelayanan transportasi.
Sistem pelayanan sosial merupakan sistem-sistem pelayanan yang dikelola oleh perusahaan transportasi seperti busway, Kereta Api, pesawat dan lain - lain
-

4. Sistem pelayanan sosial.

Sistem pelayanan sosial merupakan sistem-sistem pelayanan yang dikelola oleh kantor-kantor dan jawatan-jawatan lokal maupun nasional, seperti kantor registrasi SIM dan STNK, kantor pos, rumah sakit, puskesmas, dan lain-lain (Subagyo,2000).

2.1.2 Struktur Antrian

Ada 4 model struktur antrian dasar yang umum terjadi dalam seluruh sistem antrian :

1. *Single Channel – Single Phase*

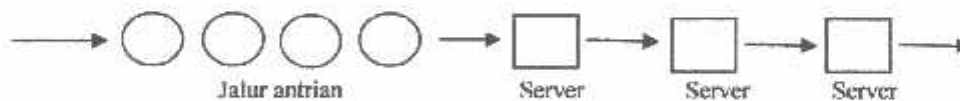
Single Channel berarti hanya ada satu jalur yang memasuki sistem pelayanan atau ada satu fasilitas pelayanan. Single Phase berarti hanya ada satu fasilitas pelayanan. Contohnya adalah sebuah kantor pos yang hanya mempunyai satu loket pelayanan dengan jalur satu antrian.



Gambar 2.1 Single Channel – Single Phase

2. *Single Channel – Multi Phase*

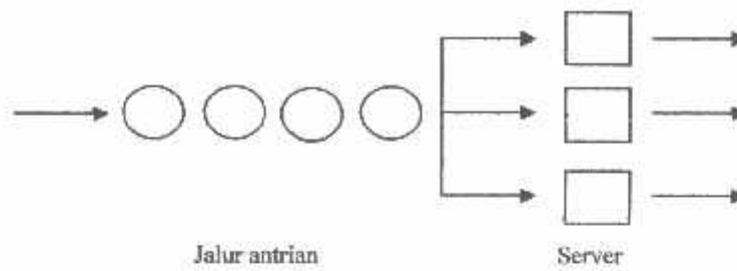
Sistem antrian jalur tunggal dengan tahapan berganda ini atau menunjukkan ada dua atau lebih pelayanan yang dilaksanakan secara berurutan. Sebagai contoh adalah : pencucian mobil, tukang cat mobil, dan sebagainya.



Gambar 2.2 Single Channel – Multi Phase

3. *Multi Channel – Single Phase*

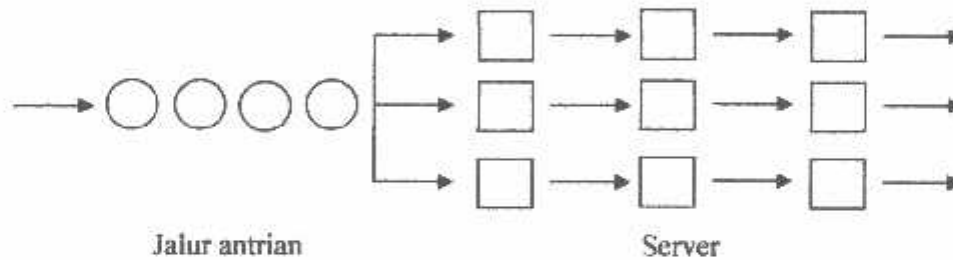
Sistem *Multi Channel – Single Phase* terjadi di mana ada dua atau lebih fasilitas pelayanan dialiri oleh antrian tunggal. Contohnya adalah antrian pada sebuah bank dengan beberapa teller, pembelian tiket atau karcis yang dilayani oleh beberapa loket, pembayaran dengan beberapa kasir, dan lain-lain.



Gambar 2.3 Multi Channel – Single Phase

4. *Multi Channel – Multi Phase*

Sistem Multi Channel – Multi Phase ini menunjukkan bahwa setiap sistem mempunyai beberapa fasilitas pelayanan pada setiap tahap sehingga terdapat lebih dari satu pelanggan yang dapat dilayani pada waktu bersamaan. Contoh pada model ini adalah : pada pelayanan yang diberikan kepada pasien di rumah sakit dimulai dari pendaftaran, diagnosa, tindakan medis, sampai pembayaran, registrasi ulang mahasiswa baru pada sebuah universitas, dan lain-lain.



Gambar 2.4 Multi Channel – Multi Phase

2.2 *Borland Delphi 7*

Borland Delphi 7 merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi visual. Keunggulan bahasa pemrograman ini terletak pada produktivitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta diperkuat dengan program yang terstruktur. Keunggulan lain dari *Delphi* ini adalah dapat digunakan untuk merancang program aplikasi yang memiliki tampilan seperti program aplikasi lain yang berbasis *Windows* (Andi, 2002:1).

Khusus untuk pemrograman *database*, *Borland Delphi 7* menyediakan fasilitas objek yang kuat dan lengkap yang memudahkan *programer* dalam membuat program.

Format *database* yang dimiliki *Delphi* adalah format *databaseParadox*, *dBase*, *MS.Access*, *ODBC*, *SyBASE*, *Oracle* dan lain-lain.

2.2.1 Sejarah Borland Delphi 7.0

Pada tahun 1992 muncul bahasa pemrograman baru bernama *Borland Pascal 7* yang merupakan penggabungan dari *Turbo Pascal* dan *Turbo pascal for windows*. Namun ternyata bahasa pemrograman baru tersebut ternyata masih sulit di gunakan. Trend penggunaan bahasa pemrograman visual untuk membangun sebuah aplikasi telah mendorong Borland membuat bahasa pemrograman baru pada tahun 1995 diperkenalkan kepengguna komputer sebuah bahasa pemrograman visual yang berbasis bahasa pascal. Bahasa pemrograman baru tersebut diberi nama *Borland Delphi* setahun kemudian versi ke dua dari *delphi* dilempar ke pasaran.

Penggunaan *delphi* dapat mempersingkat waktu pemrograman, karena tidak perlu lagi menuliskan kode program yang rumit dan panjang untuk menggambar, meletakkan dan mengatur komponen. Selain itu, dapat menyusun aplikasi yang lebih interaktif. *Delphi* menyediakan cukup banyak pilihan komponen *interface* aplikasi, antara lain berupa tombol menu, *drop down*, ataupun menu *pop up*, kotak *text*, *radio button*, *check box*, dan sebagainya.

Bahkan ada berbagai macam komponen *Skin* tampilan yang beragam yang disediakan oleh beberapa vendor lainnya, seperti, *suipack*, *Sxskincomponents*, dan lain sebagainya. Anda tinggal memilih komponen yang dibutuhkan dengan klik mouse, mengatur tampilannya kemudian menuliskan sedikit kode program, maka aplikasi anda siap dijalankan.

Delphi 7.0, versi terbaru yang dikeluarkan oleh *Borland*, memiliki *support* yang sangat tinggi terhadap *database-database* yang sudah terkenal (seperti *MS Accsses*, *Paradox*, *Foxpro*, *Dbase*, *Oracce*, dan lain sebagainya), dan dilengkapi dengan objek-objek yang baru sehingga memudahkan pembuatan *database* maupun program lainnya (*Game*, *Utility* dan lainnya).

2.2.2 Kelebihan Borland Delphi 7

Berikut ini sebagian beberapa kelebihan *Borland Delphi 7.0* diantaranya (Andi, 2002:1) :

1. Berbasiskan *OOP (Object Oriented Programming)*.
Setiap bagian yang ada pada program dipandang sebagai suatu object yang mempunyai sifat-sifat yang dapat diubah dan diatur.
2. IDE yang berkualitas.
Delphi memiliki lingkungan pengembangan yang lengkap. Terdapat menu-menu yang memudahkan anda mengatur proyek pengembangan software.
3. Proses Kompilasi yang cepat.
Delphi memiliki kecepatan kompilasi yang tidak perlu diragukan. Saat aplikasi yang anda buat dijalankan di lingkungan *Delphi*, aplikasi tersebut otomatis di *compile* secara terpisah.
4. *Multi-purpose*
Dapat digunakan untuk berbagai keputusan pengembangan aplikasi mulai perhitungan sederhana sampai aplikasi multimedia bahkan yang terkoneksi ke internet.
5. Satu file *Exe*.
Setelah merancang program dalam IDE *Delphi*, *Delphi* akan mengkompilasinya menjadi sebuah *file executeable* tunggal. Program yang dibuat langsung didistribusikan dan dijalankan pada komputer lain tanpa perlu menyertakan file *DLL* dari luar. Ini merupakan sebuah kelebihan yang sangat berarti.

2.2.3 *Interbase*

Interbase merupakan program aplikasi *database* untuk menangani *database client server* yang didistribusikan oleh sebuah perusahaan perangkat lunak yang terkenal yaitu *Borland (Wahana, 2010)*. *Interbase server* ditemukan satu paket dengan program *Delphi* untuk semua edisi.

Keuntungan menggunakan *Interbase* adalah sebagai berikut:

1. *Interbase* memiliki semua fitur, kekuatan, dan skalabilitas yang diperlukan untuk aplikasi bisnis yang kompleks yang menawarkan keunggulan performa untuk jumlah pengguna yang besar.
 2. *Borland Interbase* didesain untuk dapat diinstall dengan mudah pada lokasi *end-user* tanpa pertolongan administrator *database*. *Interbase* dapat di-install secara terselubung, sehingga *end-user* tidak tahu jika sudah terinstal.
 3. *Interbase* mendukung berbagai kumpulan bahasa untuk meningkatkan portabilitas dan skalabilitas global.
-

Interbase memiliki dua tipe arsitek, yaitu *Super Server* dan *Classic*. *Super Server* merupakan *multi-client*, implementasi *multi-thread* dari proses *server InterBase*. Implementasi ini menggantikan jenis implementasi model *Classic* yang telah digunakan pada versi *InterBase* sebelumnya.

2.3 SMS (Short Message Service)

Short Message Service (SMS) adalah kemampuan untuk mengirim dan menerima pesan dalam bentuk teks dari dan kepada ponsel. Teks tersebut bisa terdiri dari kata-kata atau nomor atau kombinasi *alphanumeric*. SMS diciptakan sebagai standar pesan (*message*) oleh ETSI (*European Telecommunication Standards Institute*), yang juga membuat standar GSM yang diimplementasikan oleh semua operator GSM. SMS yang pertama dikirimkan pada Desember 1992 dari PC ke sebuah ponsel melalui jaringan GSM Vodafone di UK. Setiap Pesan maksimal terdiri dari 160 karakter jika menggunakan alfabet Latin, dan 70 karakter jika menggunakan alfabet non-Latin seperti huruf Arab atau China. *Short Message Service* atau biasa disingkat SMS merupakan sebuah layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel (*wireless*), memungkinkan dilakukannya pengiriman pesan dalam bentuk *alphanumeric* antara terminal pelanggan atau antar terminal pelanggan dengan sistem eksternal, seperti *e-mail*, *paging*, *voice mail*, dan lain-lain. Aplikasi SMS merupakan aplikasi yang paling banyak peminat dan penggunaannya. Hal ini dapat dibuktikan dengan munculnya berbagai jenis aplikasi yang memanfaatkan fasilitas SMS. Teknologi SMS memiliki beberapa keunggulan, yaitu harganya murah, merupakan "*Delivered Oriented Service*", artinya pesan akan selalu diusahakan untuk dikirimkan ke tujuan. Jika suatu saat nomor tujuan sedang tidak aktif atau di luar jaringan, maka pesan akan disimpan di SMSC (*SMS Center*) server dan akan dikirimkan segera setelah nomor tujuan aktif kembali. Pesan juga akan terkirim ke tujuan walaupun nomor tujuan sedang melakukan pembicaraan (*sibuk*).

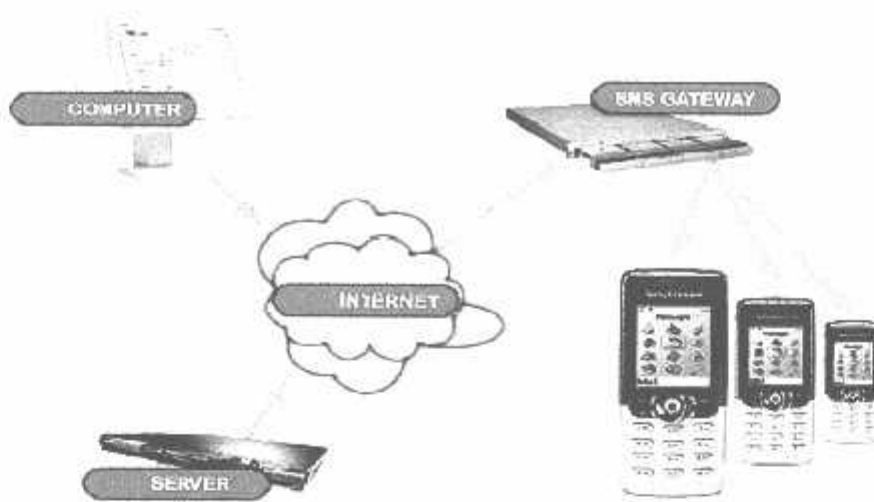
Cara kerja SMS : Saat kita menerima pesan SMS/MMS dari *handphone* (*mobile originated*), pesan tersebut tidak langsung dikirimkan ke *handphone* tujuan (*mobile terminated*), akan tetapi dikirim terlebih dahulu ke *SMS Center* (SMSC) yang biasanya berada di kantor operator telepon, baru kemudian pesan tersebut diteruskan ke *handphone* tujuan. Dengan adanya SMSC, kita dapat mengetahui status dari pesan SMS yang telah dikirim, apakah telah sampai atau gagal.

2.4 SMS Gateway

Istilah *gateway* dapat diartikan sebagai pintu gerbang. Namun pada dunia komputer, *gateway* dapat diartikan sebagai jembatan penghubung antara satu sistem dengan sistem yang lain, sehingga dapat terjadi pertukaran data antar sistem tersebut. *SMS Gateway* adalah platform yang menyediakan mekanisme untuk menghantar dan menerima *SMS* dari peralatan mobile (HP, PDA phone, dan lain-lain) yang menggunakan *keyword* tertentu (Novianti, 2009:2). *SMS Gateway* adalah komunikasi *SMS* dua arah dan merupakan salah satu perkembangan fungsi yang dimiliki *SMS*. Secara umum *SMS Gateway* adalah sebuah sistem yang digunakan untuk memudahkan seseorang atau sebuah perusahaan mengirimkan pesan *SMS* yang sama dalam waktu yang bersamaan pada banyak orang. Selain itu, semakin berkembangnya fungsi *SMS*, *SMS Gateway* juga dapat digunakan untuk keperluan lain seperti melakukan *polling*, transaksi dengan sebuah sistem, pemantauan, dan sebagainya.

SMS gateway dapat juga diartikan sebagai pintu gerbang atau jalur bagi penyebaran informasi dengan menggunakan *SMS*. Dengannya, kita dapat menyebarkan pesan ke banyak nomor sekaligus secara otomatis dan cepat. Dengan menggunakan program tambahan yang dapat dibuat sendiri, si pengirim pesan dapat lebih fleksibel dalam mengirim/menerima *SMS* sesuai dengan aturan yang ditetapkan, seperti sistem untuk antrian.

Cara kerja *SMS Gateway* adalah *SMS* dikirim pemilih ke nomor yang telah ditentukan, isi *SMS* merupakan Format yang telah ditentukan sebelumnya. *SMS* diterima oleh *Handphone* yang bekerja sebagai *SMS Gateway* kemudian akan dikirimkan ke sistem, sistem akan merespon *SMS* tersebut. Jika Format *SMS* benar maka *Handphone* yang bertindak sebagai *SMS Gateway Server* akan membalas *SMS* tersebut bahwa *SMS* telah ditampung begitu juga sebaliknya. Kemudian *SMS* tersebut ditampung dalam *database* dan akan ditampilkan kedalam *Website*.



Gambar 2.5 Cara Kerja SMS Gateway

2.5 MySQL

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Suatu sistem *database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *singleuser*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari *Postgre SQL* dan lima kali lebih cepat dibandingkan *Interhase*. *MySQL* sendiri menjadi salah satu sistem manajemen *database* yang banyak digunakan untuk pengembangan *web* (Wahana, 2010:6).

2.5.1 Keistimewaan MySQL

MySQL memiliki beberapa keistimewaan, antara lain :

1. *Portabilitas*, MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga*

2. *Open Source*, MySQL didistribusikan secara open source, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.
 3. *Multiuser*, MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
 4. *Performance tuning*, MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
 5. *Jenis Kolom*, MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
 6. *Perintah dan Fungsi*, MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
 7. *Keamanan*, MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
 8. *Skalabilitas dan Pembatasan*, MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
 9. *Konektivitas*, MySQL dapat melakukan koneksi dengan klien menggunakan protokol *TCP/IP, Unix socket (UNIX)*, atau *Named Pipes (NT)*.
 10. *Lokalisasi*, MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
 11. *Antar Muka*, MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
 12. *Klien dan Peralatan*, MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.
 13. *Struktur tabel*, MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya semacam *PostgreSQL* ataupun *Oracle*.
-

2.5.2 Perkembangan MySQL

MySQL merupakan hasil buah pikiran dari Michael "Monty" Widenius, David Axmark dan Allan Larsson yang dimulai pada tahun 1995. Mereka kemudian mendirikan perusahaan bernama MySQL AB di Swedia. Tujuan awal ditulisnya program ini adalah untuk mengembangkan aplikasi web yang akan digunakan oleh salah satu klien MySQL AB.

Tidak sama dengan proyek-proyek seperti Apache, di mana perangkat lunak dikembangkan oleh komunitas umum dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, di mana memegang hak cipta hampir atas semua kode sumbernya.

MySQL versi 1.0 dirilis pada Mei 1996 dan penggunaannya hanya terbatas di kalangan intern saja. Pada bulan Oktober 1996, MySQL versi 3.11.0 dirilis ke masyarakat luas di bawah lisensi "Terbuka Tapi Terbatas". Barulah pada bulan Juni 2000, MySQL AB mengumumkan bahwa mulai MySQL versi 3.23.19 diterapkan sebagai GPL (General Public License). Dengan lisensi ini, maka "Siapa pun boleh melihat program aslinya dan menggunakan program eksekusinya dengan cuma-cuma. Dan bila ingin memodifikasi pada program aslinya, maka program modifikasi tersebut harus dilepas di bawah lisensi GPL juga".

Hingga kini MySQL terus berkembang dengan berbagai penyempurnaan sehingga dirilis MySQL versi 5.0.21 pada bulan Mei 2006, yang semula untuk melakukan administrasi dalam basis data MySQL, dengan menggunakan modul yang berupa *Command Prompt* (perintah: `mysql` dan `mysqladmin`). Saat ini dapat dirancang sebuah *database* melalui modul berbasis grafik GUI (*Graphical User Interface*).

2.5.3 Perintah Dasar MySQL

Dalam menjalankan MySQL diperlukan berbagai perintah untuk membuat suatu *database*, berikut ini disebutkan beberapa perintah dasar dalam menggunakan MySQL. Untuk menjalankan MySQL pertama kali cukup dengan mengetikkan MySQL pada *Command Prompt*. Perintah-perintahnya adalah sebagai berikut :

1. Menampilkan *database* : `SHOW DATABASE ;`
 2. Membuat *database* baru : `CREATE DATABASE ;`
 3. Memilih *database* yang akan digunakan : `USE database ;`
 4. Menampilkan tabel : `SHOW TABLE ;`
-

5. Membuat tabel baru : `CREATE TABLE tabel (field spesifikasi_field,...) ;`
6. Menampilkan struktur tabel : `SHOW COLUMNS FROM tabel` atau `DESCRIBE tabel ;`
7. Mengubah struktur tabel : `ALTER TABLE tabel Jenis_Pengubahan ;`
8. Mengisikan data : `INSERT INTO tabel (kolom1,...) VALUES ('data_kolom1',...);` atau `INSERT INTO tabel SET kolom1 = 'data_kolom1',...;`
9. Menampilkan data : `SELECT kolom FROM tabel WHERE kriteria ORDER BY kolom` atau `SELECT * FROM tabel ;`
10. Mengubah data : `UPDATE tabel SET kolom = pengubahan_data WHERE kriteria;`
11. Menghapus data : `DELETE FROM tabel WHERE kriteria ;`
12. Menghapus tabel : `DROP tabel ;`
13. Menghapus *database* : `DROP database ;`

2.6 *Gammu*

Gammu adalah sebuah aplikasi *cross-platform* yang digunakan untuk menjembatani / mengomunikasikan antara *database SMS Gateway* dengan *sms devices*. Sedangkan menurut Novianti (2009), *Gammu* adalah *tool* untuk mengembangkan dan menerapkan aplikasi *SMS Gateway* yang mudah diimplementasikan dan gratis. Aplikasi *Gammu* berupa daemon yang berjalan secara *background*. Setiap saat, *gammu* memonitor *sms devices* dan *database sms gateway*. Saat ada sms masuk ke *sms devices*, maka *gammu* langsung memindahkannya ke dalam *inbox* dalam *database sms gateway*. Sebaliknya saat aplikasi pengirim SMS memasukkan sms ke dalam *outbox* dalam *database sms gateway*, maka *gammu* mengirimkannya melalui *sms devices*, dan memindahkan sms ke *sent item* dalam *database*.

Kelebihan *Gammu* dari *SMS Gateway* dengan lainnya adalah :

- a. *Gammu* dapat berjalan di *Windows* maupun *Linux* .
 - b. Banyak *device* ataupun ponsel yang *support* dengan *gammu*.
 - c. *Gammu* menggunakan *database MySql* dan dapat digunakan untuk penerapan aplikasi *desktop* dan *interface web-based*.
 - d. *Gammu* dapat menggunakan fitur-fitur yang ada pada ponsel dengan lebih efisien.
-

2.7 Handphone

Telepon seluler (ponsel) atau telepon genggam (telgam) atau *handphone* (HP) atau disebut pula adalah perangkat telekomunikasi elektronik yang mempunyai kemampuan dasar yang sama dengan telepon konvensional saluran tetap, namun dapat dibawa ke mana-mana (*portabel, mobile*) dan tidak perlu disambungkan dengan jaringan telepon menggunakan kabel (*nirkabel; wireless*). Saat ini Indonesia mempunyai dua jaringan telepon nirkabel yaitu sistem GSM (*Global System for Mobile Telecommunications*) dan sistem CDMA (*Code Division Multiple Access*). Badan yang mengatur telekomunikasi seluler Indonesia adalah Asosiasi Telekomunikasi Seluler Indonesia (ATSI).

Selain berfungsi untuk melakukan dan menerima panggilan telepon, ponsel umumnya juga mempunyai fungsi pengiriman dan penerimaan pesan singkat (*short message service, SMS*). Ada pula penyedia jasa telepon genggam di beberapa negara yang menyediakan layanan generasi ketiga (3G) dengan menambahkan jasa *videophone*, sebagai alat pembayaran, maupun untuk televisi *online*. Sekarang, telepon genggam menjadi *gadget* yang multifungsi. Mengikuti perkembangan teknologi digital, kini ponsel juga dilengkapi dengan berbagai pilihan fitur, seperti bisa menangkap siaran radio dan televisi, perangkat lunak pemutar audio (MP3) dan video, kamera digital, *game*, dan layanan internet (WAP, GPRS, 3G). Selain fitur-fitur tersebut, ponsel sekarang sudah ditanamkan fitur komputer. Jadi di ponsel tersebut, orang bisa mengubah fungsi ponsel tersebut menjadi mini komputer. Di dunia bisnis, fitur ini sangat membantu bagi para pebisnis untuk melakukan semua pekerjaan di satu tempat dan membuat pekerjaan tersebut diselesaikan dalam waktu yang singkat.

BAB III PERANCANGAN

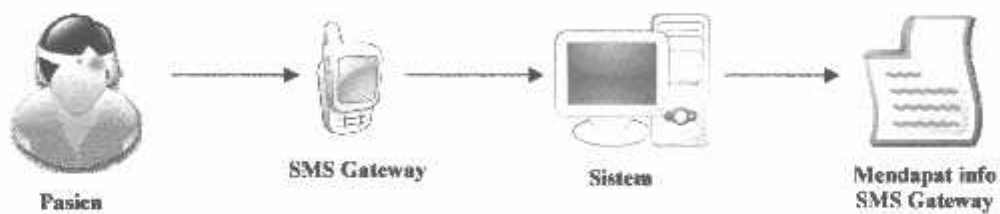
Bab ini membahas mengenai perancangan proyek akhir, perancangan sistem, pembuatan dan pengujian *database*

3.1 Rancangan Proyek Akhir

Perancangan proyek akhir ini meliputi desain sistem, DFD (*Data Flow Diagram*), ERD (*Entity Relationship Diagram*), dan diagram alir program (*Flowchart*).

3.1.1 Desain Sistem

Desain sistem yang dibuat ditunjukkan dalam Gambar 3.1



Gambar 3.1 Desain Sistem

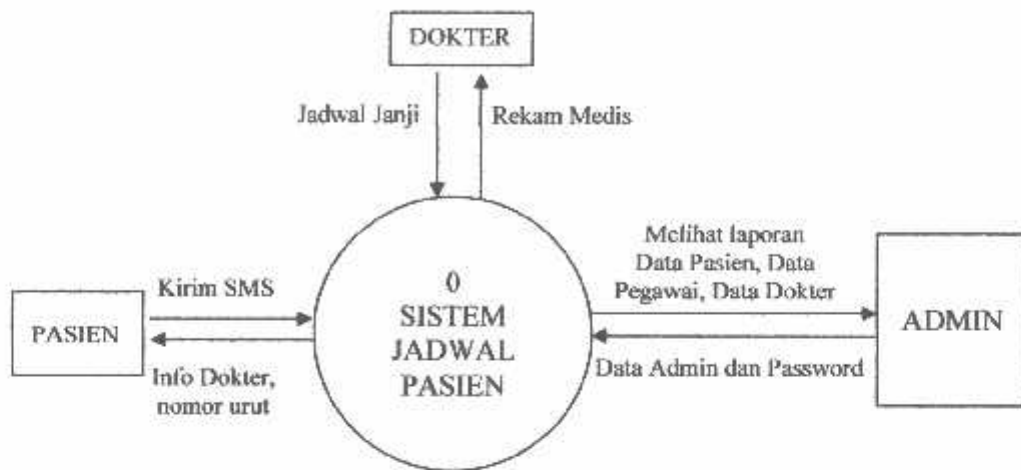
Sistem ini menggunakan *SMS Gateway* untuk akses informasi, pasien dapat mengetahui informasi tentang kehadiran dokter dan mendapatkan nomor antrian untuk pemeriksaan. Apabila pasien yang bersangkutan telah terdaftar dalam *database*, maka dapat melakukan pemesanan nomor urut antrian.

3.1.2 *Data Flow Diagram* (DFD)

DFD merupakan alat perencanaan sistem yang berorientasi pada alur data dengan konsep dekomposisi yang dapat digunakan untuk menggambarkan analisis maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

3.1.2.1 Data Flow Diagram (DFD) Level 0

Data flow diagram (DFD) level 0 dari Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway*, dapat dilihat dalam Gambar 3.2

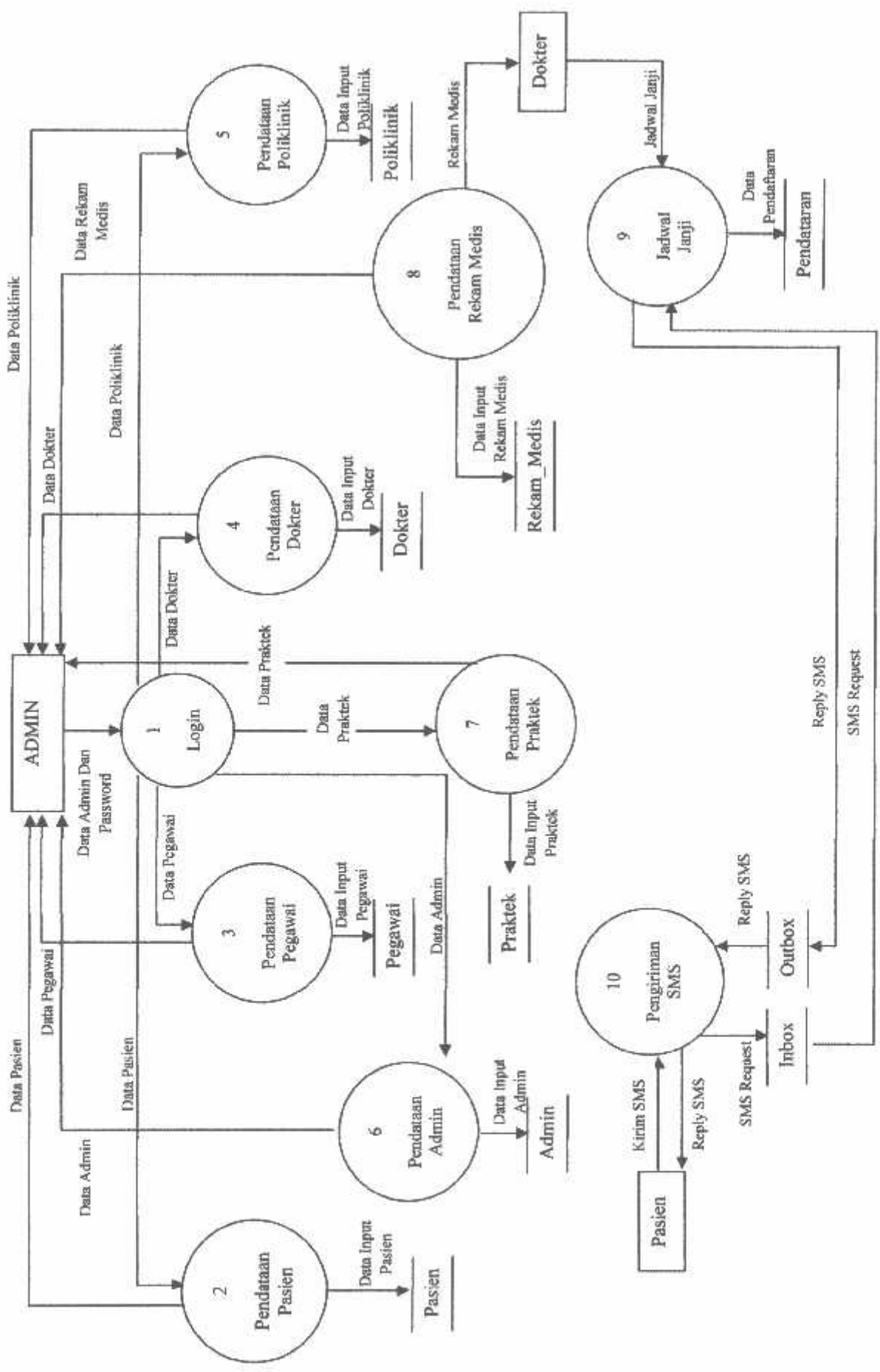


Gambar 3.2 Data Flow Diagram (DFD) Level 0 Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway*

DFD Level 0 pada gambar 3.2 menjelaskan terdapat tiga entitas yang berkaitan dengan sistem, yaitu pasien yang bertindak sebagai user yang dapat mengirim *SMS* dan mendapatkan informasi dokter jaga serta nomor urut antrian. Entitas yang kedua yaitu dokter yang mempunyai jadwal praktek dan membuat rekam medis pasien. Entitas yang ketiga yaitu admin yang bertugas menginputkan data pasien baru, data pegawai dan data dokter, sistem antrian pasien berbasis *SMS Gateway* dapat diakses apabila admin telah melakukan proses *login*.

3.1.2.2 Data Flow Diagram (DFD) Level 1

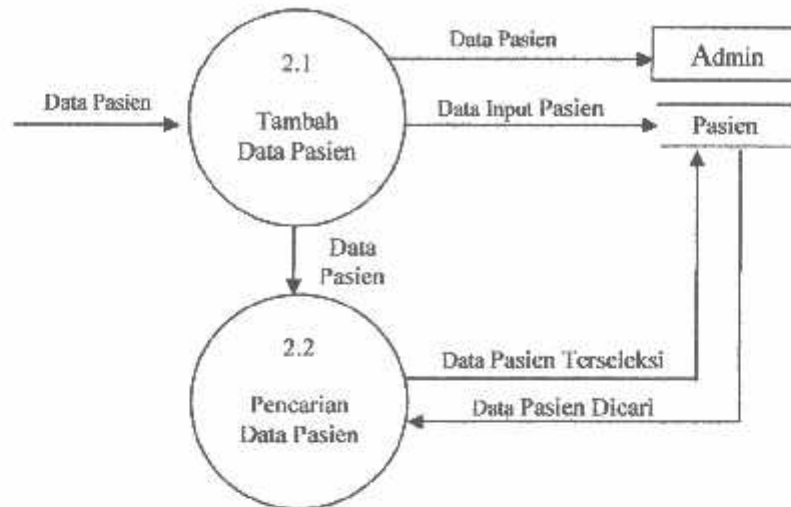
Data flow diagram (DFD) level 1 dari Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway*, dapat dilihat dalam Gambar 3.3. DFD level 1 menjelaskan jalannya proses sistem dengan lebih terperinci, sehingga terlihat *database* yang digunakan dalam sistem dan hubungan antara *database* itu dengan sistem maupun dengan pasien, dokter dan admin.



Gambar 3.3 Data Flow Diagram (DFD) Level I Perancangan Sistem Antrian Pasien Poliklinik Berbasis SMS Gateway

3.1.2.3 Data Flow Diagram (DFD) Level 2 Pendataan Pasien

Data flow diagram (DFD) level 2 pendataan pasien dapat dilihat dalam Gambar 3.4

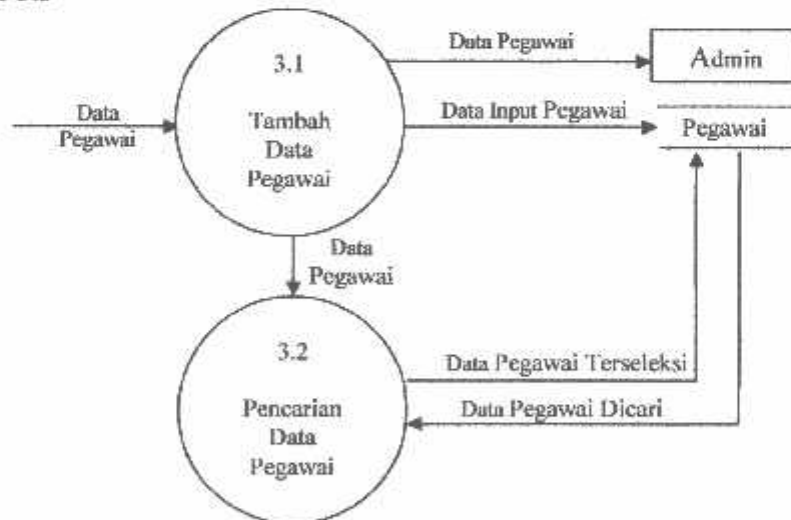


Gambar 3.4 Data Flow Diagram (DFD) Level 2 Pendataan Pasien

Penambahan pasien baru dilakukan oleh admin dengan mengisi data pasien baru yang kemudian akan disimpan pada tabel pasien. Dalam fungsi pencarian data pasien, dicari pada tabel pasien dan dicocokkan dengan input data pasien yang dicari.

3.1.2.4 Data Flow Diagram (DFD) Level 2 Pendataan Pegawai

Data flow diagram (DFD) level 2 pendataan pegawai dapat dilihat dalam Gambar 3.5

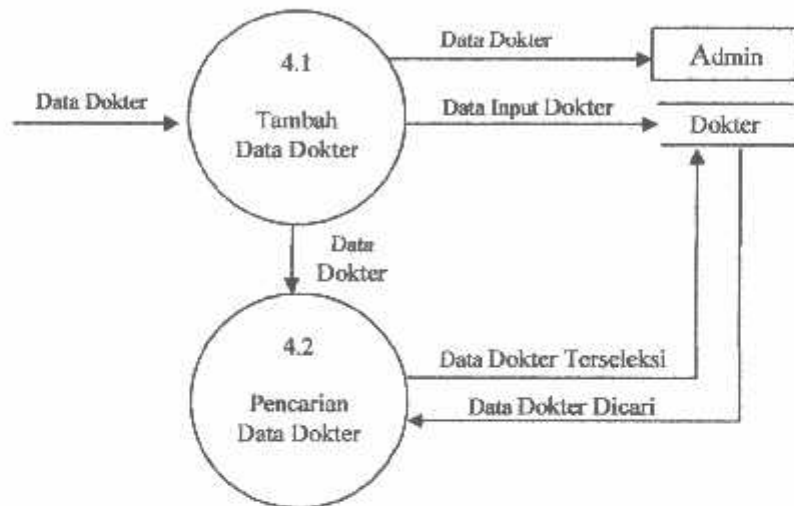


Gambar 3.5 Data Flow Diagram (DFD) Level 2 Pendataan Pegawai

Penambahan pegawai baru dilakukan oleh admin dengan mengisi data pegawai baru yang kemudian akan disimpan pada tabel pegawai. Dalam fungsi pencarian data pegawai, dicari pada tabel pegawai dan dicocokkan dengan input data pegawai yang dicari.

3.1.2.5 Data Flow Diagram (DFD) Level 2 Pendataan Dokter

Data flow diagram (DFD) level 2 pendataan dokter dapat dilihat dalam Gambar 3.6



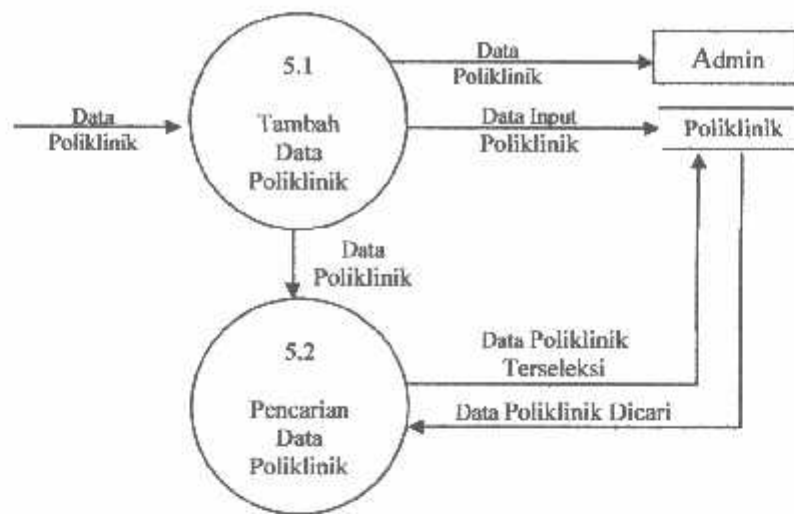
Gambar 3.6 Data Flow Diagram (DFD) Level 2 Pendataan Dokter

Penambahan dokter baru dilakukan oleh admin dengan mengisi data dokter baru yang kemudian akan disimpan pada tabel dokter. Dalam fungsi pencarian data dokter, dicari pada tabel dokter dan dicocokkan dengan input data dokter yang dicari.

3.1.2.6 Data Flow Diagram (DFD) Level 2 Pendataan Poliklinik

Penambahan poliklinik baru dilakukan oleh admin dengan mengisi data poliklinik baru yang kemudian akan disimpan pada tabel poliklinik. Dalam fungsi pencarian data poliklinik, dicari pada tabel poliklinik dan dicocokkan dengan input data poliklinik yang dicari.

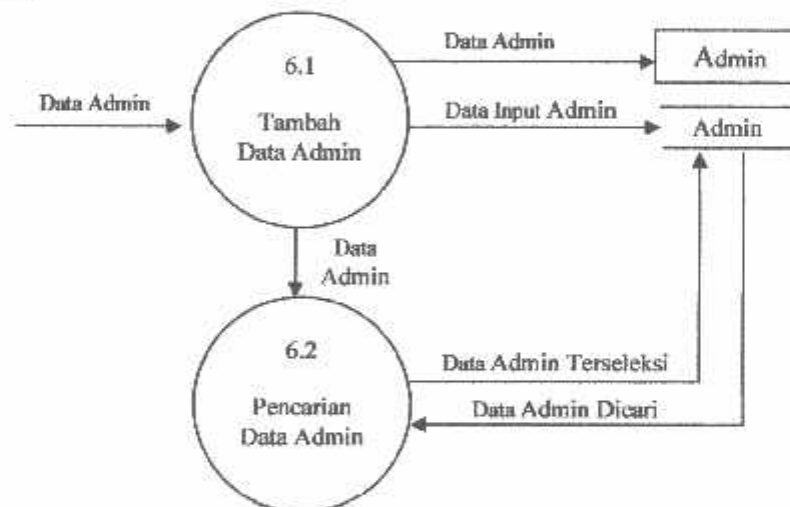
Data flow diagram (DFD) level 2 pendataan poliklinik dapat dilihat dalam Gambar 3.7



Gambar 3.7 Data Flow Diagram (DFD) Level 2 Pendaftaran Poliklinik

3.1.2.7 Data Flow Diagram (DFD) Level 2 Pendaftaran Admin

Data flow diagram (DFD) level 2 pendaftaran admin dapat dilihat dalam Gambar 3.8

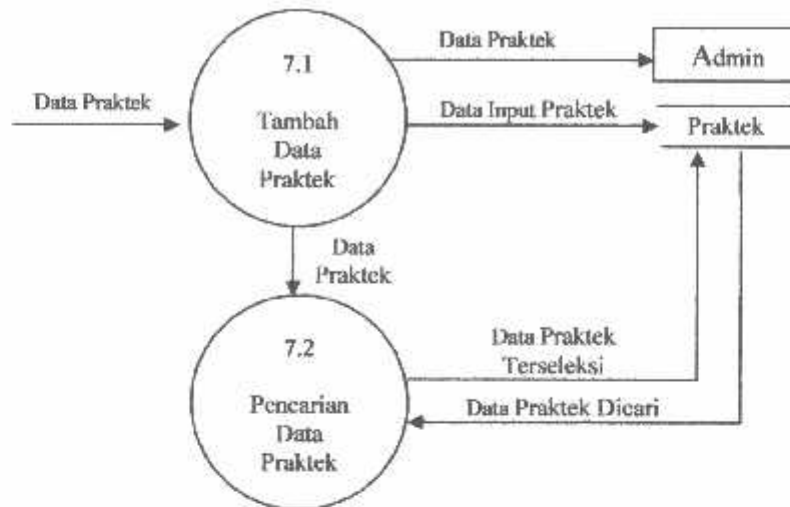


Gambar 3.8 Data Flow Diagram (DFD) Level 2 Pendaftaran Admin

Penambahan admin baru dilakukan oleh super admin dengan mengisi data admin baru yang kemudian akan disimpan pada tabel admin. Dalam fungsi pencarian data admin, dicari pada tabel admin dan dicocokkan dengan input data admin yang dicari.

3.1.2.8 Data Flow Diagram (DFD) Level 2 Pendataan Praktek

Data flow diagram (DFD) level 2 pendataan praktek dapat dilihat dalam Gambar 3.9

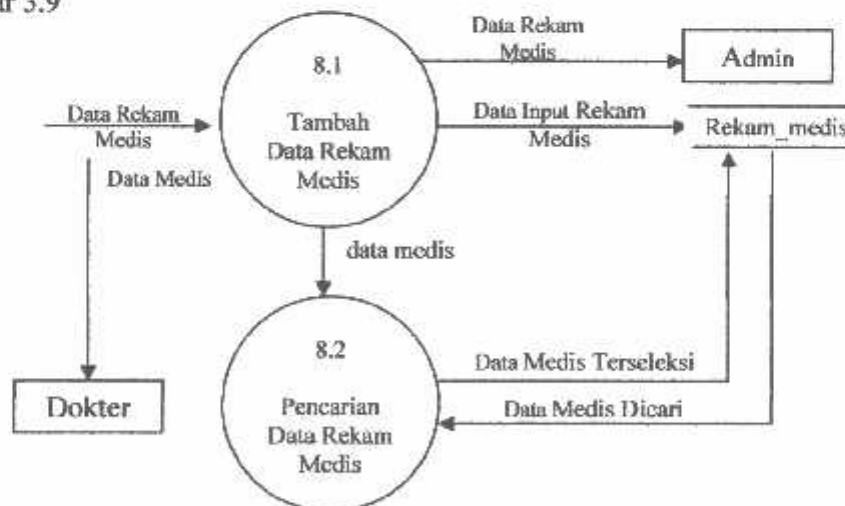


Gambar 3.9 Data Flow Diagram (DFD) Level 2 Pendataan Praktek

Penambahan praktek baru dilakukan oleh admin dengan mengisi data praktek baru yang kemudian akan disimpan pada tabel praktek. Dalam fungsi pencarian data praktek, dicari pada tabel praktek dan dicocokkan dengan input data praktek yang dicari.

3.1.2.9 Data Flow Diagram (DFD) Level 2 Pendataan Rekam Medis

Data flow diagram (DFD) level 2 pendataan rekam medis dapat dilihat dalam Gambar 3.9

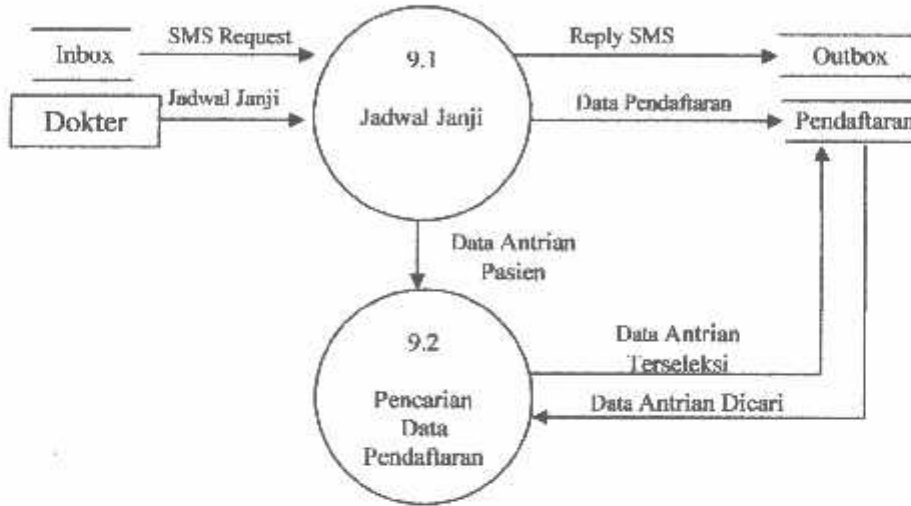


Gambar 3.10 Data Flow Diagram (DFD) Level 2 Pendataan Rekam Medis

Penambahan data rekam medis pasien dilakukan oleh admin dengan mengisi data pasien baru yang diperoleh dari diagnosa dokter, yang kemudian akan disimpan pada tabel *rekam_medis*. Dalam fungsi pencarian data medis pasien, dicari pada tabel *rekam_medis* dan dicocokkan dengan input data medis pasien yang dicari.

3.1.2.10 Data Flow Diagram (DFD) Level 2 Jadwal Janji

Data flow diagram (DFD) level 2 jadwal janji dapat dilihat dalam Gambar 3.11

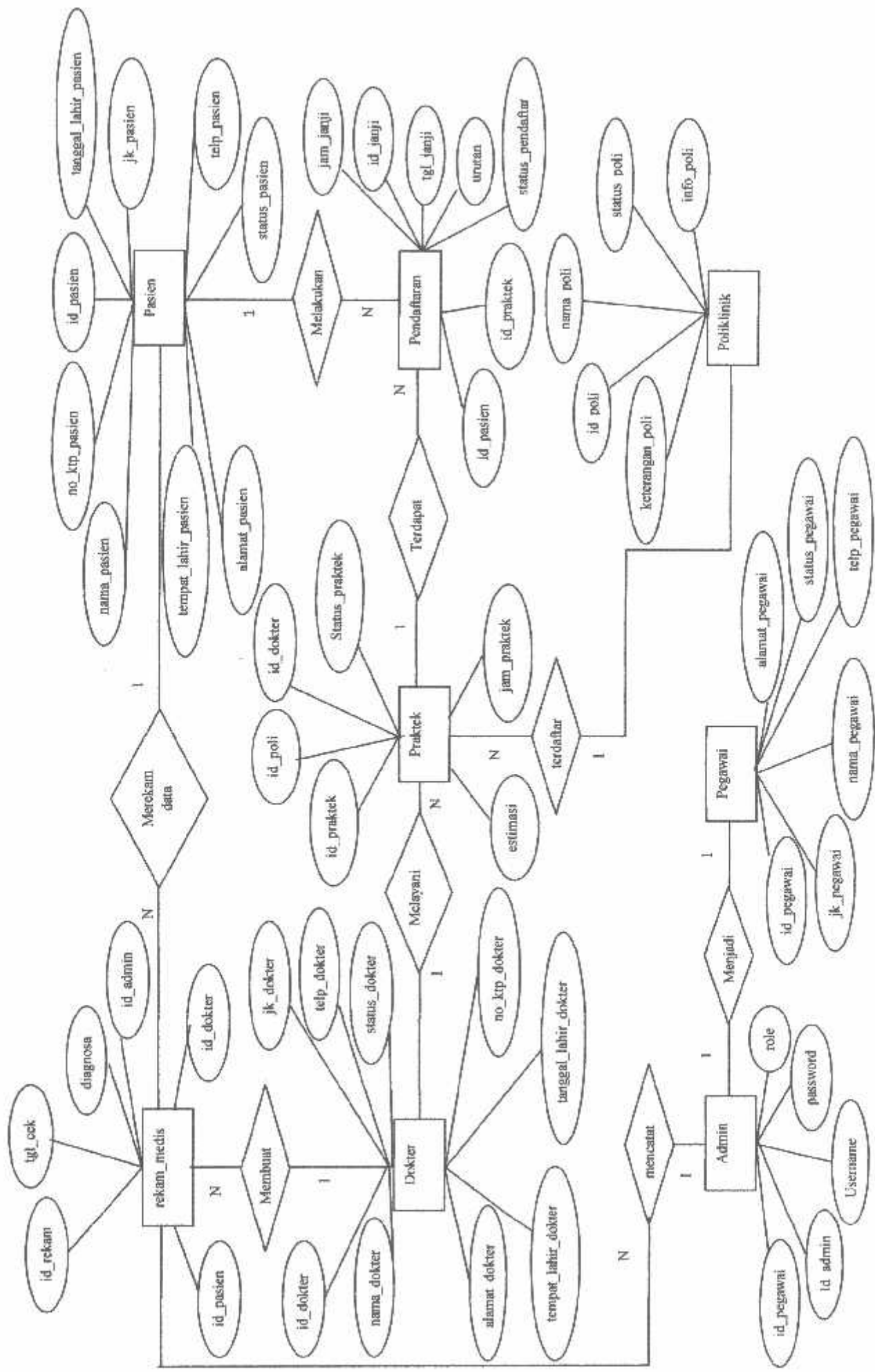


Gambar 3.11 Data Flow Diagram (DFD) Level 2 Jadwal Janji

Pendaftaran nomor antrian diperoleh dari *SMS request* pada tabel *inbox* dan disimpan pada tabel pendaftaran. *Reply SMS* pada tabel *outbox* akan dikirimkan kepada pasien. Dalam fungsi pencarian data antrian pasien, dicari pada tabel pendaftaran dan dicocokkan dengan input data antrian pasien yang dicari.

3.1.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) dari Perancangan Sistem Antrian Pasien poliklinik Berbasis *SMS Gateway*, dapat dilihat dalam Gambar 3.12



Gambar 3.12 Entity Relationship (ERD) Perancangan Sistem Antrian Pasien Poliklinik

3.1.4 *Physical Data Model*

Physical Data Model dari Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway*, seperti di bawah ini

3.1.4.1 Tabel Admin

Merupakan tabel yang menampung data-data admin

Tabel 3.1 Tabel Admin

Atribut	Tipe Data	Deskripsi
Id_Admin (PK)	Int 11	ID Admin
Username	Char 20	Username untuk login
Password	Char 20	Password untuk login

3.1.4.2 Tabel Dokter

Merupakan tabel yang menampung data-data dokter yang terdaftar pada *database* *jadwal_pasien*

Tabel 3.2 Tabel Dokter

Atribut	Tipe Data	Deskripsi
Id_Dokter (PK)	Int 11	ID dokter
No_KTP_Dokter	Char 30	Nomor KTP dokter
Nama_Dokter	Char 50	Nama dokter
Alamat_Dokter	Char 50	Alamat dokter
Tempat_Lahir_Dokter	Char 20	Tempat kelahiran dokter
Tanggal_Lahir_Dokter	Char 10	Tanggal lahir dokter
JK_Dokter	Char 2	Jenis kelamin dokter
Telp_Dokter	Char 15	Nomor telepon dokter
Status_Dokter	Char 5	Status true/false

3.1.4.3 Tabel Praktek

Merupakan tabel yang menampung data-data praktek yang terdaftar pada *database* *jadwal_pasien*

Tabel 3.7 Tabel Praktek

Atribut	Tipe Data	Deskripsi
Id_Praktek (PK)	Int 11	ID praktek
Jam_Praktek	Char 20	Jam praktek
Status_Praktek	Char 5	Status True/False

3.1.4.4 Tabel Pasien

Merupakan tabel yang menampung data-data pasien yang terdaftar pada *database* *jadwal_pasien*

Tabel 3.3 Tabel Pasien

Atribut	Tipe Data	Deskripsi
Id_Pasien (PK)	Char 10	ID pasien
No_KTP_Pasien	Char 30	Nomor KTP pasien
Nama_Pasien	Char 30	Nama pasien
Alamat_Pasien	Char 50	Alamat pasien
Tempat_Lahir_Pasien	Char 20	Tempat kelahiran pasien
Tanggal_Lahir_Pasien	Char 10	Tanggal lahir pasien
JK_Pasien	Char 2	Jenis Kelamin pasien
Telp_Pasien	Char 15	Nomor telepon
Status_Pasien	Char 5	Status True/False

3.1.4.5 Tabel Pegawai

Merupakan tabel yang menampung data-data pegawai yang terdaftar pada *database* *jadwal_pasien*

Tabel 3.4 Tabel Pegawai

Atribut	Tipe Data	Deskripsi
Id_Pegawai (PK)	Int 11	ID pegawai
Nama_Pegawai	Char 30	Nama pegawai
Alamat_Pegawai	Char 50	Alamat pegawai
JK_Pegawai	Char 2	Jenis kelamin pegawai
Telp_Pegawai	Char 15	Nomor telepon pegawai
Status_Pegawai	Char 5	Status Truc/False

3.1.4.6 Tabel Rekam Medis

Merupakan tabel yang menampung data-data rekam medis

Tabel 3.8 Tabel Rekam Medis

Atribut	Tipe Data	Deskripsi
Id_Rekam	Int 11	Id rekam medis pasien
Tgl_Cek	Char 10	Tanggal cek kesehatan pasien
Diagnosa	Char 500	Diagnosa dokter

3.1.4.7 Tabel Pendaftaran

Merupakan tabel yang menampung data-data pendaftaran yang terdaftar pada *database* *jadwal_pasien*

Tabel 3.5 Tabel Pendaftaran

Atribut	Tipe Data	Deskripsi
Id_Janji (PK)	Int 11	ID janji pasien
Tgl_Janji	Char 10	Tanggal janji pasien
Jam_Janji	Char 15	Jam janji pasien
Urutan	Int 11	Urutan nomor pendaftaran
Status_Pendaftaran	Char 15	Status True/False

3.1.4.8 Tabel Poliklinik

Merupakan tabel yang menampung data-data poliklinik yang terdaftar pada *database* *jadwal_pasien*

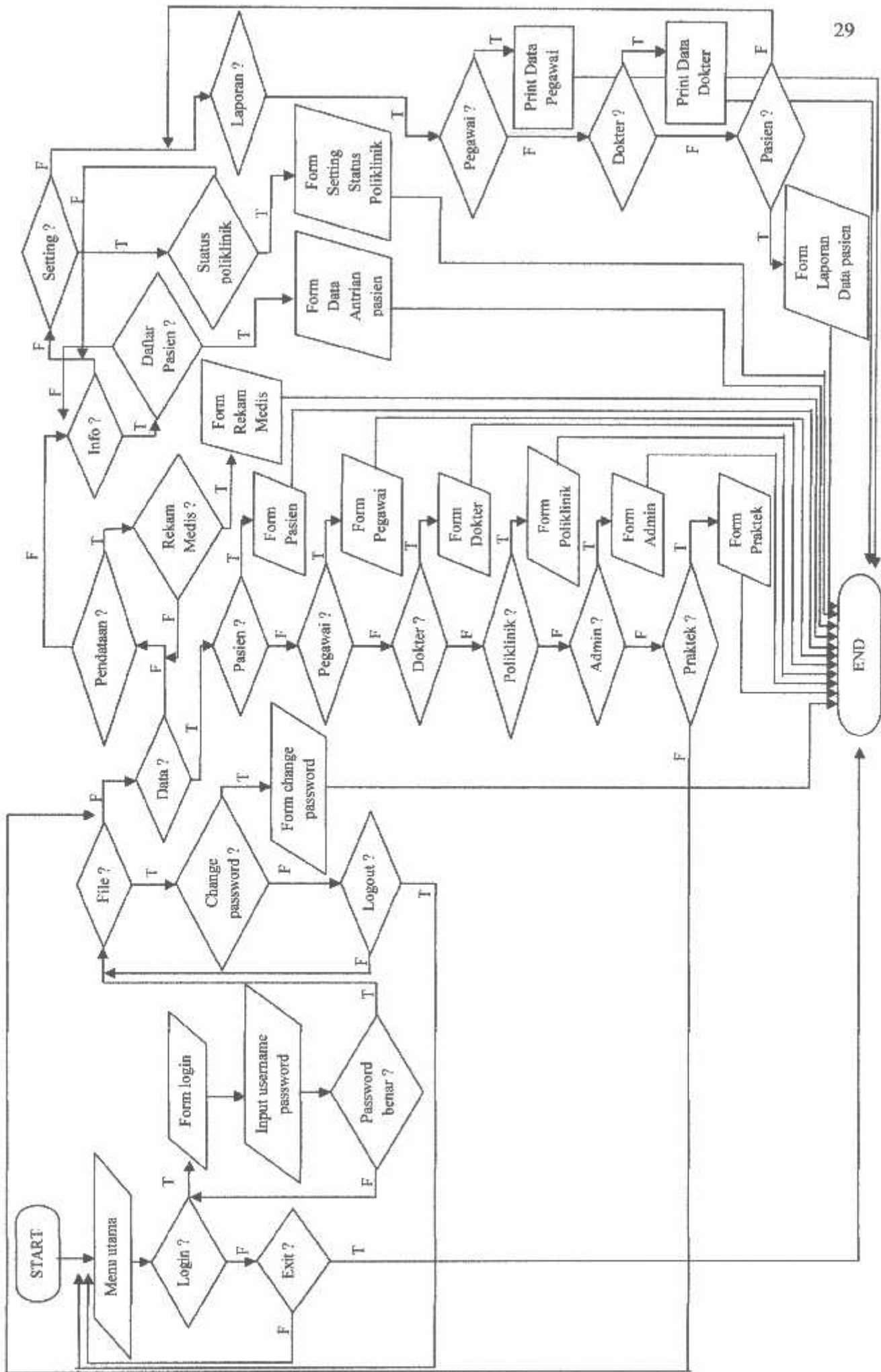
Tabel 3.6 Tabel Poliklinik

Atribut	Tipe Data	Deskripsi
Id_Poli	Int 11	ID poliklinik
Nama_Poli	Char 30	Nama poliklinik
Status_Poli	Char 5	Status True/False
Keterangan_Poli	Char 20	Keterangan poliklinik
Info_Poli	Char 200	Informasi poliklinik

3.1.5 Diagram Alir Program

3.1.5.1 Diagram Alir Menu Utama

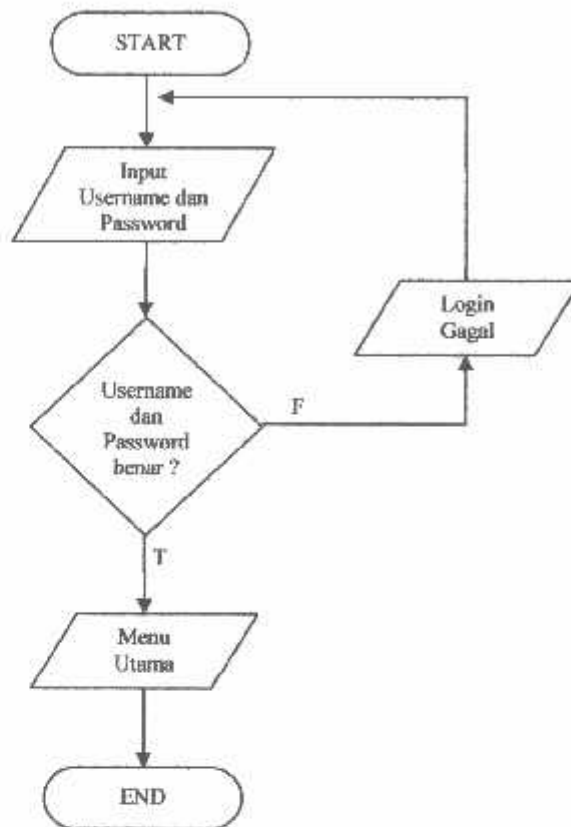
Diagram alir proses menu utama pada Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.13



Gambar 3.13 Diagram Alir Menu Utama

3.1.5.2 Diagram Alir Login

Diagram alir proses *login* pada Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.14

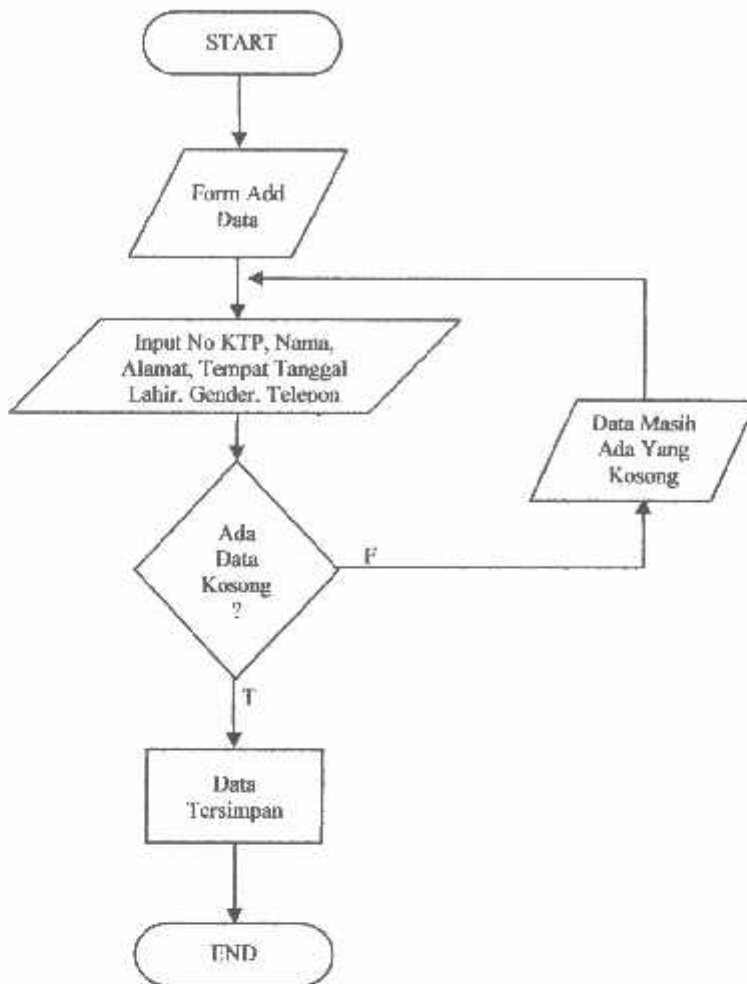


Gambar 3.14 Diagram Alir Login

Algoritma program pada proses *login* adalah admin meng-*input*-kan *username* dan *password*, jika *username* dan *password* benar maka program akan memunculkan menu utama dan semua *toolbar* akan aktif, sedangkan jika *username* dan *password* salah maka akan muncul pesan “Login Gagal”.

3.1.5.3 Diagram Alir Tambah Data Pasien

Diagram alir proses tambah data pasien pada Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.15

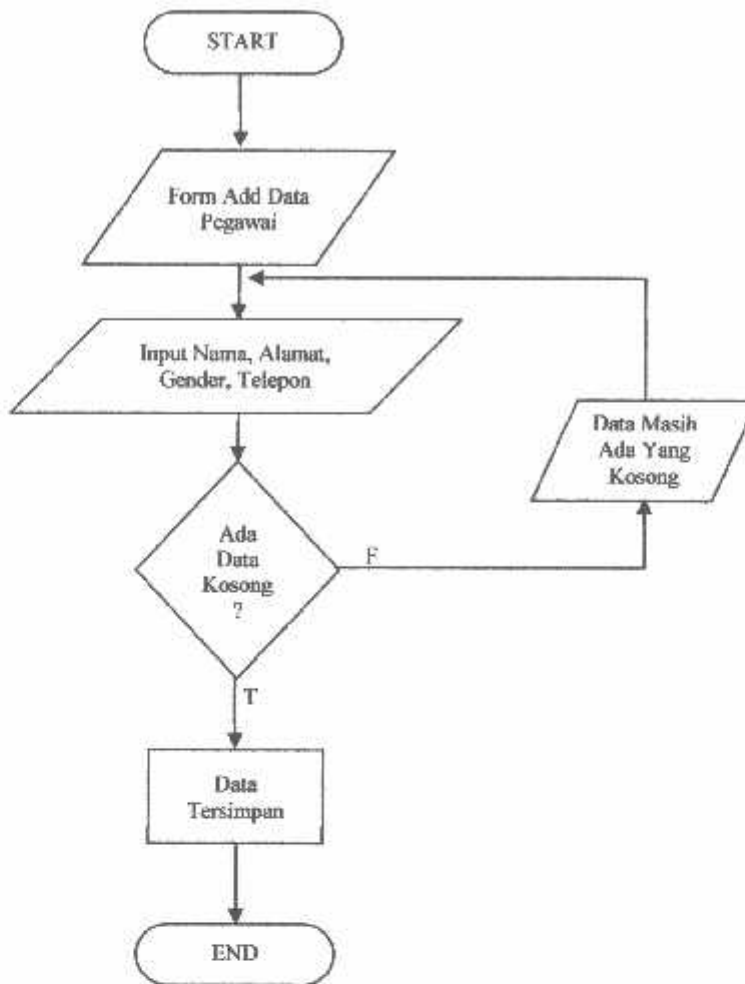


Gambar 3.15 Diagram Alir Tambah Data Pasien

Algoritma program pada proses tambah data admin adalah petugas memasukkan data pasien, yaitu : No KIP, Nama, Alamat, Tempat Tanggal Lahir, Jenis Kelamin, Nomor Telepon. Apabila ada data yang kosong, maka akan muncul pesan “Data Masih Ada Yang Kosong”. Setelah semua data terisi, data akan tersimpan di *database* pada tabel pasien.

3.1.5.4 Diagram Alir Tambah Data Pegawai

Diagram alir proses tambah data pegawai pada Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.16

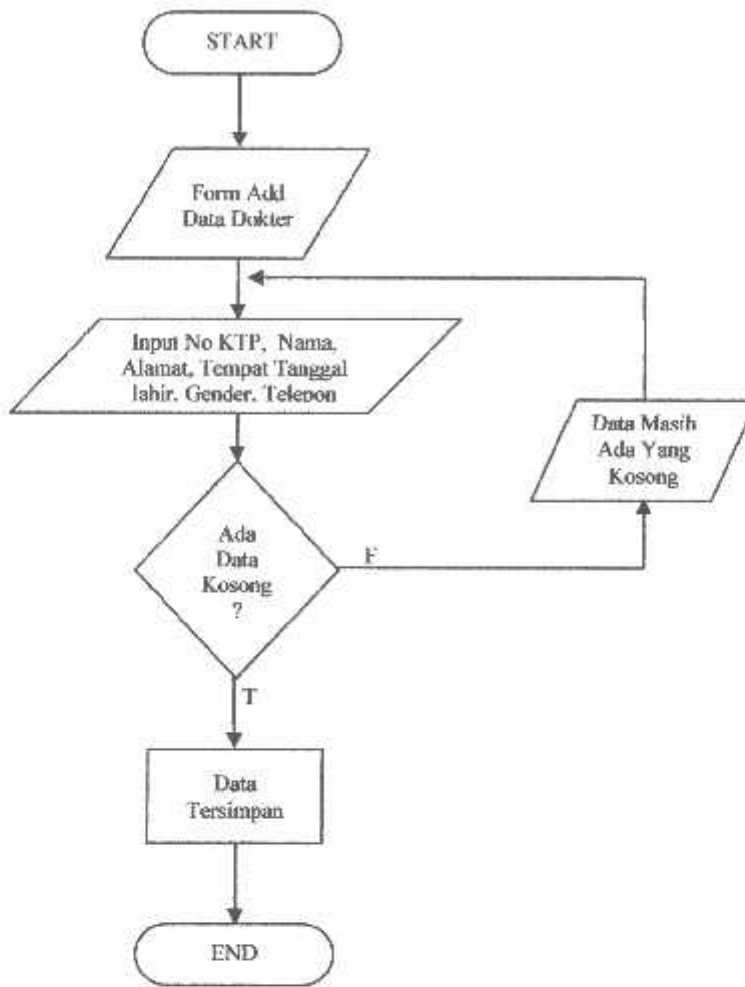


Gambar 3.16 Diagram Alir Tambah Data Pegawai

Algoritma program pada proses tambah data pegawai adalah petugas memasukkan data pegawai, yaitu : Nama, Alamat, Jenis Kelamin, Nomor Telepon. Apabila ada data yang kosong, maka akan muncul pesan “Data Masih Ada Yang Kosong”. Setelah semua data terisi, data akan tersimpan di *database* pada tabel pegawai.

3.1.5.5 Diagram Alir Tambah Data Dokter

Diagram alir proses tambah data dokter pada Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.17

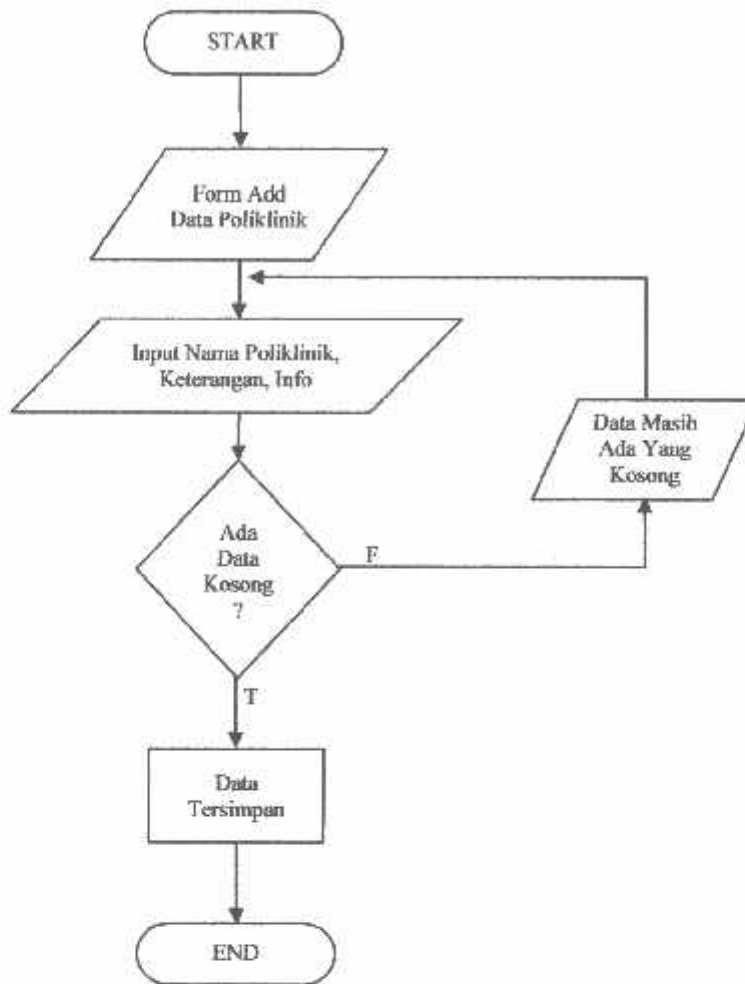


Gambar 3.17 Diagram Alir Tambah Data Dokter

Algoritma program pada proses tambah data dokter adalah petugas memasukkan data dokter, yaitu : No KTP, Nama, Alamat, Tempat Tanggal Lahir, Jenis Kelamin, Nomor Telepon. Apabila ada data yang kosong, maka akan muncul pesan “Data Masih Ada Yang Kosong”. Setelah semua data terisi, data akan tersimpan di *database* pada tabel dokter.

3.1.5.6 Diagram Alir Tambah Data Poliklinik

Diagram alir proses tambah data poliklinik pada Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.18

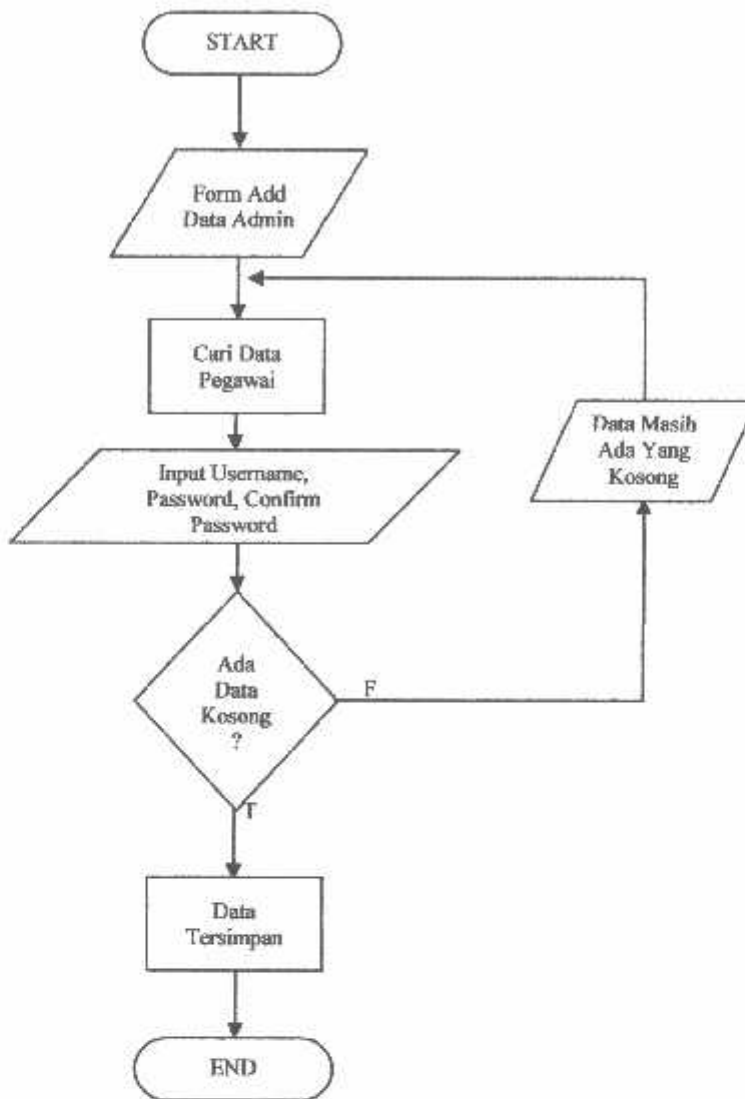


Gambar 3.18 Diagram Alir Tambah Data Poliklinik

Algoritma program pada proses tambah data poliklinik adalah petugas memasukkan data poliklinik, yaitu : Nama Poliklinik, Keterangan, Info. Apabila ada data yang kosong, maka akan muncul pesan “Data Masih Ada Yang Kosong”. Setelah semua data terisi, data akan tersimpan di *database* pada tabel poliklinik.

3.1.5.7 Diagram Alir Tambah Data Admin

Diagram alir proses tambah data admin pada Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.19

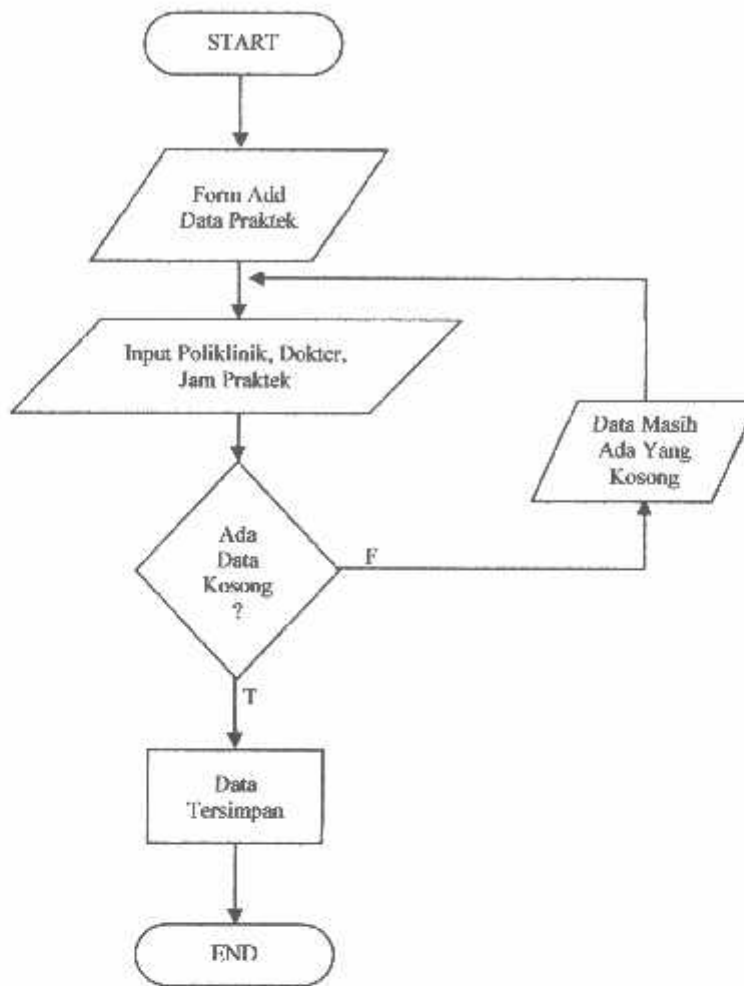


Gambar 3.19 Diagram Alir Tambah Data Admin

Algoritma program pada proses tambah data admin adalah petugas mencari data pegawai yang akan menjadi admin, setelah itu memasukkan *username*, *password*, *confirm password*. Apabila ada data yang kosong, maka akan muncul pesan “Data Masih Ada Yang Kosong”. Setelah semua data terisi, data akan tersimpan di *database* pada tabel admin.

3.1.5.8 Diagram Alir Tambah Data Praktek

Diagram alir proses tambah data praktek pada Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.20

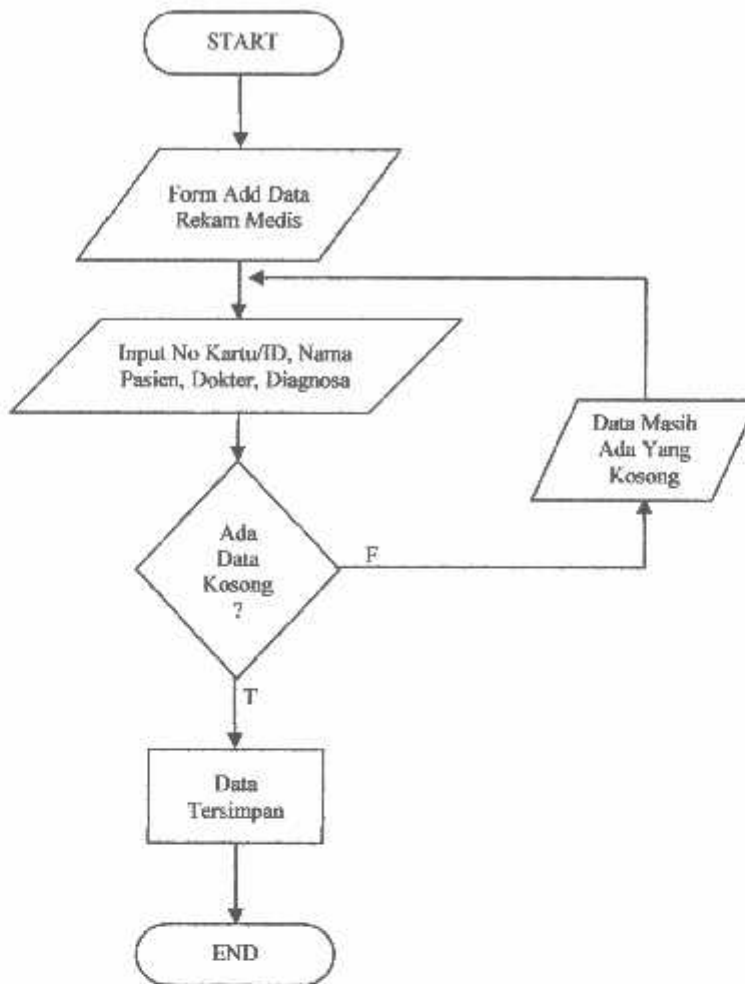


Gambar 3.20 Diagram Alir Tambah Data Praktek.

Algoritma program pada proses tambah data praktek adalah petugas memasukkan data praktek, yaitu : Nama Poliklinik, Dokter, Jam Praktek. Apabila ada data yang kosong, maka akan muncul pesan “Data Masih Ada Yang Kosong”. Setelah semua data terisi, data akan tersimpan di *database* pada tabel praktek.

3.1.5.9 Diagram Alir Tambah Data Rekam Medis

Diagram alir proses tambah data rekam medis pada Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.21

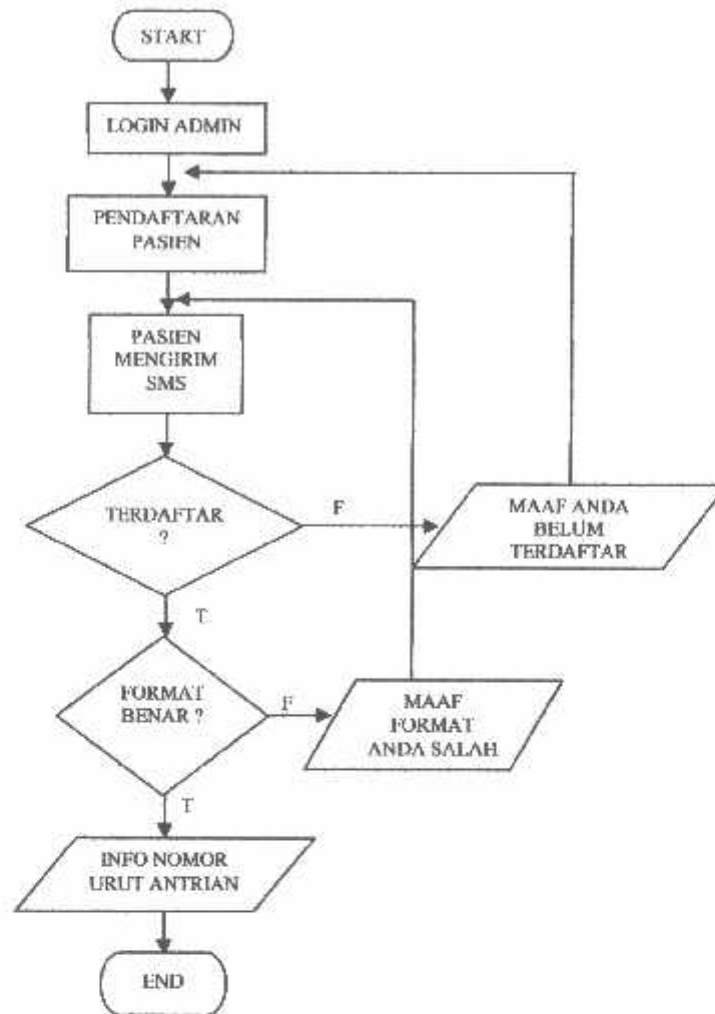


Gambar 3.21 Diagram Alir Tambah Rekam Medis

Algoritma program pada proses tambah data rekam medis adalah petugas memasukkan data rekam medis, yaitu : No Kartu/ID, Nama Pasien, Dokter, Diagnosa. Apabila ada data yang kosong, maka akan muncul pesan “Data Masih Ada Yang Kosong”. Setelah semua data terisi, data akan tersimpan di *database* pada tabel *rekam_medis*.

3.1.5.10 Diagram Alir User

Diagram alir user pada Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.22



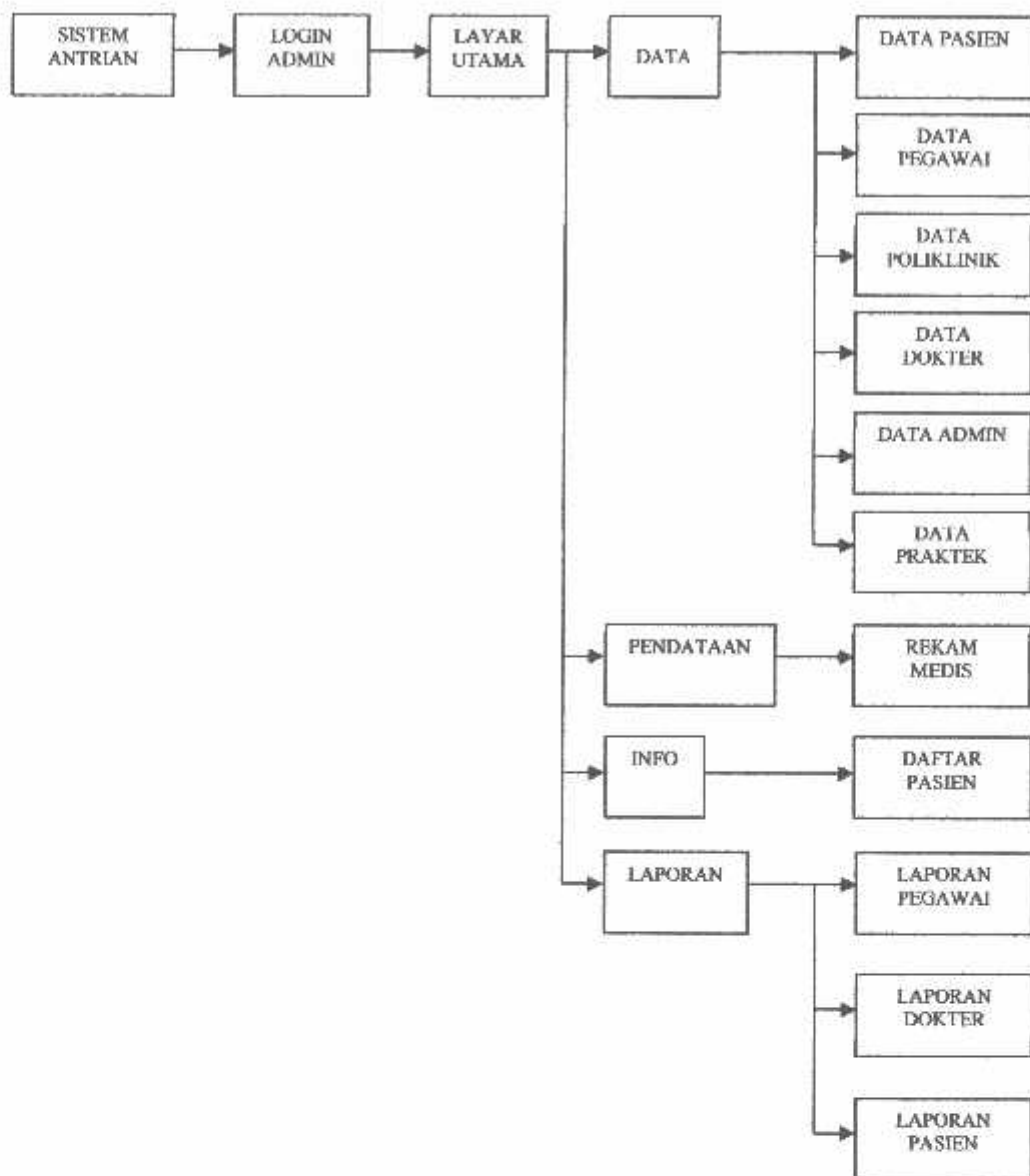
Gambar 3.22 Diagram Alir User

Program hanya dapat digunakan apabila admin sudah melakukan login, setelah itu petugas melakukan pendaftaran. Pasien mengirim SMS ke nomor center poliklinik, apabila pasien terdaftar dan format SMS benar, pasien akan mendapat nomor urut antrian untuk memerikasakan kesehatannya.

3.1.6 Desain Rancangan

3.1.6.1 Struktur Menu

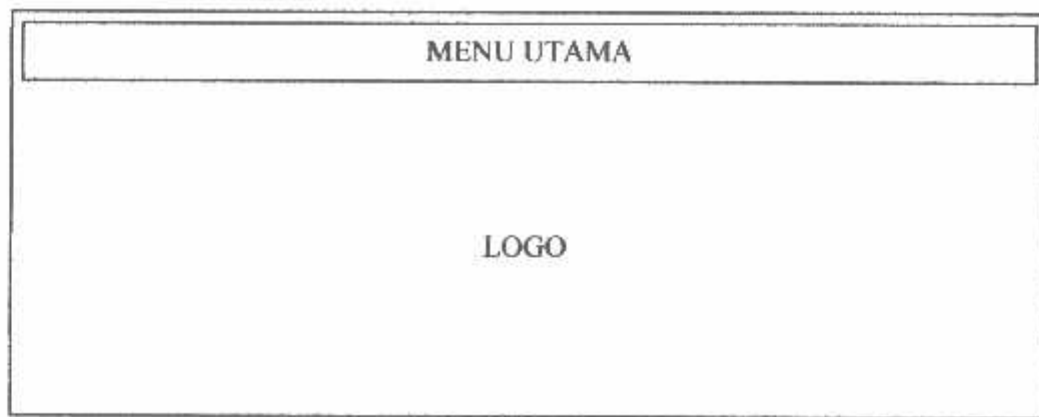
Struktur menu pada Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.23



Gambar 3.23 Struktur Menu

3.1.6.2 Rancangan Desain Menu Utama

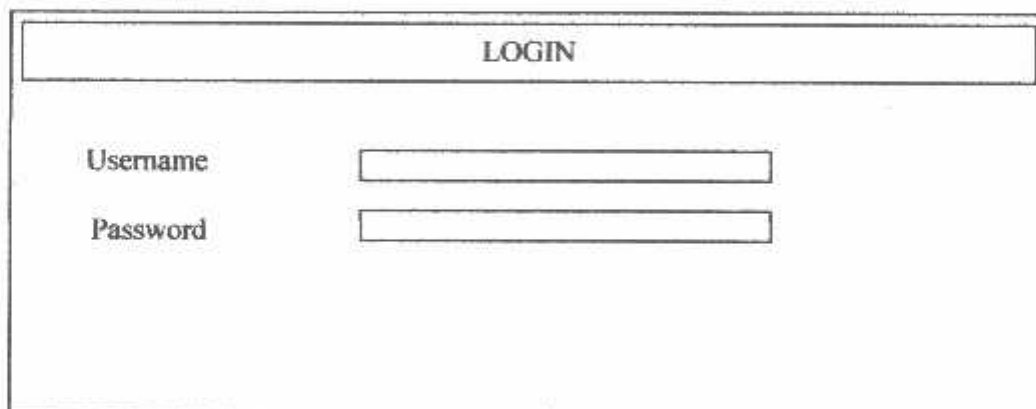
Rancangan tampilan pada menu utama Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* yang terdiri dari menu dan logo dapat dilihat pada Gambar 3.24



Gambar 3.24 Rancangan Desain Menu Utama

3.1.6.3 Rancangan Desain Menu *Login*

Rancangan tampilan pada menu *login* Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* yang terdiri dari *username* dan *password* untuk bisa masuk ke menu utama dapat dilihat pada Gambar 3.25

A rectangular box representing the login menu design. At the top, there is a horizontal bar containing the text "LOGIN". Below this bar, there are two input fields. The first field is labeled "Username" and the second field is labeled "Password".

Gambar 3.25 Rancangan Desain Menu *Login*

3.1.6.4 Rancangan Desain Menu *File*

Rancangan tampilan pada menu *login* Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* yang terdiri dari *username* dan *password* untuk bisa masuk ke menu utama dapat dilihat pada Gambar 3.26

MENU FILE
Login
Logout
Change Password
Exit

Gambar 3.26 Rancangan Desain Menu File

3.1.6.5 Rancangan Desain Menu Change Password

Rancangan tampilan pada menu *change password* Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* yang terdiri dari ID, Nama Pegawai, *username*, *old password*, dan *new password* dapat dilihat pada Gambar 3.27

Change Password	
Data Anda	
ID	<input type="text"/>
Nama Pegawai	<input type="text"/>
Data Akses	
Username	<input type="text"/>
Old Password	<input type="text"/>
New Password	<input type="text"/>
<input type="button" value="Change"/>	<input type="button" value="Close"/>

Gambar 3.27 Rancangan Desain Menu Change Password

3.1.6.6 Rancangan Desain Menu Data

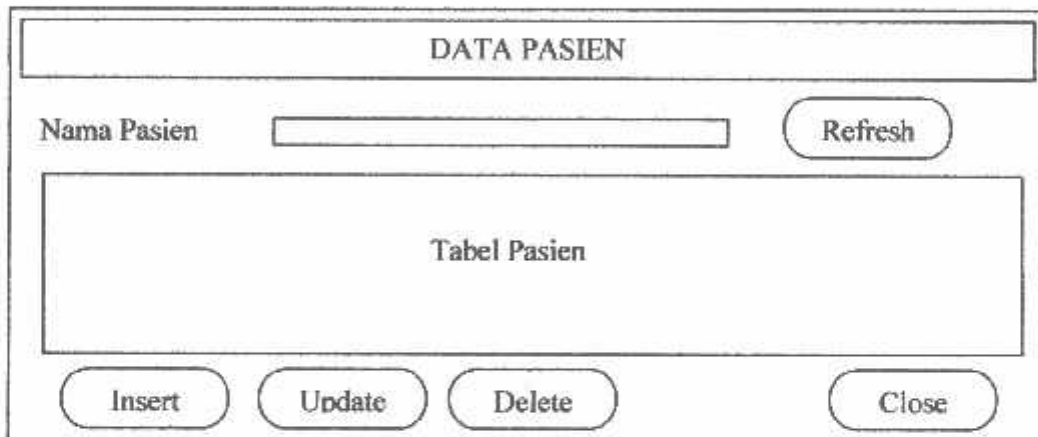
Rancangan tampilan pada menu data Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* yang terdiri dari pasien, pegawai, dokter, poliklinik, admin dan praktek dapat dilihat pada Gambar 3.28



Gambar 3.28 Rancangan Desain Menu Data

3.1.6.7 Rancangan Desain Menu Data Pasien

Rancangan tampilan pada menu data pasien Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.29



Gambar 3.29 Rancangan Desain Menu Data Pasien

3.1.6.8 Rancangan Desain Menu Data Pegawai

Rancangan tampilan pada menu data pegawai Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.30

The image shows a software interface for managing employee data. At the top, there is a title bar labeled "DATA PEGAWAI". Below the title bar, on the left, is a text input field labeled "Nama Pegawai". To the right of this field is a rounded button labeled "Refresh". In the center of the interface is a large rectangular area labeled "Tabel Pegawai", which is currently empty. At the bottom of the interface, there is a horizontal toolbar containing four rounded buttons: "Insert", "Update", "Delete", and "Close".

Gambar 3.30 Rancangan Desain Menu Data Pegawai

3.1.6.9 Rancangan Desain Menu Data Dokter

Rancangan tampilan pada menu data dokter Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.31

The image shows a software interface for managing doctor data. At the top, there is a title bar labeled "DATA DOKTER". Below the title bar, on the left, is a text input field labeled "Nama Dokter". To the right of this field is a rounded button labeled "Refresh". In the center of the interface is a large rectangular area labeled "Tabel Dokter", which is currently empty. At the bottom of the interface, there is a horizontal toolbar containing four rounded buttons: "Insert", "Update", "Delete", and "Close".

Gambar 3.31 Rancangan Desain Menu Data Dokter

3.1.6.10 Rancangan Desain Menu Data Poliklinik

Rancangan tampilan pada menu data poliklinik Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.32

The image shows a software interface window titled "DATA POLIKLINIK". At the top, there is a header bar with the title. Below the header, on the left, is a text label "Nama Poliklinik" followed by a rectangular input field. To the right of the input field is a rounded button labeled "Refresh". In the center of the window is a large, empty rectangular area labeled "Tabel Poliklinik". At the bottom of the window, there are four rounded buttons arranged horizontally: "Insert", "Update", "Delete", and "Close".

Gambar 3.32 Rancangan Desain Menu Data Poliklinik

3.1.6.11 Rancangan Desain Menu Data Admin

Rancangan tampilan pada menu data admin Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.33

The image shows a software interface window titled "DATA ADMIN". At the top, there is a header bar with the title. Below the header, on the left, is a text label "Nama Admin" followed by a rectangular input field. To the right of the input field is a rounded button labeled "Refresh". In the center of the window is a large, empty rectangular area labeled "Tabel Admin". At the bottom of the window, there are two rounded buttons arranged horizontally: "Insert" on the left and "Close" on the right.

Gambar 3.33 Rancangan Desain Menu Data Admin

3.1.6.12 Rancangan Desain Menu Data Praktek

Rancangan tampilan pada menu data praktek Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.34

The image shows a window titled "DATA PRAKTEK". At the top, there is a header bar with the title. Below the header, on the left, is the label "Poliklinik" followed by a text input field. To the right of the input field is a rounded button labeled "Refresh". In the center of the window is a large rectangular area labeled "Tabel Praktek", which is currently empty. At the bottom of the window, there is a row of four rounded buttons: "Insert", "Update", "Delete", and "Close".

Gambar 3.34 Rancangan Desain Menu Data Praktek

3.1.6.13 Rancangan Desain Menu Rekam Medis

Rancangan tampilan pada menu rekam medis Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.35

The image shows a window titled "REKAM DATA MEDIS". At the top, there is a header bar with the title. Below the header, on the left, is the label "Nama Pasien" followed by a text input field. To the right of the input field is a rounded button labeled "Refresh". In the center of the window is a large rectangular area labeled "Tabel Rekam_Medis", which is currently empty. At the bottom of the window, there is a row of three rounded buttons: "Insert", "Update", and "Close".

Gambar 3.35 Rancangan Desain Menu Rekam Medis

3.1.6.14 Rancangan Desain Menu Data Antrian Pasien

Rancangan tampilan pada menu data antrian pasien Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.36

DATA ANTRIAN PASIEN	
Tanggal	<input type="text"/>
Nama Pasien	<input type="text"/>
Poliklinik	<input type="text"/>
	<input type="button" value="Print"/> <input type="button" value="Find"/> <input type="button" value="Refresh"/>
Tabel Pendaftaran	
<input type="button" value="Update"/> <input type="button" value="Close"/>	

Gambar 3.36 Rancangan Desain Menu Data Antrian Pasien

3.1.6.15 Rancangan Desain Menu Setting Status Poliklinik

Rancangan tampilan pada menu setting status poliklinik Perancangan Sistem Antrian Pasien Poliklinik Berbasis *SMS Gateway* dapat dilihat pada Gambar 3.37

SETTING STATUS POLIKLINIK	
Poliklinik	<input type="text"/>
	<input type="button" value="Refresh"/>
Tabel Poliklinik	
Pilihan	<input type="text"/>
Keterangan	<input type="text"/>
<input type="button" value="Set"/> <input type="button" value="Close"/>	

Gambar 3.37 Rancangan Desain Menu Setting Status Poliklinik

BAB IV

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini membahas tentang hasil pengujian dan pembahasan program yang meliputi: hasil implementasi *database*, hasil implementasi sistem keseluruhan, pengujian sistem dan analisa data.

4.1 Hasil Implementasi Database

Database digunakan untuk menyimpan semua data yang berkaitan dengan pasien, dokter, dan admin (petugas). Semua data dibagi menjadi beberapa bagian yang disusun dalam tabel, sehingga hasil dari pengisian data dapat dilihat pada masing-masing tabel yang telah dibuat.

4.1.1 Implementasi Tabel Admin.

id_admin	id_pegawai	username	password	role
1	1	rizqi	sikembar	super admin
3	3	amal	amal	admin

Gambar 4.1 Implementasi Tabel Admin

4.1.2 Implementasi Tabel Dokter.

id_dokter	no_ktp_dokter	nama_dokter	alamat_dokter	tempat_lahir_dokter	tanggal_lahir_dokter	jk_dokter	telp_dokter	status_dokter
1	344343434	Dr. Rudi Selianto Spg	Jl. Kemuning Raya 07 Malang	Malang	10/1/1980	L	0341898323	true
2	08328372837	Prof. Dr. Isihartoro Soekardi,	Jl. Raya Miranjun Blambangan 08 Malang	Surabaya	8/4/1970	L	803278362	true
3	2376437612381	DR. Iike Sumantri, SpM	Jl. Kidang Merak	Surabaya	3/8/1990	P	803278362	true

Gambar 4.2 Implementasi Tabel Dokter

4.1.3 Implementasi Tabel Pasien.

id_pasien	no_ktp_pasien	nama_pasien	alamat_pasien	tempat_lahir_pasien	tanggal_lahir_pasien	jk_pasien	telp_pasien	status_pasien
1234	09090909090	Agung Dwi Prastyo	Tawangsari	Malang	10/1/2012	L	+628191452264	true
13544	6566766	Amalina	malang	malang	10/1/2012	P	+6285646364745	true
152917	35583493894839	Andrian Sucahyo	J. Abdul Gani Batu	Malang	6/1/1990	L	0341-902323	true
153549	4783778787878	Ferdita	J. Kalurung Malang	Malang	10/1/1990	P	+6281233403302	true
19355	378787878	Rizqi Euzhi Prastyo	J. Lawang	Malang	10/1/2012	L	+6285746728872	true
3214	766767868767867	Guntur Yudha Pratama	Gresik	Malang	10/1/2012	L	+6287759991333	true
92231	85739875370	Dini Eka Mulvani	Tawangsari	Malang	10/10/2012	P	+6285756049327	true

Gambar 4.3 Implementasi Tabel Pasien

4.1.4 Implementasi Tabel Pegawai.

id_pegawai	nama_pegawai	alamat_pegawai	jk_pegawai	telp_pegawai	status_pegawai
1	Hantoro Adi Sucahyo	Jl. Abdul Gani 09A batu	L	083834728214	true
3	Amalina Mahmudah	Malang	P	+6341428183	true
4	amalina mahmudah	lawang	P	085646364745	false

Gambar 4.4 Implementasi Tabel Pegawai

4.1.5 Implementasi Tabel Pendaftaran.

id_janji	tgl_janji	jam_janji	id_pasien	id_praktek	urutan	status_pendaftaran
1	10/27/2012		153549	2	1	Selesai
2	12/25/2012		135544	2	1	Selesai
3	1/7/2013		19355	2	1	mengantri
4	1/7/2013		92231	2	2	mengantri
5	1/7/2013		9214	2	3	mengantri

Gambar 4.5 Implementasi Tabel Pendaftaran

4.1.6 Implementasi Tabel Poliklinik.

id_poli	nama_poli	status_poli	keterangan_poli	info_poli
1	kandungan	true	Buka	Praktek Hari Rabu-Jumat dari 07.00 - 12.00
2	mata	true	Buka	Praktek Hari Senin-Jumat dari 10.00 - 15.00

Gambar 4.6 Implementasi Tabel Poliklinik

4.1.7 Implementasi Tabel Praktek.

id_praktek	id_poli	id_dokter	jam_praktek	status_praktek	estimasi
2	1	1	7:00:00 AM-12:00:00	true	25200
3	2	2	7:00:00 AM-12:00:00	true	25200
4	2	3	12:00:00 AM-6:00:00	true	43200

Gambar 4.7 Implementasi Tabel Praktek

4.1.8 Implementasi Tabel Rekam Medis.

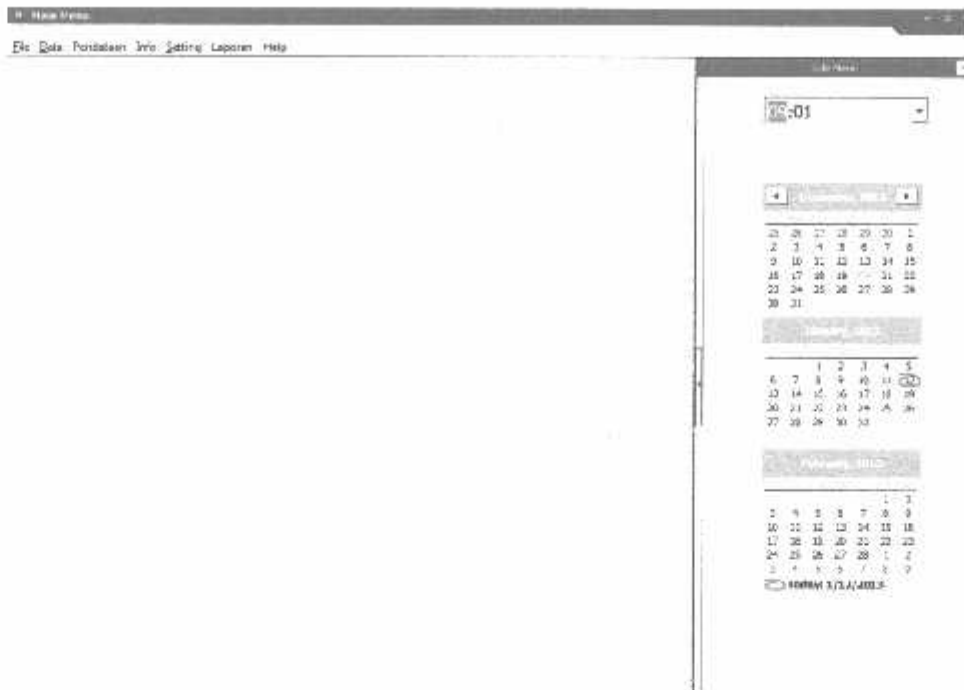
id_rekam	id_pasien	id_dokter	tgl_cek	diagnosa	id_admin
1	152917	1	10/27/2012	Terjadi kontraksi pada dinding rahim	1
2	152917	2	12/8/2012	Kontraksi	1

Gambar 4.8 Implementasi Tabel Rekam Medis

4.2 Implementasi Sistem Keseluruhan

Implementasi sistem keseluruhan perancangan sistem antrian pasien poliklinik berbasis *SMS gateway* adalah sebagai berikut.

4.2.1 Hasil Form Utama.



Gambar 4.9 Form Utama

Form utama muncul pertama kali ketika program dijalankan dan terdiri dari beberapa menu, diantaranya adalah File, Data, Pendaftaran, Info, Setting, Laporan dan Help.

4.2.2 Hasil Form Login.

 The image shows a screenshot of a 'Login' dialog box. The dialog has a title bar with the text 'Login' and standard window control buttons (minimize, maximize, close). Inside the dialog, there is a label 'Login' at the top left. Below it, there are two input fields: 'Username' and 'Password'. At the bottom of the dialog, there are two buttons: 'Login' and 'Cancel'.

Gambar 4.10 Form Login

Form login digunakan untuk proses *login* petugas, apabila petugas belum melakukan proses *login*, maka proses sistem antrian pasien tidak bisa dilakukan, karena sub menu lain dalam Menu Utama selain File - *Login* dan File - *Exit* tidak aktif.

4.2.3 Hasil *Form Change Password*.

Gambar 4.11 *Form Change Password*

Form change password digunakan untuk mengubah *password* admin yang sudah melakukan proses *login*. Menu ini hanya bisa digunakan untuk mengubah *password* admin yang sedang aktif.

4.2.4 Hasil *Form Data Pasien*.

No	No. I.D	Nama	Alamat	Tempat Lahir	Tanggal Lahir	Jk	Foto
152917	35983433694838	Andean Sucotyo	Jl. Abdul Giri Batu	Malang	6/1/1990	L	0341-992323
153549	0703770707070	Ferdia	Jl. Kakurang Malang	Malang	10/1/1990	P	+6201233403302
19365	070767070	Rizq Budi Prastyo	Jl. Lawang	Malang	10/1/2012	L	+6205746720072
01056	654321	Yusuf Dudi Irawan	Malang	Malang	8/1/1985	L	+62011367463
01441	76564342	Anaknu Mahmudah	Malang	Malang	8/1/2003	P	+6295646364745
9840	12345678	Suryo Adi	Malang	Malang	8/1/1985	L	+6295645780577

Gambar 4.12 *Form Data Pasien*

Form data pasien digunakan untuk menampilkan data tentang pasien yang telah terdaftar.

4.2.5 Hasil *Form Add Data Pasien*.

The screenshot shows a window titled "Add Data Pasien". Inside, there is a section labeled "Data" with the following fields:

- ID: 1112
- No. KTP: (empty)
- Nama: (empty)
- Alamat: (empty)
- Tmp / Tgl Lahir: (empty) / 10/ 1/2012
- Gender: Laki Perempuan
- Telp: (empty)

At the bottom of the form are two buttons: "Simpan" and "Batal".

Gambar 4.13 *Form Add Data Pasien*

Form add data pasien digunakan untuk input data pasien baru dengan mengisi identitas pasien baru. Apabila data yang diisikan ada yang kosong maka akan muncul pesan "Data Masih Ada Yang Kosong" dan apabila ID yang diisikan sama, maka akan muncul "Cek Kembali".

4.2.6 Hasil *Form Update Data Pasien*.

The screenshot shows a window titled "Update Data Pasien". Inside, there is a section labeled "Data" with the following fields:

- ID: 1234
- No. KTP: 8888888888
- Nama: Agung Dwi Prastiyo
- Alamat: Tawangsari
- Tmp / Tgl Lahir: Malang / 10/ 1/2012
- Gender: Laki Perempuan
- Telp: +628990452254

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

Gambar 4. 14 *Form Update Data Pasien*

Form update data pasien digunakan untuk memperbarui data pasien yang telah terdaftar jika terdapat perubahan. Perubahan dilakukan dengan cara klik data pasien yang akan diubah – klik update – isikan data sesuai perubahan – klik ubah.

4.2.7 Hasil *Form Data Pegawai.*

The screenshot shows a window titled "Data Pegawai". At the top, there is a search bar labeled "Find" with the text "Nama Pegawai" and a "Refresh" button. Below this is a table with the following data:

ID	Nama	Alamat	Jenis Kelamin	No. Telp
1	Rizki Budhi Pratyo	Jl. Tawangsari 55 Lingsar	L	085746728872
2	dit	dit	L	tdkda343

At the bottom of the window, there are buttons for "Insert", "Update", "Delete", and "Close".

Gambar 4.15 *Form Data Pegawai*

Form data pegawai digunakan untuk menampilkan data pegawai yang terdaftar.

4.2.8 Hasil *Form Add Data Pegawai.*

The screenshot shows a window titled "Add Data Pegawai". It contains a form with the following fields and controls:

- Nama:** A text input field.
- Alamat:** A text input field.
- Gender:** Radio buttons for "Laki" (selected) and "Perempuan".
- Telp:** A text input field.

At the bottom of the form, there are two buttons: "Simpan" (Save) and "Batal" (Cancel).

Gambar 4.16 *Form Add Data Pegawai*

Form add data pegawai digunakan untuk input data pegawai baru dengan mengisi identitas pegawai baru. Apabila data yang diisi ada yang kosong maka akan muncul pesan "Data Masih Ada yang Kosong".

4.2.9 Hasil Form Update Data Pegawai.

Update Data Pegawai

Data

Nama: Heriwo Adi Sucahyo

Alamat: Jl. Abdul Gani 09A batu

Gender: Laki Perempuan

Telp: 083834728214

Buttons: Ubah, Reset, Batal

Gambar 4.17 Form Update Data Pegawai

Form update data pegawai digunakan untuk memperbarui data pegawai yang telah terdaftar jika terdapat perubahan. Perubahan dilakukan dengan cara klik data pegawai yang akan diubah pada *form* data pegawai – klik *update* – isikan data sesuai perubahan – klik *ubah*.

4.2.10 Hasil Form Data Dokter.

Data Dokter

Find: Nama Dokter Refresh

No	No. IDP	Nama	Menit	Tempat Lahir	Tanggal Lahir	M	Telp
1	344343434	Dr. Rudi Saktiwo Spg		Jl. Kemuning Raya 07 Malang	Malang	10/1/1980	L 0341898323
2	08320372807	Prof. Dr. Idris Sodikin		Jl. Raja Mawang Emasbungas 08 Malang	Surabaya	8/4/1970	L 803278362
3	2376437617381	DR. Ika Sunami, SpM		Jl. Kibang Merah	Surabaya	3/8/1990	P 803278362
4	857658576	Azzahra Mahasrah, SST		Malang	Malang	11/1/2012	P 085646364745

Buttons: Insert, Update, Delete, Close

Gambar 4.18 Form Data Dokter

Form data dokter digunakan untuk menampilkan data dokter yang praktek di Rumah Sakit.

4.2.11 Hasil *Form Add Data Dokter*.

The screenshot shows a window titled "Add Data Dokter". Inside, there is a section labeled "Data" with the following fields:

- No. KIP: [Empty text box]
- Nama: [Empty text box]
- Alamat: [Empty text box]
- Tmp / Tgl Lahir: [Empty text box] / 10/ 1/2012 (dropdown menu)
- Gender: Laki Perempuan
- Telp: [Empty text box]

At the bottom of the form are two buttons: "Simpan" and "Batal".

Gambar 4.19 *Form Add Data Dokter*

Form add data dokter digunakan untuk *input* data dokter baru dengan mengisi identitas dokter baru. Apabila data yang diisikan ada yang kosong maka akan muncul pesan "Data Masih Ada yang Kosong".

4.2.12 Hasil *Form Update Data Dokter*.

The screenshot shows a window titled "Update Data Dokter". Inside, there is a section labeled "Data" with the following fields:

- No. KTP: 84433332
- Nama: Dr. Rudi Setianto Spg
- Alamat: Jl. Kemuning Raya 07 Malang
- Tmp / Tgl Lahir: Malang / 10/ 1/1980 (dropdown menu)
- Gender: Laki Perempuan
- Telp: 0341898323

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

Gambar 4.20 *Form Update Data Dokter*

Form update data dokter digunakan untuk memperbarui data dokter yang telah terdaftar jika terdapat perubahan. Perubahan dilakukan dengan cara klik data dokter yang akan diubah pada form data dokter – klik update – isikan data sesuai perubahan – klik ubah.

4.2.13 Hasil *Form Data Poliklinik*.

The screenshot shows a window titled "Data Poliklinik". At the top, there is a "Form" section with a text input field labeled "Nama Poliklinik" and a "Refresh" button. Below this is a table with the following data:

No	Nama	Alamat	Waktu
1	Kambang	Buka	Poltek Han Fisku-jarak dari 07.00 - 12.00
2	anata	buka	Poltek Han Serah-jarak dari 10.00 - 15.00

At the bottom of the window, there are four buttons: "Insert", "Update", "Delete", and "Close".

Gambar 4.21 *Form Data Poliklinik*

Form data poliklinik digunakan untuk menampilkan data poliklinik yang ada di Rumah Sakit.

4.2.14 Hasil *Form Add Data Poliklinik*.

The screenshot shows a window titled "Add Data Poliklinik". It contains a "Data" section with three input fields: "Nama Poliklinik", "Keterangan", and "Info". At the bottom, there are two buttons: "Simpan" (Save) and "Batal" (Cancel).

Gambar 4.22 *Form Add Data Poliklinik*

Form add data poliklinik digunakan untuk *input* data poliklinik baru dengan mengisi identitas poliklinik baru. Apabila data yang diisikan ada yang kosong maka akan muncul pesan “Data Masih Ada yang Kosong”.

4.2.15 Hasil *Form Update Data Poliklinik*.

Gambar 4.23 *Form Update Data Poliklinik*

Form update data poliklinik digunakan untuk memperbarui data poliklinik yang telah terdaftar jika terdapat perubahan. Perubahan dilakukan dengan cara klik data poliklinik yang akan diubah pada form data poliklinik – klik update – isikan data sesuai perubahan – klik ubah.

4.2.16 Hasil *Form Data Admin*.

No.	Nama	Telepon
1	T. Rizqi Budhi Prastya	085745728872

Gambar 4.24 *Form Data Admin*

Form data praktek digunakan untuk menampilkan jadwal praktek dokter pada semua klinik yang meliputi ID dokter, jenis klinik, nama dokter, dan jam praktek.

4.2.19 Hasil *Form Add Data Praktek*.

ID	Poliklinik	Nama	Jam Praktek
2	kandungan	Dr. Rudi Setianto Spg	7:00:00 AM-12:00:00
3	mata	Prof. Dr. Intanora Soekardi	7:00:00 AM-12:00:00
4	inata	DRI. Ikke Sumantri, SpM	12:00:00 AM-6:00:00
5	kandungan	Amalina Mahmudah, SST	12:10:00 PM-6:10:00

Gambar 4.27 *Form Add Data Praktek*

Form add data praktek digunakan untuk *input* data jadwal praktek baru dengan mengisi data ID, poliklinik, nama dokter serta jam praktek yang baru. Apabila data yang diisikan ada yang kosong maka akan muncul pesan “Data Masih Ada yang Kosong”.

4.2.20 Hasil *Form Rekam Data Medis*.

No	Tanggal	ID Pasien	Nama	Alamat Pasien	No Dokter	Dokter	Diagnosa
3	12/16/2012	15235	Rizal Budi Prasetyo	J. Lamong	2	Prof. Dr. Intanora Soekardi	aman tak ada
2	12/8/2012	152317	Andean Sucalya	J. Abdi Dam Batu	2	Prof. Dr. Intanora Soekardi	amnesia
1	10/27/2012	152317	Andean Sucalya	J. Abdi Dam Batu	1	Dr. Rudi Setianto Spg	Tersedih kontrol di pada dinding kanan / l

Gambar 4.28 *Form Rekam Data Medis*

Form update data rekam medis digunakan untuk memperbarui data medis pasien yang telah terdaftar jika melakukan pemeriksaan ulang / *follow up*. Perubahan hanya bisa dilakukan pada diagnosa. Perubahan dilakukan dengan cara klik pendataan rekam data medis yang akan diubah pada form pendataan rekam data medis – klik update – isikan data sesuai perubahan – klik ubah.

4.2.23 Hasil *Form Data Antrian Pasien*.

The screenshot shows a window titled "Data Antrian Pasien". It features a search bar with a date dropdown set to "17/7/2013". Below the search bar are radio buttons for "Nama Pasien" and "Poliklinik", followed by input fields and "Find" and "Refresh" buttons. A "Print" button is also present. The main area contains a table with the following data:

No	Tanggal	ID/No. Rekam Medis	Nama	Alamat	Poliklinik	Dokter	Daftar	Status
3	17/7/2013	13390	Rizki Brutha Poesito	Jl. Lawang	Kandungan	Dr. Rudi Sekanto Sag	1	menunggu
4	17/7/2013	92251	Dini Eka Mujani	Tawangmali	Kandungan	Dr. Rudi Sekanto Sag	2	menunggu
5	17/7/2013	8214	Guntur Yuchia Pratama	Seoak	Kandungan	Dr. Rudi Sekanto Sag	3	menunggu

At the bottom of the window are "Insert", "Update", and "Close" buttons.

Gambar 4.31 *Form Data Antrian Pasien*

Form data antrian pasien digunakan untuk menampilkan data antrian pasien pada hari tersebut.

4.2.24 Hasil *Form Add Pendaftaran Pasien*.

The screenshot shows a window titled "Add Pendaftaran Pasien". It contains several input fields: "No. Kartu/ID" with a "Find" button, "Nama Pasien", "Untuk Praktek", "Poliklinik" with a "Search" button, and a "Tanggal" dropdown set to "17/7/2013". Below these is a large empty text area. At the bottom, there are fields for "Praktek yang dipilih" (with sub-fields for "Id Praktek", "Poliklinik", and "Jam Praktek") and "No. Antrian Anda" (with a sub-field for "Estimasi Waktu").

Gambar 4.32 *Form Add Pendaftaran Pasien*

Form add pendaftaran pasien digunakan untuk menambahkan data identitas pasien yang mendaftar untuk diperiksa di poliklinik tertentu pada hari tertentu.

4.2.25 Hasil *Form Setting Status Poliklinik.*

ID	Nama	Kategori
1	Bandung	Buka
2	Mea	Buka

Gambar 4.33 *Form Setting Status Poliklinik*

Form setting status poliklinik digunakan untuk menampilkan status dari suatu klinik di Rumah Sakit, apakah buka atau tutup pada hari tersebut.

4.2.26 Hasil Laporan Data Pegawai.

Laporan Data Pegawai					
ID	Nama	Alamat	JK	Telep	Status
1	Hartono Adi Sucatyo	Jl. Abdul Gani 024 Batu	L	000034720214	true
3	Amelia Manmanan	Malang	F	+621425188	true

Gambar 4.34 *Laporan Data Pegawai*

Laporan data pegawai digunakan untuk menampilkan data pegawai yang bekerja di Rumah Sakit.

4.2.27 Hasil Laporan Data Dokter.

ID	NTP	Nama	Alamat	JK	Telp	Tempat Lahir	Tgl Lahir	Status
1	000045434	Dr. Rudi Setanto Spg	Jl. Pemuda Raya 07 Malang	L	324-150033	Malang	18/11/1955	Tua
2	00033272837	Prof. Dr. Istanto Soekand	Jl. Raya Mabarip, Mangrove, 05 45	P	85579570	Surabaya	04/1973	Tua
3	2378437812081	Dr. Ika Sutarna, Spk	J. Kidang Mera	P	83327630	Surabaya	26/1880	Tua

Gambar 4.35 Laporan Data Dokter

Laporan data dokter digunakan untuk menampilkan identitas dokter yang praktek di Rumah Sakit.

4.2.28 Hasil Form Laporan Data Pasien.

Laporan Data Pasien

Find

All

Nama Pasien

ID	Tanggal	Pemerik. P. rawat	Namael. Pasien	IL. Dokter	Dokter	Diagnosa
2	12/8/2012	152917	Andrian Sucahyo	Jl. Abdul Gani Basu	2 Prof. Dr. Istanto Soekand	gagal ginjal
1	18/27/2012	152917	Andrian Sucahyo	Jl. Abdul Gani Basu	1 Dr. Rudi Setanto Spg	Tersedih kometeksi pada dinding rebar

Gambar 4.36 Form Laporan Data Pasien

4.3 Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui fungsi *button* pada aplikasi dan diujikan di *operating system* yang berbeda, yaitu *Windows XP*, *Windows Vista* dan *Windows 7*, hasil pengujian dapat dilihat pada tabel 4.1.

Tabel 4.1 Tabel Pengujian Sistem

No	Pengujian	Windows XP	Windows Vista	Windows 7
1.	Insert	√	√	√
2.	Update	√	√	√
3.	Delete	√	√	√
4.	Refresh	√	√	√
5.	Close	√	√	√
6.	Print	√	√	√

BAB V
PENUTUP

5.1 Kesimpulan

Dari hasil perancangan serta implementasi dan pengujian aplikasi Sistem Antrian Pasien Poliklinik Berbasis SMS *Gateway*, maka dapat dibuat kesimpulan sebagai berikut:

1. Penggunaan Sistem Antrian Berbasis SMS *Gateway* untuk pemesanan nomor urut antrian pasien poliklinik lebih mudah dan tidak harus menunggu lama untuk memeriksakan kesehatannya,
2. Sistem Antrian Berbasis SMS *Gateway* menghasilkan informasi jadwal praktek dokter langganan pasien.
3. Dari hasil pengujian dapat diambil kesimpulan sistem antrian pasien poliklinik berbasis SMS *Gateway* ini dapat berjalan dengan sempurna pada *operating system* berbasis *windows*.

5.2 Saran

1. Untuk pendaftaran pasien baru, disarankan menggunakan menu XML, sehingga pasien lebih mudah untuk mendapatkan nomor urut antrian.
2. Pengembangan lebih lanjut dapat dilakukan dengan menambahkan sistem *online*, sehingga pasien bisa mendaftar melalui internet.

DAFTAR PUSTAKA

- Andi, 2000. *Pemrograman Borland Delphi 7*. Madiun : Andi dan Madcoms.
- Ariza Novianti dan Ami Fauziah. 2009. *Sistem Informasi Sekolah Dasar Berbasis Sms*.(Online), <http://journal.uii.ac.id/index.php/Snati/article/viewFile/949/912>), diakses tanggal 2 Desember 2012.
- Hillier, Frederick. S dan Lieberman, Gerald. I. 1980. *Introduction to Operations Research*. Holden Day, Inc. San Francisco.
- Kadir, Abdul. 1998. *Konsep & Tuntutan Praktis Basis Data*. Yogyakarta : Andi Yogyakarta.
- Komputer, Wahana. 2010. *Panduan Belajar MySQL Database Server*. Jakarta: Mediakita.
- Rosa, A.S., Shalahuddin, M. 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula.
- Setiawan, Sandi. 1991. *Simulasi*. ANDI OFFSET. Yogyakarta.
- Siagian, P. 1987. *Penelitian Operasional : Teori dan Praktek*. Universitas Indonesia Press. Jakarta.
- Subagyo, Pangestu, dkk. 2000. *Dasar-Dasar Operations Research*. BPPFE. Yogyakarta.
- http://id.wikipedia.org/wiki/Telepon_genggam, tanggal akses : 21 Januari 2013.
- <http://www.gandhionline.web.id/index.php/home/viewpdf?id=38>, tanggal akses : 2 Oktober 2012.
-

LAMPIRAN I

Lampiran 1 : Surat Pernyataan

SURAT PERNYATAAN

Nama Mahasiswa : Rizqi Budhi Prastiyo
Tempat / tanggal Lahir : Malang, 13 April 1988
NIM : 11.18.902

Menyatakan bahwa karya skripsi saya yang berjudul :

PERANCANGAN SISTEM ANTRIAN PASIEN POLIKLINIK BERBASIS SMS GATEWAY

Adalah bukan merupakan karya tulis orang lain, baik sebagian maupun keseluruhan, kecuali dalam bentuk kutipan yang kami sebutkan sumbernya. Demikian surat pernyataan ini kami buat dengan sebenar-benarnya dan apabila pernyataan ini tidak benar, kami bersedia mendapatkan sanksi akademis.

Malang, 13 April 2013

Yang Menyatakan,










Rizqi Budhi Prastiyo

LAMPIRAN II

Lampiran 2 : Formulir Bimbingan Skripsi 1

FORMULIR BIMBINGAN SKRIPSI

Nama : RIZQI BUDHI PRASTIYO
NIM : 1118902
Masa Bimbingan : 17 Oktober 2012 - 17 April 2013
Judul Skripsi : PERANCANGAN SISTEM ANTRIAN PASIEN
POLIKLINIK BERBASIS SMS GATEWAY

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1	20/10/2012	BAB I, II, DAN III	
2	17/11/2012	REVISI BAB II, DAN III	
3	15/12/2012	BAB IV, V, DAN REVISI PROGRAM	
4	05/01/2013	REVISI BAB IV, V, DAN PROGRAM	
5	26/01/2013	ACC SEMINAR HASIL	
6	14/02/2013	ACC LAPORAN	
7	15/02/2013	ACC KOMPRE	
8			
9			
10			

Malang, 13 April 2013

Dosen Pembimbing,



Yoseph Dedy Irawan, ST, MT.

NIP. 19740416 200501 1 002


FORMULIR BIMBINGAN SKRIPSI

Nama : RIZQI BUDHI PRASTIYO
NIM : 1118902
Masa Bimbingan : 17 Oktober 2012 - 17 April 2013
Judul Skripsi : PERANCANGAN SISTEM ANTRIAN PASIEN
POLIKLINIK BERBASIS SMS GATEWAY

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1	06/10/2012	PERSIAPAN SEMPRO (BAB I, II, DAN III)	
2	07/10/2012	ACC SEMPRO (BAB I, II, DAN III)	
3	26/01/2013	ACC SEMINAR HASIL	
4	15/02/2013	PERBAIKI DFD, DAN PROGRAM	
5	16/02/2013	BAB IV : TAMBAHI PENGUJIAN BROWSER, BAB V : PERBAIKI KESIMPULAN & SARAN	
6	16/02/2013	ACC LAPORAN SKRIPSI	
7	18/02/2013	ACC PROGRAM	
8			
9			
10			

Malang, 13 April 2013

Dosen Pembimbing,


Suryo Adi Wibowo, ST.

NIP.P.1031000438

LAMPIRAN III

Lampiran 3 : Berita Acara Ujian Skripsi



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Rizqi Budhi Prastiyo
NIM : 1118902
Jurusan : Teknik Informatika S-1
Judul : Perancangan Sistem Antrian Pasien Poliklinik Berbasis SMS Gateway

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada:

Hari : Selasa
Tanggal : 19 Februari 2013
Nilai : 85.70 (A)

Panitia Ujian Skripsi :


Ketua Majelis Penguji


Joseph Dedy Irawan, ST, MT

NIP.19740416 200501 1 002


Anggota Penguji :

Penguji Pertama


Ali Mahmudi, BEng, PhD

NIP.P. 1031000429

Penguji Kedua


Nurlaily Vendyansyah, ST

7/2013
15

LAMPIRAN IV

Lampiran 4 : Formulir Perbaikan Skripsi



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo Km. 2 Malang

FORMULIR PERBAIKAN SKRIPSI

Nama : Rizqi Budhi Prastiyo
NIM : 1118902
Jurusan : Teknik Informatika S-1
Judul : Perancangan Sistem Antrian Pasien Poliklinik Berbasis SMS Gateway

Penguji	Perbaikan	Tanda Tangan
Penguji 1	1. Halaman 19,25,29 landscape menghadap keluar 2. Perbaiki tujuan, rumusan masalah 3. Halaman 63 analisa data dihapus 4. Referensi kasih nomor urut	
Penguji 2	1. Tambahkan struktur menu 2. Alur sistem dalam flowchart 3. ID pasien auto increment 4. DFD : data alir adalah kata benda 5. Perbaiki kesimpulan dan saran	

Anggota Penguji :

Penguji Pertama

Ali Mahmudi, BEng, PhD

NIP.P. 1031000429

Penguji Kedua

Nurlaily Vendyansyah, ST

Mengetahui,

Dosen Pembimbing I

Yoseph Dedy Irawan, ST, MT.
NIP. 19740416 200501 1 002

Dosen Pembimbing II

Suryo Adh Wibowo, ST.
NIP.P.103100043

LAMPIRAN V

Lampiran 5 : Lampiran Script

```
unit Unit_admin;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, SUIForm, Grids, DBGrids, SUIDBCtrls, SUIButton,
  StdCtrls, SUIEdit, DB, ADODB;

type
  TForm_admin = class(TForm)
    suiForm1: TsuiForm;
    GroupBox1: TGroupBox;
    Label1: TLabel;
    Edit_admin: TsuiEdit;
    Button_clear: TsuiButton;
    suiDBGrid1: TsuiDBGrid;
    Button_insert: TsuiButton;
    Button_close: TsuiButton;
    DataSource1: TDataSource;
    ADOQuery_admin: TADOQuery;
    ADOQuery_cek: TADOQuery;
    procedure FormActivate(Sender: TObject);
    procedure Edit_adminKeyPress(Sender: TObject; var Key: Char);
    procedure Button_closeClick(Sender: TObject);
    procedure Button_clearClick(Sender: TObject);
    procedure Button_insertClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form_admin: TForm_admin;

implementation

uses Unit_insert_admin;

{$R *.dfm}

procedure SetGridColumnWidths(Grid: Tdbgrid);
const
  DEFBORDER = 10;
var
  temp, n: Integer;
  lmax: array [0..30] of Integer;
begin
```

```

// prosedur ini digunakan untuk menyesuaikan ukuran lebar setiap field di dbgrid dengan lebar
kata-kata yang ada di database
try
with Grid do
begin
Canvas.Font := Font;
for n := 0 to Columns.Count - 1 do
lmax[n] := Canvas.TextWidth(Fields[n].FieldName) + DEFBORDER;
grid.DataSource.DataSet.First;
while not grid.DataSource.DataSet.EOF do
begin
for n := 0 to Columns.Count - 1 do
begin
temp := Canvas.TextWidth(trim(Columns[n].Field.DisplayText)) + DEFBORDER;
if temp > lmax[n] then lmax[n] := temp;
end; {for}
grid.DataSource.DataSet.Next;
end;
grid.DataSource.DataSet.First;
for n := 0 to Columns.Count - 1 do
if lmax[n] > 0 then
Columns[n].Width := lmax[n];
end;
except
end;
end; {SetGridColumnWidths }

procedure admin;
begin
// prosedur ini digunakan untuk menampilkan data admin
with Form_admin.ADOQuery_admin do
begin
Close;
SQL.Text:='select id_admin as "ID", nama_pegawai as Nama,alamat_pegawai as Alamat,
telp_pegawai as Telp from view_admin where status_pegawai="true"';
Open;
end;
//setelah data diselect/ditampilkan maka lebar field di dbgrid akan disesuaikan dengan
panjang kata-kata yg ada ditabasnya
SetGridColumnWidths(Form_admin.suiDBGrid1);
end;

procedure TForm_admin.FormActivate(Sender: TObject);
begin
//untuk memanggil fungsi prosedur cukup menggunakan namanya saja
admin;

Edit_admin.Clear;
Edit_admin.SetFocus;

end;

```

```

procedure TForm_admin.Edit_adminKeyPress(Sender: TObject; var Key: Char);
begin
admin;
  with ADOQuery_admin do
  begin
  //digunakan untuk menampilkan data saat dilakukan pencarian/searching jika ditemukan
data yg sesuai dengan inputan. BOF itu Begin Of File
  if bof then
  begin
  close;
  sql.clear;
  SQL.Text:='select id_admin as "ID", nama_pegawai as Nama,alamat_pegawai as Alamat,
telp_pegawai as Telp from view_admin where nama_pegawai like"' +Edit_admin.Text+'%' and
status_pegawai="true" ';
  open;
  SetGridColumnWidths(suiDBGrid1);
  end;
  //apabila tidak ditemukan data yg sesuai dng inputan maka akan ditampilkan data
keseluruhan dari prosedur admin
  if eof then
  begin
  admin;
  end;
  end;

end;

procedure TForm_admin.Button_closeClick(Sender: TObject);
begin
//digunakan untuk keluar/close form
Close;
end;

procedure TForm_admin.Button_clearClick(Sender: TObject);
begin
//digunakan untuk menjalankan fungsi yg ada di prosedur admin
admin;
end;

procedure TForm_admin.Button_insertClick(Sender: TObject);
begin
//digunakan untuk menampilkan form insert admin
Form_insert_admin.ShowModal;
end;

end.
unit Unit_cari_pegawai;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

```

Dialogs, ExtCtrls, SUIForm, Grids, DBGrids, SUIDBCtrls, SUIButton,
StdCtrls, SUIEdit, DB, ADODB;

```
type
TForm_cari_pegawai = class(TForm)
  suiForm1: TsuiForm;
  GroupBox1: TGroupBox;
  Label1: TLabel;
  Edit_pegawai: TsuiEdit;
  Button_clear: TsuiButton;
  suiDBGrid1: TsuiDBGrid;
  Edit_pilihan: TsuiEdit;
  Label2: TLabel;
  Label_pilihan: TLabel;
  Button_close: TsuiButton;
  DataSource1: TDataSource;
  ADOQuery_pegawai: TADOQuery;
  procedure FormActivate(Sender: TObject);
  procedure suiDBGrid1CellClick(Column: TColumn);
  procedure Button_clearClick(Sender: TObject);
  procedure Button_closeClick(Sender: TObject);
  procedure Edit_pegawaiKeyPress(Sender: TObject; var Key: Char);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form_cari_pegawai: TForm_cari_pegawai;

implementation

uses Unit_insert_admin, Unit_pegawai;

{$R *.dfm}

procedure SetGridColumnWidths(Grid: Tdbgrid);
const
  DEFBORDER = 10;
var
  temp, n: Integer;
  lmax: array [0..30] of Integer;
begin
  // prosedur ini digunakan untuk menyesuaikan ukuran lebar setiap field di dbgrid dengan lebar
  kata-kata yang ada di database
  try
    with Grid do
      begin
        Canvas.Font := Font;
        for n := 0 to Columns.Count - 1 do
          lmax[n] := Canvas.TextWidth(Fields[n].FieldName) + DEFBORDER;
```

```

grid.DataSource.DataSet.First;
while not grid.DataSource.DataSet.EOF do
begin
for n := 0 to Columns.Count - 1 do
begin
temp := Canvas.TextWidth(trim(Columns[n].Field.DisplayText)) + DEFBORDER;
if temp > lmax[n] then lmax[n] := temp;
end; {for}
grid.DataSource.DataSet.Next;
end;
grid.DataSource.DataSet.First;
for n := 0 to Columns.Count - 1 do
if lmax[n] > 0 then
Columns[n].Width := lmax[n];
end;
except
end;
end; {SetGridColumnWidths }

procedure pegawai;
//digunakan untuk menampilkan data pegawai
begin
with Form_cari_pegawai.ADOQuery_pegawai do
begin
Close;
SQL.Text:='select id_pegawai as "ID",nama_pegawai as Nama,alamat_pegawai as Alamat,
jk_pegawai as JK, telp_pegawai as Telp from pegawai where status_pegawai="true";
Open;
end;
SetGridColumnWidths(Form_pegawai.suiDBGrid1);
end;

procedure TForm_cari_pegawai.FormActivate(Sender: TObject);
begin
//memanggil fingsi yg ada di prosedur pegawai
pegawai;

Edit_pegawai.Clear;
Edit_pegawai.SetFocus;
end;

procedure TForm_cari_pegawai.suiDBGrid1CellClick(Column: TColumn);
begin
//digunakan untuk mngirimkan data ke form insert admin(disetiao editnya) pada saat dbgrid
diklik
Form_insert_admin.Edit_nip.Text:=ADOQuery_pegawai.Fields[0].AsString;
Form_insert_admin.Edit_nama.Text:=ADOQuery_pegawai.Fields[1].AsString;
Edit_pilihan.Text:=ADOQuery_pegawai.Fields[1].AsString;

end;

```

```

procedure TForm_cari_pegawai.Button_clearClick(Sender: TObject);
begin
    pegawai;
end;

procedure TForm_cari_pegawai.Button_closeClick(Sender: TObject);
begin
    Close;
end;

procedure TForm_cari_pegawai.Edit_pegawaiKeyPress(Sender: TObject;
    var Key: Char);
begin
    pegawai;
    //digunakan untuk melakukan pencarian data sesuai dengan kata yg diinputkan
    with ADOQuery_pegawai do
    begin
        if bof then
        begin
            close;
            sql.clear;
            SQL.Text:='select id_pegawai as "ID",nama_pegawai as Nama,alamat_pegawai as Alamat,
jk_pegawai as JK, telp_pegawai as Telp from pegawai where nama_pegawai
like '"+Edit_pegawai.Text+'%' and status_pegawai="true" ';
            open;
            SetGridColumnWidths(suiDBGrid1);
            end;
        if eof then
        begin
            pegawai;
            end;
        end;
    end;

end;

end.

unit Unit_change_password;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, SUIButton, StdCtrls, SUIEdit, ExtCtrls, SUIForm, ADODB, DB;

type
    TForm_change_password = class(TForm)
        suiForm1: TsuiForm;
        GroupBox1: TGroupBox;
        GroupBox2: TGroupBox;
        Label3: TLabel;
        Label4: TLabel;
    end;

```

```

Label5: TLabel;
Edit_username: TsuiEdit;
Edit_old_password: TsuiEdit;
Edit_new_password: TsuiEdit;
GroupBox3: TGroupBox;
Label2: TLabel;
Label1: TLabel;
Edit_nip: TsuiEdit;
Edit_nama: TsuiEdit;
Button_change: TsuiButton;
Button_close: TsuiButton;
ADOTable1: TADOTable;
ADOQuery_cek: TADOQuery;
ADOQuery_password: TADOQuery;
procedure FormActivate(Sender: TObject);
procedure Button_changeClick(Sender: TObject);
procedure Button_closeClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form_change_password: TForm_change_password;

implementation

{$R *.dfm}

procedure clear;
begin
  with Form_change_password do
  begin
    Edit_username.Clear;
    Edit_old_password.Clear;
    Edit_new_password.Clear;
    Edit_username.SetFocus;
  end;
end;

procedure TForm_change_password.FormActivate(Sender: TObject);
begin
  clear;
  with ADOQuery_password do
  begin
    Close;
    SQL.Clear;
    SQL.Text:='select nama_pegawai from pegawai where id_pegawai="'+Edit_nip.Text+"' AND
status_pegawai="true";
    Open;
  end;

```



```

Edit_nama.Text:=ADOQuery_password.Fields[0].AsString;
end;

procedure TForm_change_password.Button_changeClick(Sender: TObject);
begin
try
//apabila semua data belum diisi maka akan muncul peringatan
if
(Edit_nip.Text=""or(Edit_nama.Text=""or(Edit_username.Text=""or(Edit_new_password.Text=""
)or(Edit_old_password.Text=""then
begin
ShowMessage('Data masih ada yang kosong');
end else

//apabila semua data diisi
if
(Edit_nip.Text<>")and(Edit_nama.Text<>")and(Edit_username.Text<>")and(Edit_new_passwor
d.Text<>")and(Edit_old_password.Text<>")then
begin
// muncul konfirmasi untuk perubahan data
if messagedlg('Apakah Password Ingin Diubah?',mtconfirmation,[mbYes]+[mbNo],0)=mrYes
then
begin
//dilakukan pencarian data admin dengan username dan password yg diinputkan
with ADOQuery_cek do
begin
Close;
SQL.Clear;
SQL.Text:='select * from admin where id_pegawai="'+Edit_nip.Text+" and
username="'+Edit_username.Text+" AND password="'+Edit_old_password.Text+'";
Open;
end;
//apabila ada data yang ditemukan
if ADOQuery_cek.RecordCount<>0 then
begin
//digunakan untuk melakukan pencocokan data(NIP)
ADOTable1.Active:=false;
ADOTable1.Active:=true;
if (ADOTable1.Locate('id_pegawai',Edit_nip.Text,[])) then
begin
//dilakukan perubahan data password sesuai dengan inoutan password baru
ADOTable1.Edit;
ADOTable1.FieldValues['password']:=Edit_new_password.Text;
ADOTable1.Post;
//ditampilkan pesan data berubah
ShowMessage('Data Berubah');
end;
end else
//apabila data tidak sesuai
if ADOQuery_cek.RecordCount=0 then
begin
ShowMessage('Cek Kembali Username dan Old Password Anda');

```

```

        Edit_username.SetFocus;
    end;
end;
end;
//apabila ada data yg tidak diisi
except
    ShowMessage('Cek Kembali');
end;
end;

procedure TForm_change_password.Button_closeClick(Sender: TObject);
begin
    Close;

end;

end.

unit Unit_daftar_pasien;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, ExtCtrls, SUIForm, Grids, DBGrids, SUIDBCtrls, SUIButton,
    StdCtrls, SUIEdit, SUIComboBox, DB, ADODB, ComCtrls, QuickRpt, QRCtrls;

type
    TForm_daftar_pasien = class(TForm)
        suiForm1: TsuiForm;
        GroupBox1: TGroupBox;
        Label1: TLabel;
        Edit_pasien: TsuiEdit;
        Button_clear: TsuiButton;
        suiDBGrid1: TsuiDBGrid;
        Button_update: TsuiButton;
        Button_close: TsuiButton;
        Label2: TLabel;
        suiRadioButton_pasien: TsuiRadioButton;
        suiRadioButton_poli: TsuiRadioButton;
        suiComboBox_poli: TsuiComboBox;
        Label3: TLabel;
        DataSource1: TDataSource;
        ADOQuery_antrian: TADOQuery;
        ADOQuery_cek: TADOQuery;
        DateTimePicker1: TDateTimePicker;
        ADOQuery_poli: TADOQuery;
        Label_poli: TLabel;
        suiButton_cari: TsuiButton;
        DateTimePicker2: TDateTimePicker;
        QuickRep1: TQuickRep;
        DetailBand1: TQRBand;
    end;
end.

```

```

TitleBand1: TQRBand;
QRLabel1: TQRLabel;
QRLabel2: TQRLabel;
QRLabel3: TQRLabel;
QRLabel4: TQRLabel;
QRLabel5: TQRLabel;
QRShape8: TQRShape;
QRShape9: TQRShape;
QRShape10: TQRShape;
QRShape11: TQRShape;
QRShape12: TQRShape;
QRShape13: TQRShape;
QRShape14: TQRShape;
QRDBText7: TQRDBText;
QRDBText6: TQRDBText;
QRDBText5: TQRDBText;
QRDBText4: TQRDBText;
QRDBText3: TQRDBText;
QRDBText2: TQRDBText;
QRLabel_poli: TQRLabel;
QRLabel_tgl: TQRLabel;
suiButton_print: TsuiButton;
QRShape7: TQRShape;
QRShape6: TQRShape;
QRShape5: TQRShape;
QRShape4: TQRShape;
QRShape3: TQRShape;
QRShape2: TQRShape;
QRShape1: TQRShape;
QRLabel9: TQRLabel;
QRLabel8: TQRLabel;
QRLabel7: TQRLabel;
QRLabel6: TQRLabel;
QRLabel12: TQRLabel;
QRLabel11: TQRLabel;
QRLabel10: TQRLabel;
QRDBText1: TQRDBText;
Button_insert: TsuiButton;
suiButton_print_hari: TsuiButton;
procedure suiComboBox_poliChange(Sender: TObject);
procedure suiComboBox_poliEnter(Sender: TObject);
procedure Button_closeClick(Sender: TObject);
procedure Edit_pasienKeyPress(Sender: TObject; var Key: Char);
procedure FormActivate(Sender: TObject);
procedure Button_clearClick(Sender: TObject);
procedure suiButton_cariClick(Sender: TObject);
procedure Button_updateClick(Sender: TObject);
procedure suiButton_printClick(Sender: TObject);
procedure Button_insertClick(Sender: TObject);
procedure suiButton_print_hariClick(Sender: TObject);
private
{ Private declarations }

```

```

public
  { Public declarations }
end;

var
  Form_daftar_pasien: TForm_daftar_pasien;

implementation

uses Unit_insert_daftar_pasien;

{$R *.dfm}

procedure SetGridColumnWidths(Grid: Tdbgrid);
const
  DEFBORDER = 10;
var
  temp, n: Integer;
  lmax: array [0..30] of Integer;
begin
  // prosedur ini digunakan untuk menyesuaikan ukuran lebar setiap field di dbgrid dengan lebar
  // kata-kata yang ada di database
  try
    with Grid do
      begin
        Canvas.Font := Font;
        for n := 0 to Columns.Count - 1 do
          lmax[n] := Canvas.TextWidth(Fields[n].FieldName) + DEFBORDER;
        grid.DataSource.DataSet.First;
        while not grid.DataSource.DataSet.EOF do
          begin
            for n := 0 to Columns.Count - 1 do
              begin
                temp := Canvas.TextWidth(trim(Columns[n].Field.DisplayText)) + DEFBORDER;
                if temp > lmax[n] then lmax[n] := temp;
              end; {for}
            grid.DataSource.DataSet.Next;
          end;
          grid.DataSource.DataSet.First;
          for n := 0 to Columns.Count - 1 do
            if lmax[n] > 0 then
              Columns[n].Width := lmax[n];
            end;
          except
            end;
        end; {SetGridColumnWidths }

procedure antrian;
begin
  //menseset tanggal sekarang
  Form_daftar_pasien.DateTimePicker2.Date:=now;
  //menampilkan data pendaftaran sesuai tgl sekarang

```

```

with Form_daftar_pasien.ADOQuery_antrian do
begin
Close;
SQL.Text:='select id_janji as ID, tgl_janji as Tanggal, id_pasien as "ID/No.Kartu", nama_pasien
as Nama, alamat_pasien as Alamat, nama_poli as Poli, nama_dokter as Dokter, urutan as
Urutan, status_pendaftaran as Status from view_pendaftaran where
tgl_janji="'+DateToStr(Form_daftar_pasien.DateTimePicker2.Date)+'";
Open;
end;
//menyesuaikan ukuran lebar dgrid dengan panjang datanya
SetGridColumnWidths(Form_daftar_pasien.suiDBGrid1);
end;

procedure TForm_daftar_pasien.suiComboBox_poliChange(Sender: TObject);
begin
//menampilkan id poliklinik sesuai dengan nama poliklinik yg dipilih kedalam label_poli
ADOQuery_poli.Close;
ADOQuery_poli.SQL.Text:='select id_poli from poliklinik where
nama_poli="'+suiComboBox_poli.text+'";
ADOQuery_poli.Open;
Label_poli.Caption:=ADOQuery_poli.Fields.Fields[0].AsString;

end;

procedure TForm_daftar_pasien.suiComboBox_poliEnter(Sender: TObject);
begin
suiComboBox_poli.Clear;
//menampilkan data nama piliklinik kedalam combobox poli
ADOQuery_poli.Close;
ADOQuery_poli.sql.Text:='select distinct * from poliklinik';
ADOQuery_poli.Open;
while not ADOQuery_poli.Eof do
begin
suiComboBox_poli.Items.Add(ADOQuery_poli.Fields.Fields[1].asString);
ADOQuery_poli.Next;
end;

end;

procedure TForm_daftar_pasien.Button_closeClick(Sender: TObject);
begin
Close;
end;

procedure TForm_daftar_pasien.Edit_pasienKeyPress(Sender: TObject;
var Key: Char);
begin
//digunakan untuk melakukan pencarian data sesuai dengan inputan yg diketik oleh user
antrian;
with ADOQuery_antrian do
begin
if bof then

```

```

begin
  close;
  sql.clear;
  SQL.Text:='select id_janji as ID, tgl_janji as Tanggal, id_pasien as "ID/No.Kartu",
nama_pasien as Nama, alamat_pasien as Alamat, nama_poli as Poli, nama_dokter as Dokter,
urutan as Urutan, status_pendaftaran as Status from view_pendaftaran where
tgl_janji="'+DateToStr(DateTimePicker1.Date)+'" and nama_pasien like "'+Edit_pasien.Text+'%'
';
  open;
  SetGridColumnWidths(suiDBGrid1);
  end;
if eof then
begin
  antrian;
  end;
end;

end;

procedure TForm_daftar_pasien.FormActivate(Sender: TObject);
begin
  //ketika form ini tampil maka akan dilakukan penampilan data antrian/pendaftaran
  antrian;
  Edit_pasien.Clear;
  DateTimePicker1.Date:=now;
  end;

procedure TForm_daftar_pasien.Button_clearClick(Sender: TObject);
begin
  antrian;
  end;

procedure TForm_daftar_pasien.suiButton_cariClick(Sender: TObject);
begin
  //apabila radio button pasien dipilih/dicheck dan edit nama pasien diisi maka akan dilakukan
  pencarian data pendaftaran sesuai dengan nama pasien tersebut dan tgl di datetimepicker.
  if suiRadioButton_pasien.Checked=true then
  begin
    with ADOQuery_antrian do
    begin
      close;
      sql.clear;
      SQL.Text:='select id_janji as ID, tgl_janji as Tanggal, id_pasien as "ID/No.Kartu",
nama_pasien as Nama, alamat_pasien as Alamat, nama_poli as Poli, nama_dokter as Dokter,
urutan as Urutan, status_pendaftaran as Status from view_pendaftaran where
tgl_janji="'+DateToStr(DateTimePicker1.Date)+'" and nama_pasien like "'+Edit_pasien.Text+'%'
';
      open;
      SetGridColumnWidths(suiDBGrid1);
      end;
    end else
  //dilakukan pencarian data pendaftaran/antrian sesuai dengan poliklinik yg dipilih

```

```

with ADOQuery_antrian do
begin
close;
sql.clear;
SQL.Text:='select id_janji as ID, tgl_janji as Tanggal, id_pasien as "ID/No.Kartu",
nama_pasien as Nama, alamat_pasien as Alamat, nama_poli as Poli, nama_dokter as Dokter,
urutan as Urutan, status_pendaftaran as Status from view_pendaftaran where
tgl_janji="' + DateToStr(DateTimePicker1.Date) + "' and id_poli="' + Label_poli.Caption + '%' ' ';
open;
SetGridColumnWidths(suiDBGrid1);
end;
end;

procedure TForm_daftar_pasien.Button_updateClick(Sender: TObject);
var
kode:string;
begin
//digunakan untuk mengubah status pendaftaran pasien yang awalnya mengantri menjadi
selesai
kode:=ADOQuery_antrian.Fields.Fields[0].AsString;
if kode<>' ' then
begin
if messagedlg('Status Pendaftaran Pasien: ' + ADOQuery_antrian.Fields.Fields[3].AsString +
'+ akan diubah?', mtconfirmation, [mbYes] + [mbNo], 0) = mrYes then
begin
try
with ADOQuery_antrian do
begin
Close;
SQL.Clear;
SQL.Text:='update pendaftaran set status_pendaftaran="Selesai" where
id_janji="' + kode + "'";
ExecSQL;
end;
antrian;
except
ShowMessage('Data Masih Digunakan');
antrian;
end;
end else

end else
ShowMessage('Harap Memilih Data Terlebih Dahulu');
end;

procedure TForm_daftar_pasien.suiButton_printClick(Sender: TObject);
begin
//digunakan untuk mencetak laporan data pendaftaran/antrian sesuai dengan tgl dan poliklinik
yg dipilih
with ADOQuery_antrian do
begin
close;

```

```

    sql.clear;
    SQL.Text:='select id_janji, tgl_janji , id_pasien , nama_pasien, alamat_pasien, nama_poli ,
nama_dokter, urutan, status_pendaftaran from view_pendaftaran where
tgl_janji="'+DateToStr(DateTimePicker1.Date)+'" and id_poli="'+Label_poli.Caption+'%" order
by id_janji ASC';
    open;
    SetGridColumnWidths(suiDBGrid1);
    end;
    QRLabel_poli.Caption:=suiComboBox_poli.Text;
    QRLabel_tgl.Caption:=DateToStr(DateTimePicker1.Date);
    ADOQuery_antrian.Active:=true;
    QuickRep1.Preview;
end;

```

```

procedure TForm_daftar_pasien.Button_insertClick(Sender: TObject);
begin
Form_add_pendaftaran.ShowModal;
end;

```

```

procedure TForm_daftar_pasien.suiButton_print_hariClick(Sender: TObject);
begin
//digunakan untuk mencetak laporan data pendaftaran/antrian sesuai dengan tgl yg dipilih
with ADOQuery_antrian do
begin
close;
sql.clear;
SQL.Text:='select id_janji, tgl_janji , id_pasien , nama_pasien, alamat_pasien, nama_poli ,
nama_dokter, urutan, status_pendaftaran from view_pendaftaran where
tgl_janji="'+DateToStr(DateTimePicker1.Date)+'" order by id_janji ASC';
open;
SetGridColumnWidths(suiDBGrid1);
end;
QRLabel_poli.Caption:='Semua Poliklinik';
QRLabel_tgl.Caption:=DateToStr(DateTimePicker1.Date);
ADOQuery_antrian.Active:=true;
QuickRep1.Preview;

```

```
end;
```

```
end.
```