BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan Sistem

Perancangan sistem adalah tahapan setelah analisis dari siklus pengembangan sistem yang didefinisikan dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi yang menggambarkan bagaimana suatu sistem dibentuk, yang dapat berupa penggambaran, perancangan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi, juga menyangkut konfigurasi dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem.

4.1.1 Tujuan Perancangan Sistem

Perancangan sistem bertujuan untuk memberikan gambaran yang jelas dan rancang bangun yang sesuai dengan kebutuhan *user* atau pemakai sistem itu sendiri. Perancangan sistem atau desain sistem dilakukan apabila tahap dari analisis sistem telah selesai dilakukan. Maka untuk selanjutnya seorang analisis sistem merancang bagaimana membentuk sistem yang baru ataupun memperbaharui sistem yang lama. Tahap inilah yang dinamakan dengan istilah dari perancangan sistem.

Adapun tujuan dari perancangan sistem yang diusulkan adalah sebagai berikut:

1. Meningkatkan efisiensi dari pemakaian sumber daya dan meningkatkan kinerja bagian pendaftaran pada Puskesmas Cibiru

2. Memperbaiki pengolahan data yang masih dilakukan secara manual menjadi terkomputerisasi

4.1.2 Gambaran Umum sistem Yang Diusulkan

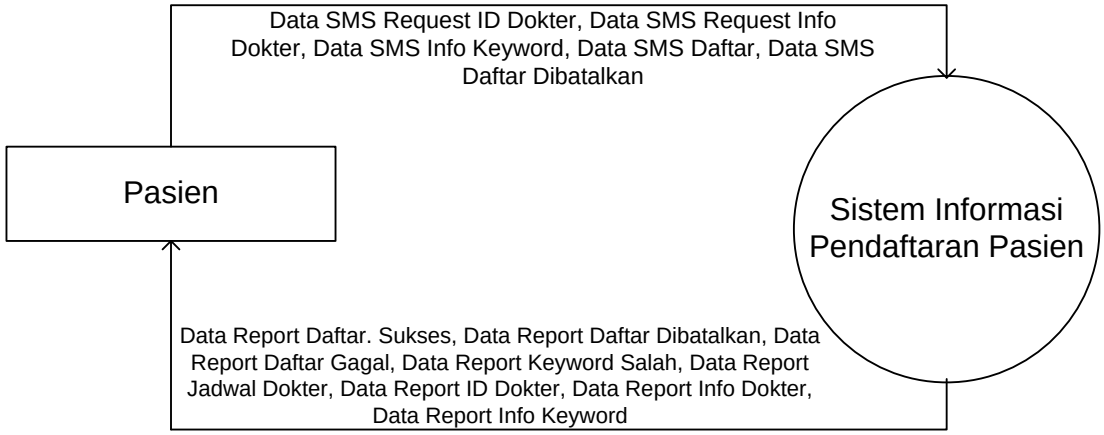
Sistem informasi yang akan dibangun oleh penulis adalah sistem informasi yang akan digunakan oleh bagian pendaftaran pasien, dimana sistem informasi ini memungkinkan pasien untuk mendaftar melalui SMS tanpa menghilangkan prosedur lama yaitu secara manual, dan bagian pendaftaran dapat menggunakan sistem informasi ini untuk membuat daftar antrian pasien pada hari yang bersangkutan. Sistem informasi yang dibangun juga menyediakan media penyimpanan database yang relatif banyak, yang dapat digunakan untuk menyimpan data-data biodata pasien dan daftar pasien yang berobat di Puskesmas Cibiru.

4.1.3 Perancangan Prosedur Yang Diusulkan

Dalam perancangan prosedur yang diusulkan ini meliputi *Diagram Konteks, Data Flow Diagram* dan Kamus Data, yang bertujuan untuk memudahkan dalam pembuatan program dan memudahkan dalam menganalisa aliran dokumen.

4.1.3.1 Diagram Konteks

Diagram Konteks merupakan alat bantu analisis yang menggambarkan secara garis besar dari alur proses data yang terdapat pada sistem. Berikut adalah Diagram Konteks dari sistem yang diusulkan :



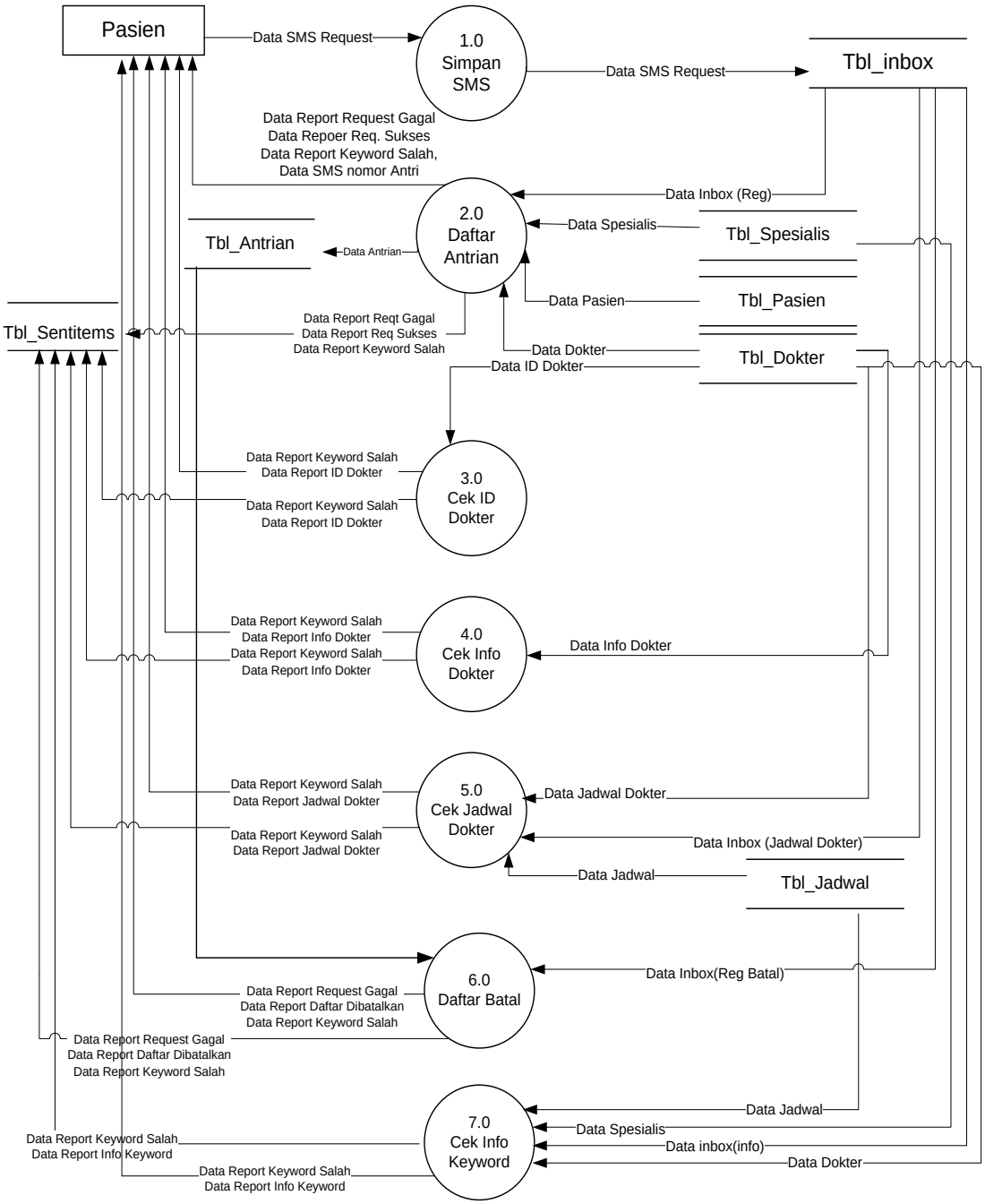
Gambar 4.1 Diagram Konteks yang Diusulkan

4.1.3.2 Data Flow Diagram

Data Flow Diagram merupakan gambaran alur proses data secara detail, dengan kata lain, *Data Flow Diagram* merupakan perkembangan atau penguraian dari proses pada Diagram Konteks sebelumnya.

4.1.3.2.1 DFD Level 1

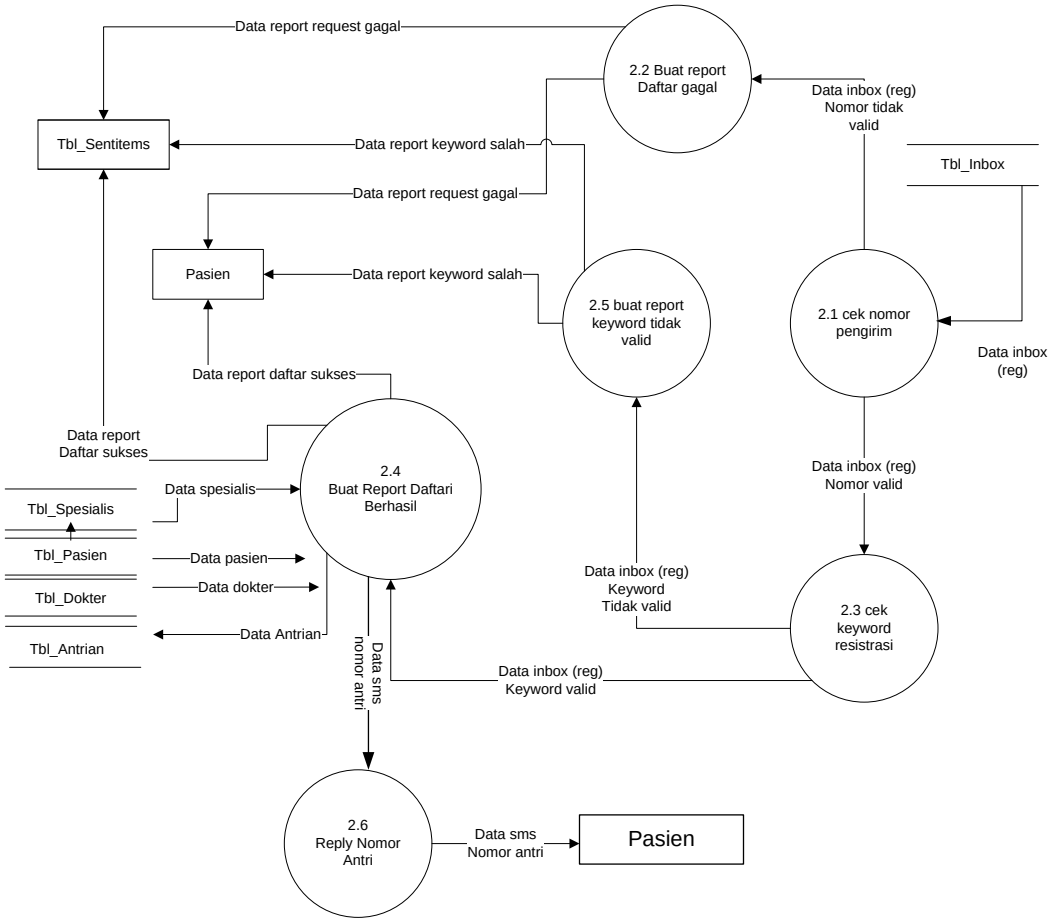
DFD Level 1 dari sistem yang diusulkan adalah sebagai berikut



Gambar 4.2 DFD Level 1 yang Diusulkan

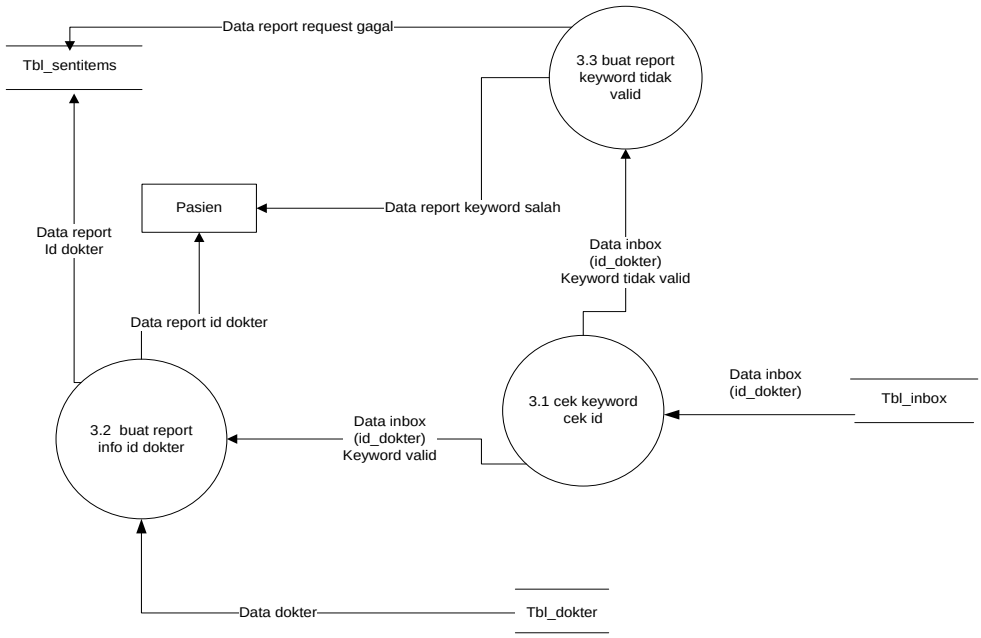
DFD Level 2

DFD Level 2 merupakan penguraian dari masing-masing proses yang terdapat pada DFD Level 1 agar lebih jelas. Berikut adalah DFD Level 2 proses 2 sistem yang diusulkan pada Puskesmas CIbiru:



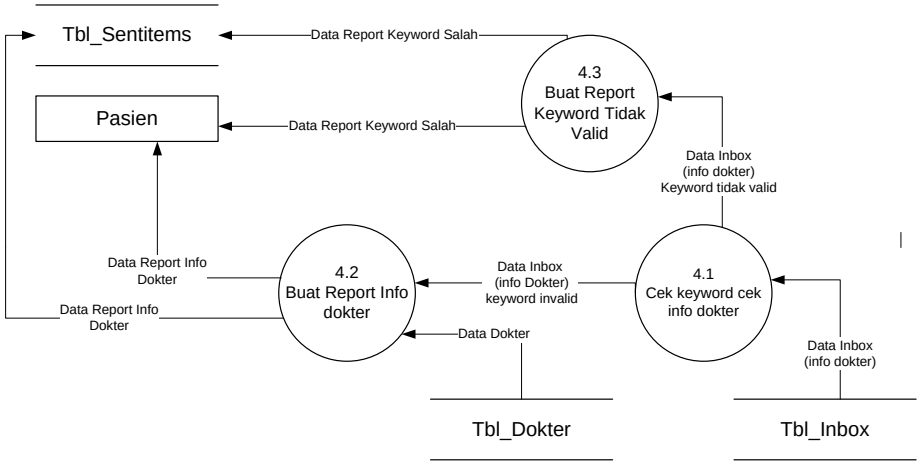
Gambar 4.3 DFD Level 2 Proses 2

DFD Level 2 Proses 3



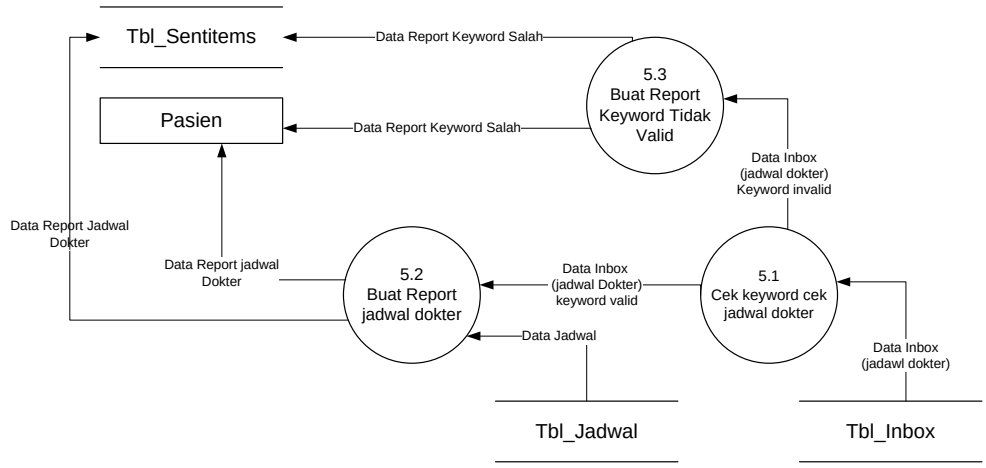
Gambar 4.4 DFD Level 2 Proses 3

DFD Level 2 Proses 4



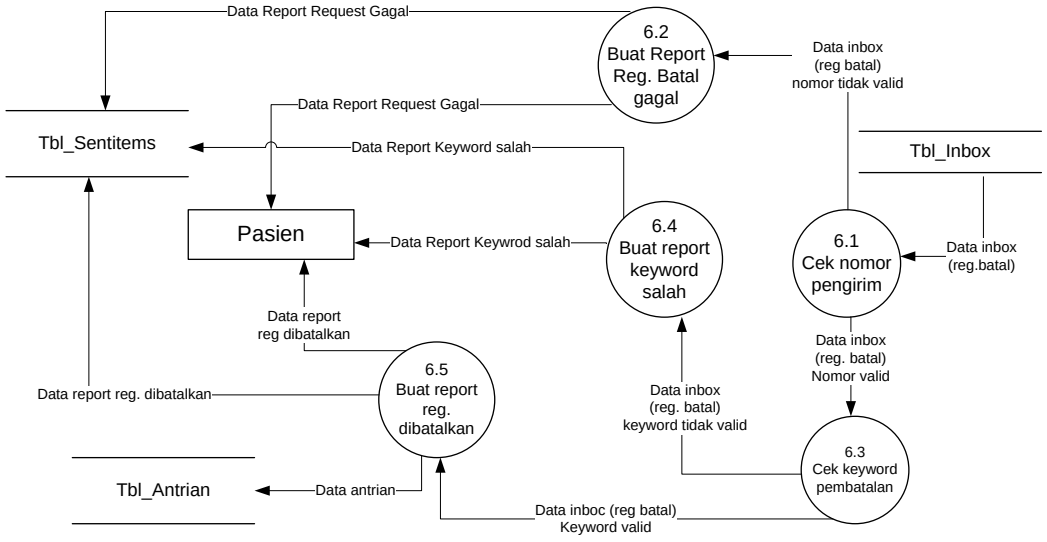
Gambar 4.5 DFD Level 2 Proses 4

DFD Level 2 Proses 5



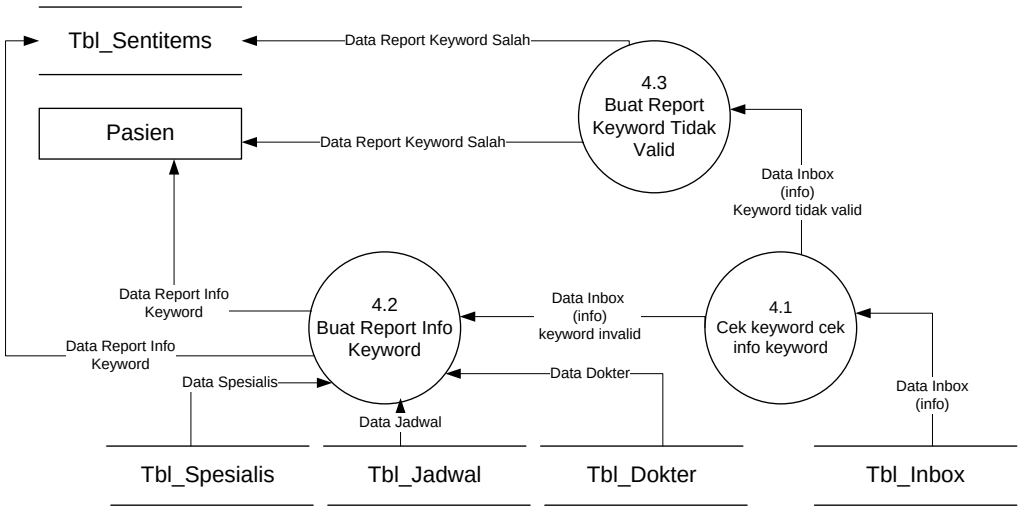
Gambar 4.6 DFD Level 2 Proses 5

DFD Level 2 Proses 6



Gambar 4.7 DFD Level 2 Proses 6

DFD Level 2 Proses 7



Gambar 4.8 DFD Level 2 Proses 7 yang Diusulkan

4.1.3.3 Kamus Data

Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Kamus data dibuat berdasarkan arus data yang ada di *data flow diagram*. Berikut kamus data yang digunakan :

1. Nama Arus Data : Data SMS Request Info Dokter

   Alias : Data Inbox (Info Dokter), Data Inbox (Jadwal Dokter), Data Report Info Dokter, Data Report Jadwal Dokter, Data Inbox (Info Dokter) keyword tidak valid, Data Inbox (Info Dokter) keyword valid, Data Inbox (Jadwal Dokter) Keyword tidak valid, Data Inbox (Jadwal Dokter) Keyword valid

Desktripsi : Data berupa SMS dalam bentuk tejs dengan maksimal 160 karakter

Aliran Data : Entitas Pasien – Proses 1.0, Proses 1.0 – Tbl_inbox, Tbl_inbox – Proses 4.0

Struktur File : no_pengirim, text

2. Nama Arus Data : Data SMS Request ID Dokter

Alias : Data Inbox (ID Dokter), Data Report ID Dokter, Data Inbox (ID Dokter) keyword tidak valid, Data Inbox (ID Dokter) keyword valid

Desktripsi : Data berupa SMS dalam bentuk tejs dengan maksimal 160 karakter

Aliran Data : Entitas Pasien – Proses 1.0, Proses 1.0 – Tbl_inbox, Tbl_inbox – Proses 3.0

Struktur File : no_pengirim, text

3. Nama Arus Data : Data SMS Daftar

Alias : Data Inbox (Reg), Data Report Reg. Sukses, Data Inbox (Reg) Nomor Valid, Data Inbox (Reg) Keyword Valid, Data Inbox (Reg) Nomor Tidak Valid, Data Inbox (Reg) Keyword tidak Valid, Data Inbox (Reg. Batal) Nomor Tidak Valid, Data Inbox (Reg. Batal) Nomor Valid, Data Inbox (Reg. Batal) Keyword Valid, Data Inbox (Reg. Batal) Keyword tidak Valid, Data SMS Nomor Antri

Desktripsi : Data berupa SMS dalam bentuk tejs dengan maksimal 160 karakter

Aliran Data : Entitas Pasien – Proses 1.0, Proses 1.0 – Tbl_inbox, Tbl_inbox – Proses 2.0

Struktur File : no_pengirim, text

4. Nama Arus Data : Data SMS Registrasi Dibatalkan

   Alias : Data Inbox (Reg. Batal), Data Report Reg. Dibatalkan, Data Report Keyword Salah

   Desktripsi : Data berupa SMS dalam bentuk tejs dengan maksimal 160 karakter

   Aliran Data : Entitas Pasien – Proses 1.0, Proses 1.0 – Tbl_inbox, Tbl_inbox – Proses 7.0

   Struktur File : no_pengirim, text

5. Nama Arus Data : Data Request Gagal

   Alias : Data Report Request Gagal

   Desktripsi : Data berupa SMS dalam bentuk tejs dengan maksimal 160 karakter

   Aliran Data : Entitas Pasien – Proses 1.0, Proses 1.0 – Tbl_inbox

   Struktur File : no_pengirim, text

6. Nama Arus Data : Data Pasien

   Alias : -

   Deskripsi : Berisikan data informasi pasien yang dibutuhkan dalam proses pengolahan data

Aliran Data : Tbl_pasien – Proses 2.4

Struktur File : no_pasien, nama, alamat, pekerjaan, kelamin, umur, telepon

7. Nama Arus Data : Data SMS Request Info Keyword

   Alias : Data Inbox (Info), Data Report Info, Data Report, Data Inbox (Info) keyword tidak valid, Data Inbox (Info) keyword valid

   Desktripsi : Data berupa SMS dalam bentuk tejs dengan maksimal 160 karakter

   Aliran Data : Entitas Pasien – Proses 1.0, Proses 1.0 – Tbl_inbox, Tbl_inbox – Proses 7.0

   Struktur File : no_pengirim, text

8. Nama Arus Data : Data Dokter

   Alias : -

   Deskripsi : Berisikan data informasi dokter yang dibutuhkan dalam proses pengolahan data

   Aliran Data : Tbl_Dokter – Proses 2.4, Tbl_Dokter – Proses 3.2, Tbl_Dokter – Proses 4.2, Tbl_Dokter – Proses 5.2

   Struktur File : id_dokter, nama_dokterm nick, id_spesialis, max_antrian

9. Nama Arus Data : Data Jadwal

   Alias : -

   Deskripsi : Berisikan data informasi jadwal praktek dokter

Aliran Data : Tbl_Jadwal – Proses 5.2

Struktur File : id_jadwal, id_dokter, hari_praktek, jam_praktek

10. Data Spesialis : Data Spesialis

Alias : -

Deskripsi : Berisikan data informasi bidang spesialisasi dokter

Aliran Data : Tbl_Spesialis – Proses 2.4, Tbl_SPesialis – Proses 4.2

Struktur File : id_spesialis, spesialis

4.1.4 Perancangan Basis Data

Perancangan basis data adalah langkah untuk menentukan basis data yang diharapkan dapat mewakili seluruh kebutuhan pengguna. Perancangan *Database* dalam Sistem Informasi Pendaftaran Pasien berbasis SMS Gateway ini ditujukan agar dalam pengoperasian dan pengimplementasian, dapat diperoleh informasi yang lebih lengkap serta dapat membantu mempermudah proses manipulasi data. Pada perancangan basis data ini akan dibahas mengenai Normalisasi, Relasi Tabel, *Entity-Relationship Diagram* (ERD), Struktur File, dan Kodifikasi.

4.1.4.1 Normalisasi

Normalisasi adalah suatu proses pengelompokkan data elemen menjadi tabeltabel yang menunjukkan entity dan relasinya yang berfungsi untuk menghilangkan redudansi data atau, menentukan *key* yang unik untuk mengakses data item atau merupakan pembentukan *database* relation sedemikian rupa sehingga *database* tersebut menjadi modul modifikasi. Salah satu kegunaan

normalisasi adalah memudahkan identifikasi entitas atau objek. Adapun normalisasi dari perancangan basis data yang digunakan adalah sebagai berikut:

1. Bentuk Unnormal (UNF)

Bentuk Tidak Normal atau *Un Normalized Form* (UNF), merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti suatu format tertentu, dapat saja data tersebut tidak lengkap maupun terduplikasi. Data dikumpulkan dengan apa adanya sesuai dengan kedatangannya. Berikut ini merupakan bentuk tidak normal atau *Un Normalized Form* (UNF) yaitu: { id_antrian, id_dokter, urutan, tgl_antrian, id_pasien, status_antrian, id_dokter, nama_dokter, nick, id_spesialis, max_antrian, id_jadwal, id_dokter, hari_praktek, jam_praktek, id_pasien, nama, alamat, pekerjaan, kelamin, umur, tlp, id_spesialis, spesialis, id_antrian, id_spesialis, urutan, tgl_antrian, id_pasien, status_antrian}.

2. Bentuk Normal Pertama (1NF)

Suatu relasi dikatakan mempunyai bentuk *normal form* pertama atau *First Norm Form* (1NF) bila semua *domain* adalah sederhana *(anomatik).* Artinya setiap *atribut* mempunyai *domain* tunggal. Adapun bentuk normal pertama atau *Firs Norm Form* (1NF) yaitu: { id_pasien, nama, kelamin, alamat, umur, pekerjaan, tlp, id_spesialis, spesialis, id_dokter, nama_dokter, id_antrian, urutan, tgl_antrian, status_antrian, id_jadwal, hari_praktek, jam_praktek }

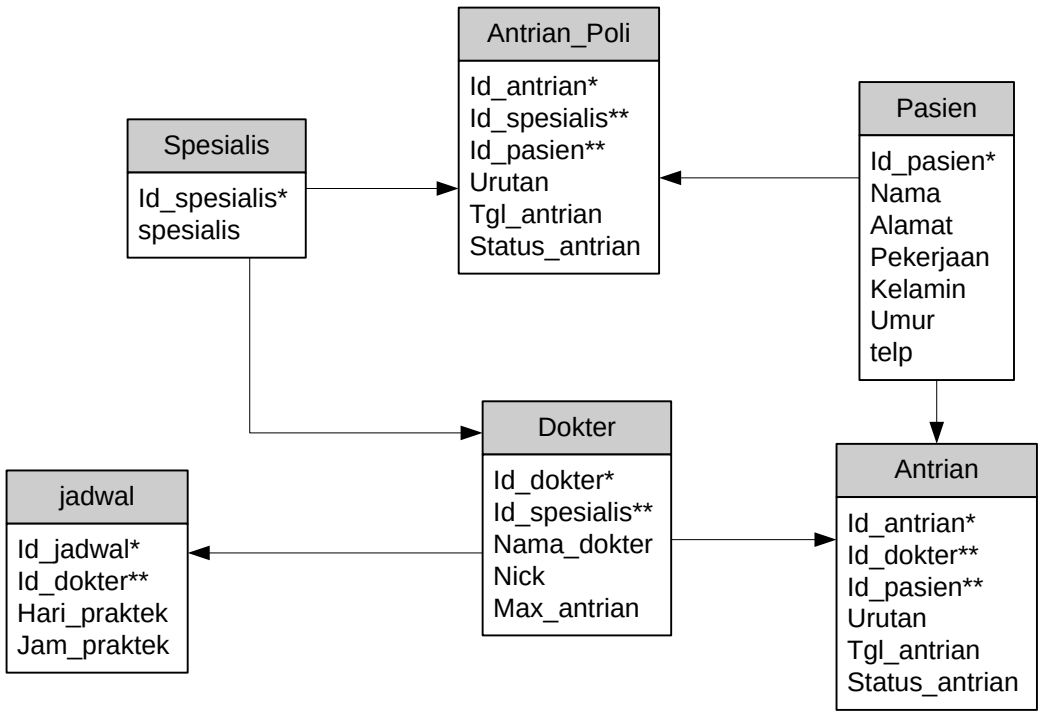
3. Bentuk Normal Kedua (2NF)

Aturan normal kedua atau Second Norm Form (2NF), menyatakan bahwa setiap field yang tidak termasuk dalam key primer memiliki ketergantungan fungsional pada key primer secara utuh. Adapun normal keduanya adalah:

Tbl_Pasien	:{id_pasien*, nama, kelamin, alamat, umur, pekerjaan, tlp}
Tbl_Spesialis	:{id_spesialis*, spesialis}
Tbl_Dokter	:{id_dokter*, nama_dokter, id_spesialiss**}
Tbl_Antrian	:{id_antrian*, id_dokter**, urutan, tgl_antrian, id_pasien**, status_antrian}
Tbl_Jadwal	:{id_jadwal*, id_dokter**, hari_praktek, jam_praktek}
Tbl_Antrian_Poli	:{id_antrian*, id_spesialis**, urutan, tgl_antrian, id_pasien**, status_antrian}

4.1.4.2 Relasi Tabel

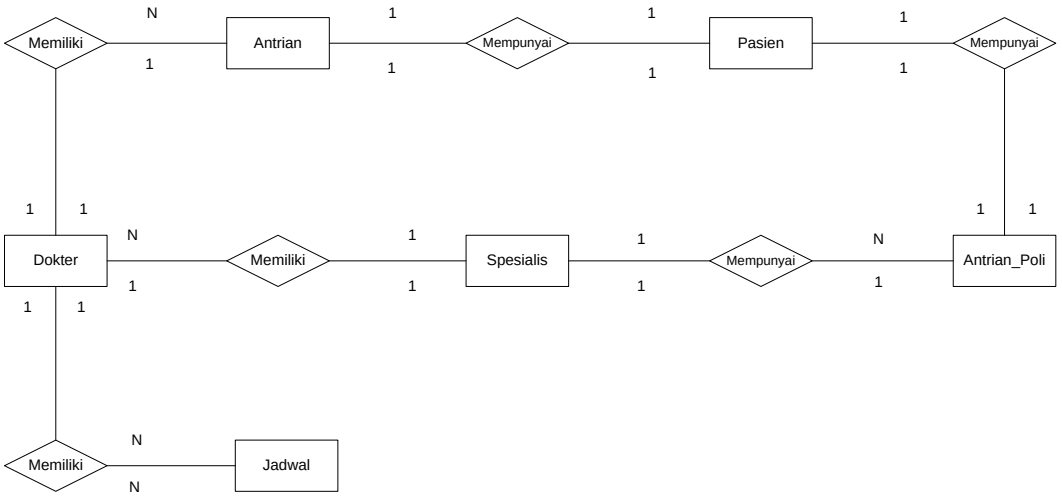
Proses relasi antar tabel merupakan pengelompokan data menjadi tabel-tabel yang menunjang entitas dan relasinya, yang berfungsi untuk mengakses data item sedemikian rupa sehingga *database* menjadi mudah dimodifikasi. Berikut ini adalah tabel relasi yang menggambarkan hubungan antar tabel yang terdapat pada *database* Sistem Informasi Pendaftaran Pasien berbasis SMS Gateway :



Gambar 4.9 Relasi Tabel

4.1.4.3 Entity Relationship Diagram

Entity Relation Diagram (ERD) adalah pengekspresian dari keadaan sebenarnya ke dalam kumpulan objek-objek dasar yang disebut entitas melalui relasi diantara entitas-entitas tersebut. Adapun Diagram ERD pada Sistem Informasi Pendafaran Pasien Berbasis SMS Gateway adalah sebagai berikut :



Gambar 4.10 Entity Relational Diagram

4.1.4.4 Struktur File

Struktur file adalah penggambaran tentang file-file dalam tabel sehingga dapat dilihat bentuk file-file tersebut baik field-fieldnya, tipe datanya, serta ukuran dari data tersebut. Berikut ini adalah struktur file pada Sistem Informasi Pendaftaran Pasien Berbasis SMS Gateway :

1. Tabel Pasien

   Nama Tabel : Pasien

   Primary Key : id_pasien

   Foreign Key : -

Tabel 4.1 Struktur File Tabel Pasien

No	Nama Field	Type	Size	Keterangan
1	id_pasien	Varchar	15	Atribut yang menyimpan ID Pasien
2	nama	Varchar	50	Atribut yang menyimpan nama dari pasien

No	Nama Field	Type	Size	Keterangan
3	alamat	Text	-	Atribut yang menyimpan alamat pasien
4	pekerjaan	Varchar	50	Atribut yang menyimpan pekerjaan pasien
5	kelamin	Varchar	5	Atribut yang menyimpan jenis kelamin Pasien
6	umur	Varchar	5	Atribut yang menyimpan umur Pasien
7	telp	Varchar	20	Atribut yang menyimpan nomor telepon pasien

2. Tabel Spesialis

   Nama Tabel : Spesialis

   Primary Key : id_spesialis

   Foreign Key : -

Tabel 4.2 Struktur File Tabel Spesialis

No	Nama Field	Type	Size	Keterangan
1	id_spesialis	Int	15	Atribut yang menyimpan ID Spesialisasi dokter
2	Spesialis	Varchar	50	Atribut yang menyimpan nama dari spesialisasi dokter

3. Tabel Dokter

   Nama Tabel : Dokter

   Primary Key : id_dokter

   Foreign Key : id_spesialis

Tabel 4.3 Struktur File Tabel Dokter

No	Nama Field	Type	Size	Keterangan
1	id_dokter	Varchar	15	Atribut yang menyimpan ID Dokter

No	Nama Field	Type	Size	Keterangan
2	nama_dokter	Varchar	50	Atribut yang menyimpan nama lengkap dokter
3	nick	Varchar	20	Atribut yang menyimpan inisial dokter
4	id_spesialis	Int	15	Atribut yang menyimpan kode spesialisasi
5	max_antrian	Int	3	Atribut yang menyimpan maksimal antrian per hari

4. Tabel Jadwal

   Nama Tabel : Jadwal

   Primary Key : id_jadwal

   Foreign Key : id_dokter

Tabel 4.4 Struktur File Tabel Jadwal Dokter

No	Nama Field	Type	Size	Keterangan
1	id_jadwal	Int	5	Atribut yang menyimpan ID jadwal dokter
2	id_dokter	Varchar	15	Atribut yang menyimpan Kode Dokter
3	hari_praktek	Varchar	60	Atribut yang menyimpan hari praktek dokter
4	jam_praktek	Int	15	Atribut yang menyimpan jam praktek dokter

5. Tabel Antrian

   Nama Tabel : Antrian

   Primary Key : id_antrian

   Foreign Key : id_dokter, id_pasien

Tabel 4.5 Struktur File Tabel Antrian

No	Nama Field	Type	Size	Keterangan
1	id_antrian	Varchar	15	Atribut yang menyimpan ID antrian pasien
2	id_dokter	Varchar	15	Atribut yang menyimpan kode dokter
3	urutan	Int	10	Atribut yang menyimpan nomor urutan
4	tgl_antrian	Date	-	Atribut yang menyimpan tanggal antrian
5	id_pasien	Varchar	15	Atribut yang menyimpan kode pasien
6	status_antrian	Varchar	5	Atribut yang menyimpan status antrian

6. Tabel Antrian Poli

   Nama Tabel : Antrian_Poli

   Primary Key : id_antrian

   Foreign Key : id_spesialis, id_pasien

Tabel 4.6 Struktur File Tabel Antrian Poli

No	Nama Field	Type	Size	Keterangan
1	id_antrian	Varchar	15	Atribut yang menyimpan ID antrian pasien
2	id_spesialis	Int	15	Atribut yang menyimpan ID Spesialis
3	urutan	Int	10	Atribut yang menyimpan nomor urutan
4	tgl_antrian	Date	-	Atribut yang menyimpan tanggal antrian
5	id_pasien	Varchar	15	Atribut yang menyimpan kode pasien
6	status_antrian	Varchar	5	Atribut yang menyimpan status antrian

GAMMU juga memiliki databasenya sendiri dimana struktur dari tabel-tabel yang sistem GAMMU butuhkan tidak dapat diubah. Berikut adalah struktur dari tabel yang dibutuhkan GAMMU

1. Tabel Inbox

   Nama Tabel : Inbox

   Primary Key : ID

   Foreign Key : -

Gambar 4.7 Struktur File Tabel Inbox

No	Nama Field	Type	Size	Keterangan
1	ID	Int	10	Nomor ID SMS
2	UpdateInDB	Timestamp	-	Waktu record diupdate
3	ReceivingDateTime	Timestamp	-	Waktu sms diterima
4	Text	Text	-	Encode isi SMS
5	SenderNumber	Varchar	20	Nomor Pengirim
6	Coding	enum('Default_No_Compression', 'Unicode_No_Compression', '8bit', 'Default_Compression', 'Unicode_Compression')	-	-
7	UDH	Text	-	-
8	SMSCNumber	Varchar	20	No.Center Provider
9	Class	Int	11	Tipe SMS
10	TextDecoded	Varchar	160	Decode isi SMS
11	RecipientID	Text	-	Nama GAMMU
12	Processed	enum(‘false’,’true’)	-	Status SMS

2. Tabel Outbox

Nama Tabel : Outbox

Primary Key : ID

Foreign Key : -

Tabel 4.8 Struktur File Tabel Outbox

No	Nama Field	Type	Size	Keterangan
1	ID	Int	10	No ID SMS
2	UpdateInDB	Timestamp	-	Waktu record diupdate
3	InsertIntoDB	Timestamp	-	Waktu masuk ke database
4	SendingDateTime	Timestamp	-	Waktu pengiriman
5	Text	Text	-	Encode isi SMS
6	DestinationNumber	Varchar	20	Nomor Tujuan
7	Coding	enum('Default_No_Compression', 'Unicode_No_Compression', '8bit', 'Default_Compression', 'Unicode_Compression')	-	-
8	UDH	Text	-	-
9	SMSCNumber	Varchar	20	No.Center Provider
10	Class	Int	11	Tipe SMS
11	TextDecoded	Varchar	160	Decoded isi SM
12	RelativeValidity	Int	11	-
13	Multipart	enum(‘false’,’true’)	-	Short/Long SMS
14	SenderID	Varchar	255	ID Nomor Pengirim
15	SendingTimeOut	Timestamp		Waktu pengiriman
16	DeliveryReport	enum(‘default’,’yes’,’no’)		Report

No	Nama Field	Type	Size	Keterangan
17	CreatorID	text		GAMMU

3. Tabel Outbox Multipart

   Nama Tabel : outbox_multipart

   Primary Key : ID, SequencePosition

   Foreign Key : -

Tabel 4.9 Struktur File Tabel outbox_multipart

No	Nama Field	Type	Size	Keterangan
1	ID	Int	10	No ID SMS
2	SequencePosition	Int	11	-
3	UDH	Text	-	-
4	Text	Text	-	Encode isi SMS
5	Class	Int	11	Tipe SMS
6	Coding	enum('Default_No_Compression', 'Unicode_No_Compression', '8bit', 'Default_Compression', 'Unicode_Compression')	-	-
7	TextDecoded	Varchar	160	Decode isi SMS

4. Tabel Daemons

   Nama Tabel : Daemons

   Primary Key : -

   Foreign : -

Tabel 4.10 Struktur File Tabel Daemons

No	Nama Field	Type	Size	Keterangan
1	Start	Text	-	-
2	Info	Text	-	-

5. Tabel Gammu

   Nama Tabel : Gammu

   Primary Key : -

   Foreign Key : -

Tabel 4.11 Struktur File Tabel Gammu

No	Nama Field	Type	Size	Keterangan
1	Version	Int	11	Versi Gammu

6. Tabel Phones

   Nama Tabel : Phones

   Primary Key : IMEI

   Foreign Key : -

Tabel 4.12 Struktur File Tabel Phones

No	Nama Field	Type	Size	Keterangan
1	IMEI	Varchar	35	No IMEI
2	ID	Text	-	ID Phone
3	UpdateInDB	Timestamp	-	-
4	InsertIntoDB	Timestamp	-	-

No	Nama Field	Type	Size	Keterangan
5	TimeOut	Timestamp	-	-
6	Send	enum(‘yes’,’no’)	-	-
7	Recieve	enum(‘yes’,’no’)	-	-
8	Client	Text	-	-
9	Battery	Int	11	Status Batrai
10	Signal	Int	11	Status Signal
11	Sent	Int	11	-
12	Received	Int	11	-

7. Tabel Sentitems

Nama Tabel : Sentitems

Primary Key : ID, SequencePosition

Foreign Key : -

Tabel 4.13 Struktur File Tabel Sentitems

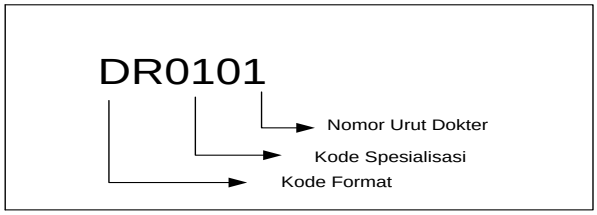
No	Nama Field	Type	Size	Keterangan
1	UpdateInDB	Timestamp	-	Waktu record diupdate
2	InsertIntoDB	Timestamp	-	Waktu record diproses
3	SendingDateTime	Timestamp	-	Waktu pengiriman
4	DeliveryDateTime	Timestamp	-	Waktu SMS Diterima
5	Text	Text	-	Encode isi SMS
6	DestinationNumber	Varchar	20	No. Tujuan
7	Coding	enum('Default_No_Compression', 'Unicode_No_Compression', '8bit', 'Default_Compression', 'Unicode_Compression')	-	-
8	UDH	Text	-	-

No	Nama Field	Type	Size	Keterangan
9	SMSCNumber	Varchar	20	-
10	Class	Int	11	Tipe SMS
11	TextDecoded	Varchar	160	Decode isi SMS
12	ID	Int	10	No ID
13	SenderID	Varchar	255	No ID Pengirim
14	SequencePosition	Int	11	-
15	Status	enum('SendingOK', 'SendingOKNoReport', 'SendingError', 'DeliveryOK', 'DeliveryFailed', 'DeliveryPending', 'DeliveryUnknown', 'Error')	-	-
16	StatusError	Int	11	-
17	TPMR	Int	11	-
18	RelativeValidity	Int	11	-
19	CreatorID	Text	-	-

4.1.4.5 Kodeifikasi

Pengkodean dibuat untuk mendefinisikan suatu objek secara singkat, dengan adanya sistem pengkodean diharapkan dapat mengklasifikasikan data, memasukkan data kedalam komputer dan untuk mengambil informasi yang dibutuhkan. Adapun kodifikasi yang telah dirancang adalah sebagai berikut:

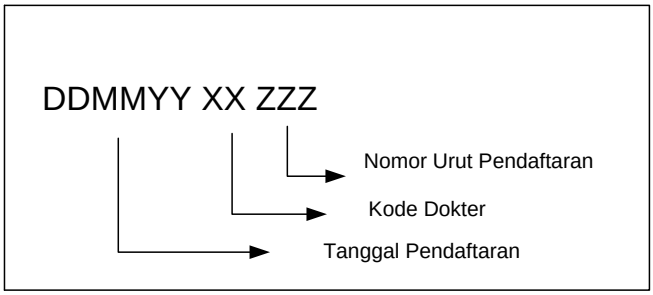
a. Kode Dokter



Contoh kode dokter:

DR0101 mengandung arti bahwa dokter praktek di dengan kode spesialisasi 01 yaitu Dokter Umum dan memiliki nomor urut pertama.

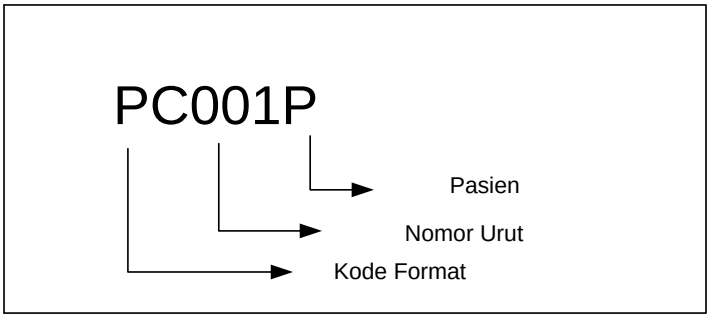
b. Kode Antrian



Contoh kode antrian:

29061402003 mengandung arti bahwa pasien mendaftar pada tanggal 29 Juni 2014 dengan dokter tujuan yaitu dokter umum dengan urutan ketiga pada hari itu. Sedangkan untuk antrian poli, kode dokter diganti dengan kode Spesiali.

c. Kode Pasien



Contoh kode pasien:

PC001 mengandung arti bahwa pasien tersebut memiliki nomor urut pertama.

4.1.5 Format Penulisan SMS

Format penulisan SMS yang dapat diproses didalam sistem informasi ini sudah ditentukan terlebih dahulu dan kemudian diimplementasikan ke dalam sistem, sehingga jika ada SMS dengan format yang tidak dikenali maka sistem tidak akan memprosesnya.

4.1.5.1 Format Penulisan SMS Keyword

SMS Keyword adalah sms yang dikirim oleh pasien ke nomor provider yang digunakan oleh sistem informasi SMS Gateway ini. SMS Keyword yang dikirim meliputi SMS Keyword untuk registrasi berobat, cek id dokter, cek informasi dokter, cek jadwal dokter, cek jadwal konsultasi dan pembatalan registrasi. SMS Keyword tidak bersifat case sensitive.

1. SMS Keyword Registrasi

   Keyword ini digunakan pada saat pasien yang sudah terdaftar ingin melakukan registrasi berobat ke dokter yang diinginkan. Format SMS dari registrasi dokter adalah:

   REG ID_DOKTER

   Contoh: REG DR0101

2. SMS Keyword Cek ID Dokter

   Keyword ini digunakan jika pasien ingin mengetahui Id Dokter dari dokter yang diinginkan. Formatnya adalah:

   DOKTER<spasi>ID

Contoh: DOKTER ID

3. SMS Keyword Cek Info Dokter

   SMS Keyword ini digunakan jika pasien ingin mengetahui spesialisasi dari dokter yang bersangkutan. Untuk menggunakan format ini pasien harus terlebih dahulu mengetahui Id Dokter yang diinginkan. Format cek info dokter adalah sebagai berikut:

   DOKTER<spasi>ID DOKTER

   Contoh: DOKTER DR0101

4. SMS Keyword Cek Jadwal Dokter

   Keyword ini dapat digunakan pasien jika pasien ingin mengetahui jadwal dokter yang diinginkan. Sama halnya dengan cek info dokter, untuk mengetahui keyword ini pasien harus mengetahui ID_DOKTER yang bersangkutan. Formatnya adalah:

   JADWAL<spasi>ID_DOKTER

   Contoh: JADWAL DR0101

5. SMS Keyword Pembatalan Registrasi

   Keyword SMS ini digunakan jika pasien telah melakukan registrasi namun ingin membatalkannya. Format keywordnya adalah sebagai berikut:

   BATAL<spasi>ID_ANTRIAN

   Contoh: BATAL 29061402003

6. SMS Cek Info Keyword

Keyword ini digunakan untuk mengetahui keyword apa saja yang digunakan untuk mengirimkan SMS. Formatnya adalah sebagai berikut:

INFO

7. SMS Registrasi Berdasarkan Poli

   Apabila pasien ingin mendaftarkan diri tanpa harus memmilih dokter yang dituju, maka dapat mengirimkan format ini, dimana pasien akan masuk antrian berdasarkan poli yang diinginkan. Berikut adalah formatnya:

   REG<spasi>ID_SPESIALIS

   Contoh: REG ANAK

4.1.5.2 Format SMS Balasan

SMS Balasan adalah respon dalam bentuk SMS yang akan dikirimkan kembali kepada pasien jika keyword yang dikirimkan benar dan proses berhasil dilakukan. Jika keyword tidak sesuai, maka sistem tidak akan mengirimkan SMS balasan.

1. SMS Balasan Registrasi

   Jika pasien mengirimkan keyword registrasi sesuai dengan format yang ditentukan, maka sistem akan merespon keyword. Format SMS balasan dari proses registrasi adalah sebagai berikut :

   ID Antrian: 16071401001 Sukses! Data anda telah didaftarkan untuk tanggal 2014/07/16, urutan anda 1. Untuk pembatalan ketik BATAL.

2. SMS Balasan Cek ID Dokter

   Pasien yang mengirimkan keyword Cek ID Dokter dengan benar maka akan mendapatkan balasan sebagai berikut:

   DR0101:Raranta

   DR0102:Yeppi

   DR0204:Ida

   DR0205:Yulia

   DR0206:Tutin

   DR0207:Rela

   DR0303:Ryan

3. SMS Balasan Cek Info Dokter

   Pasien yang mengirimkan keyword Cek Info Dokter akan mendapatkan balasan sebagai berikut:

   DR0101 dr Raranta Leopold (Dokter Umum)

4. SMS Balasan Cek Jadwal Dokter

   SMS Balasam bagi pasien yang menggunakan keyword Cek Jadwal Dokter adalah sebagai berikut:

   Jadwal Dokter:

   dr. Raranta Leopold : Senin, Rabu (08.00-10.00)

5. SMS Balasan Pembatalan Registrasi

   SMS balasan bagi pasien yang ingin membatalkan pendaftaran antrian adalah sebagai berikut:

   Antrian anda telah dibatalkan

6. SMS Balasan Info Keyword

   Apabila pasien mengirimkan SMS dengan keyword INFO maka balasannya adalah:

   Daftar: REG ID_DOKTER/REG ID_SPESIALIS

   Pembatalan: BATAL ID_REG

   Info ID Dokter: DOKTER ID

   Info Dokter: DOKTER ID_DOKTER

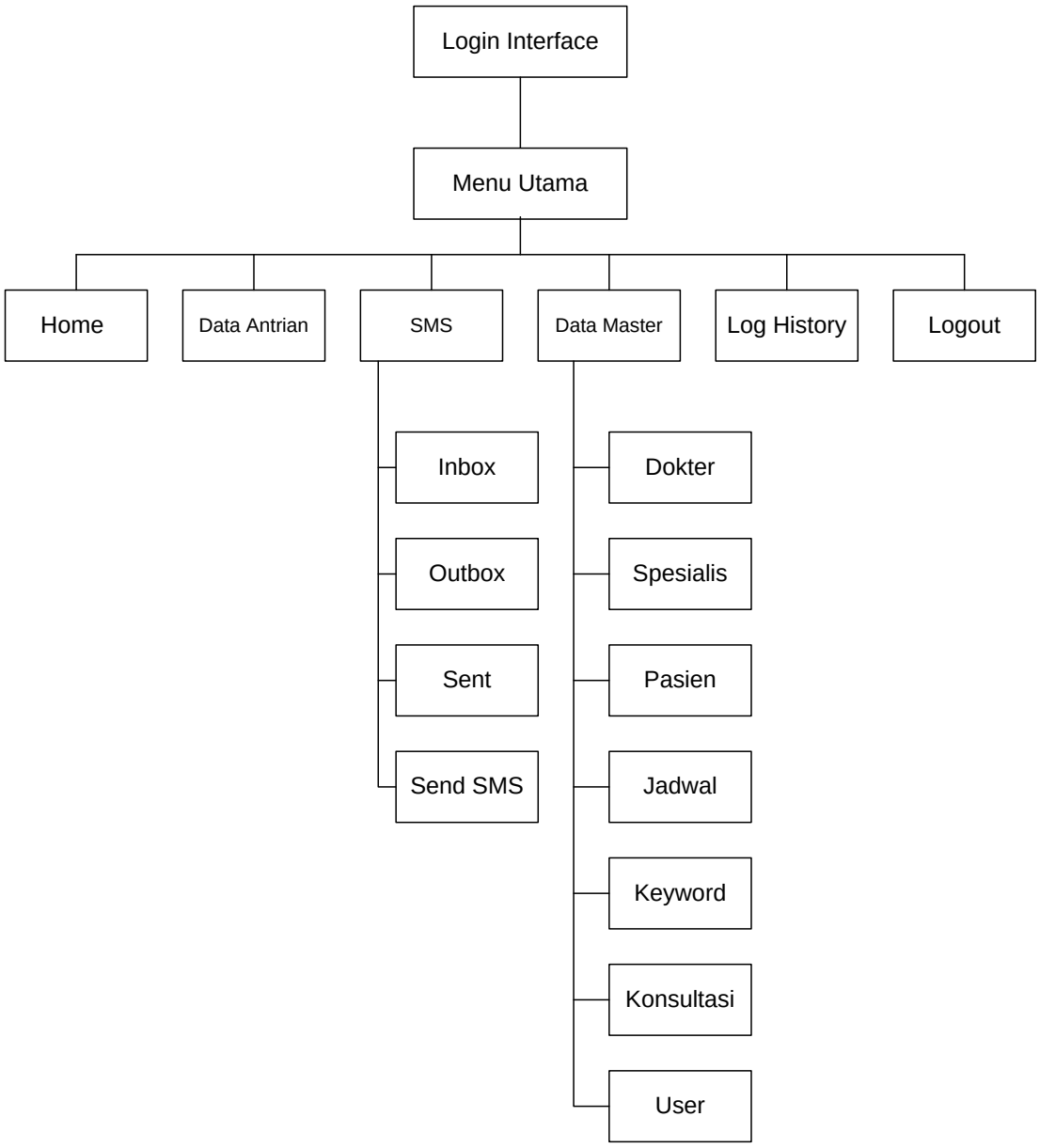
   Info Jadwal: JADWAL ID_DOKTER

4.2 Perancangan Antar Muka

Perancangan antar muka merupakan proses atau tahapan membuat gambaran dari sebuah sistem dan tampilan dari sebuah sistem yang akan dibuat. Perancangan tersebut meliputi struktur menu, perancangan input dan perancangan output.

4.2.1 Struktur Menu

Struktur menu merupakan gambaran tampilan menu secara umum untuk memudahkan pemakai dalam pencarian, adapun struktur menunya adalah sebagai berikut :



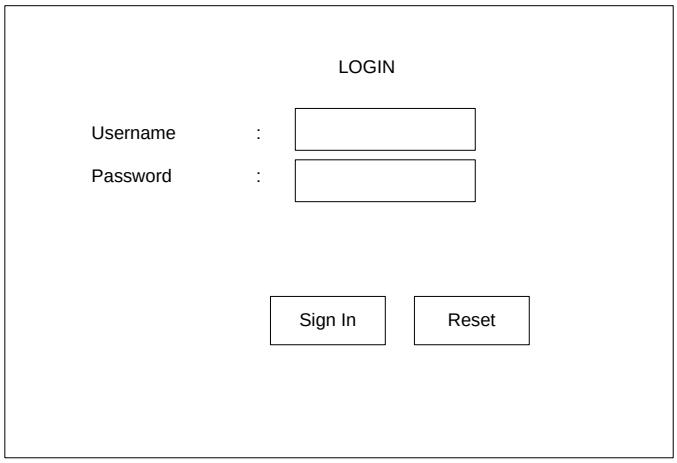
Gambar 4.11 Perancangan Struktur Menu

4.2.2 Perancangan Input

Perancangan input adalah struktur masukan yang akan diolah oleh sistem sehingga menghasilkan informasi yang dibutuhkan, adapun rancangannya adalah sebagai berikut :

1. Rancangan Input Login Admin

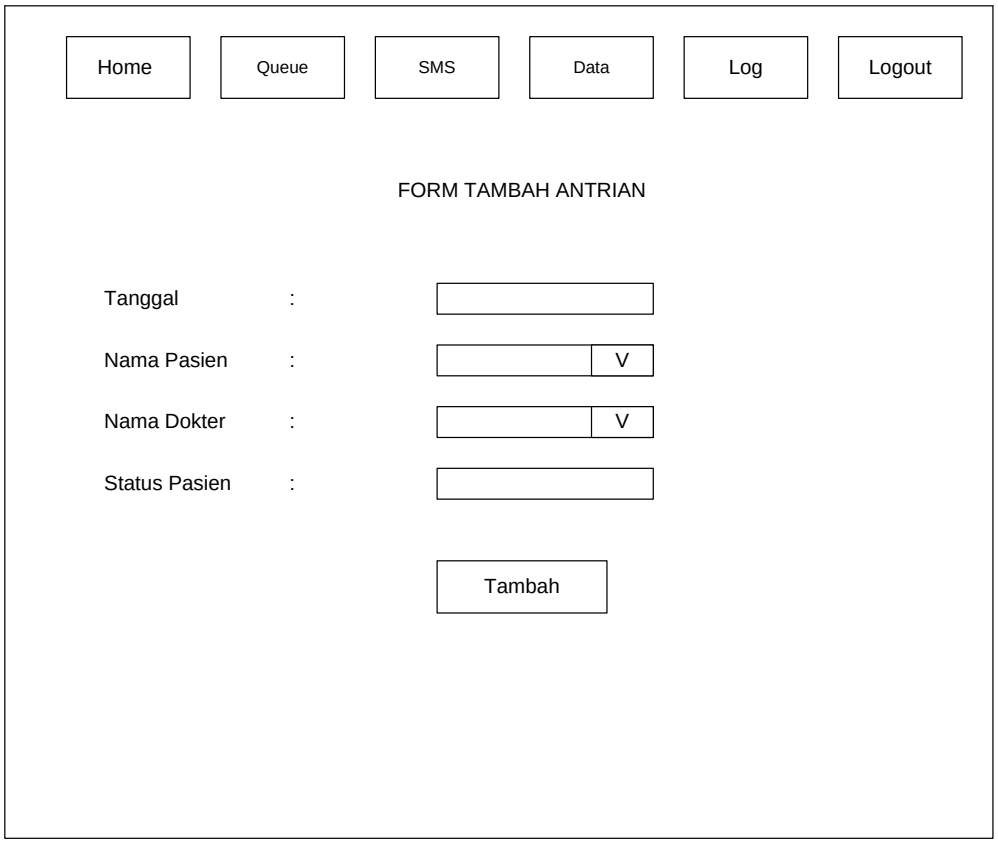
   Antarmuka ini merupakan antarmuka proses login administrator, dimana terdapat inputan username dan password.



Gambar 4.12 Perancangan Struktur Login

2. Rancangan input registrasi pasien

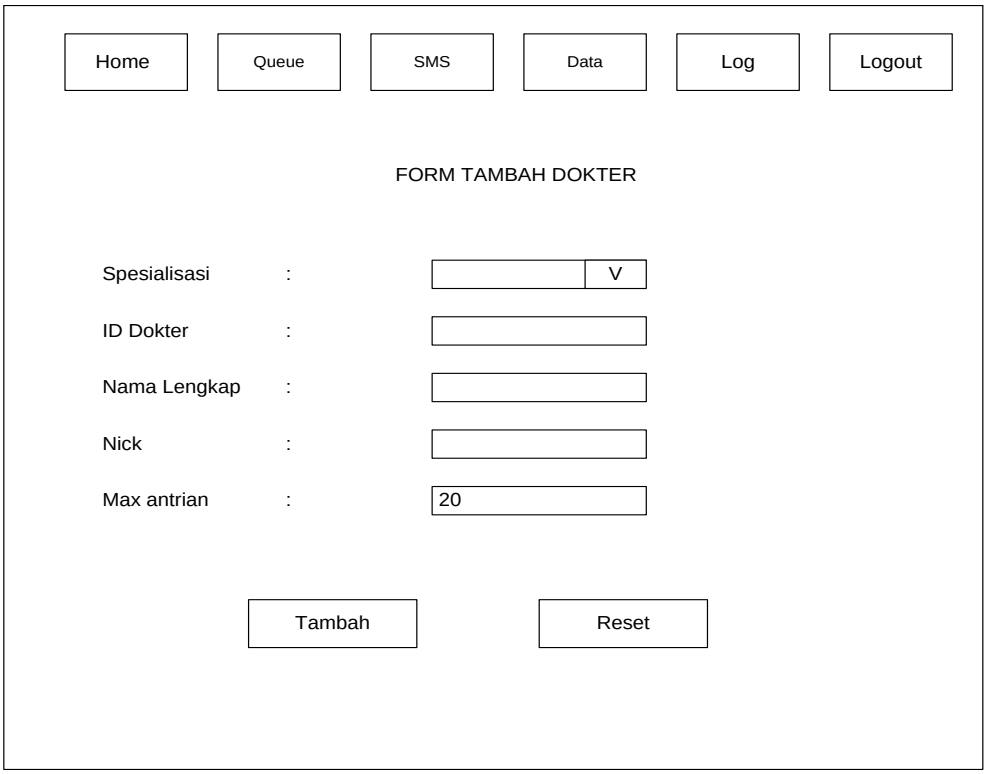
   Merupakan antarmuka untuk proses regustrasi manual, dimana seperti yang diketahui registrasi tetap dapat dilakukan secara manual dengan datang langsung ke Puskesmas Cibiru.



Gambar 4.13 Perancangan Input Registrasi Pasien

3. Rancangan Input Data Dokter

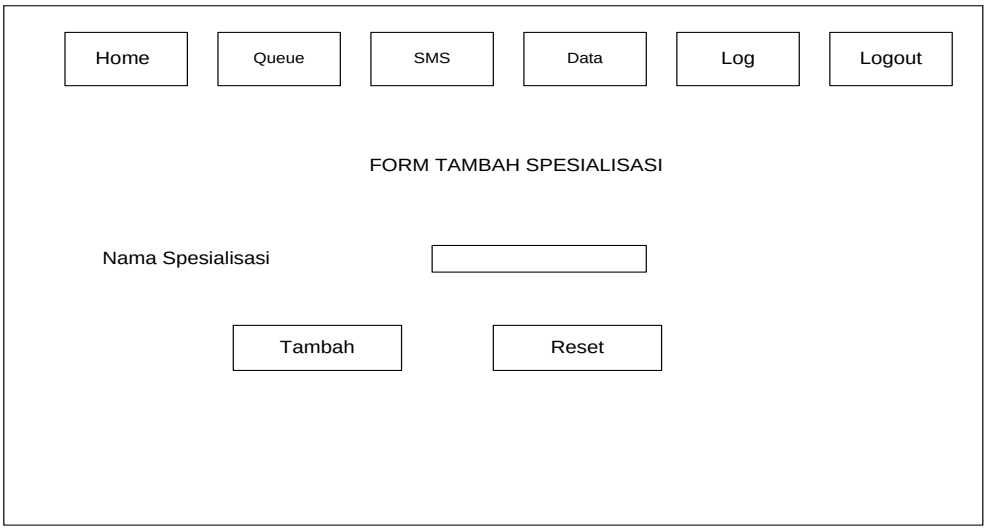
   Berikut merupakan antarmuka form input data dokter baru, digunakan jika ada dokter baru yang masuk, maka data-datanya akan disimpan melalui antarmuka ini dan berikutnya disimpan kedalam database.



Gambar 4.14 Perancangan Input Data Dokter

4. Perancangan Input Data Spesialisasi

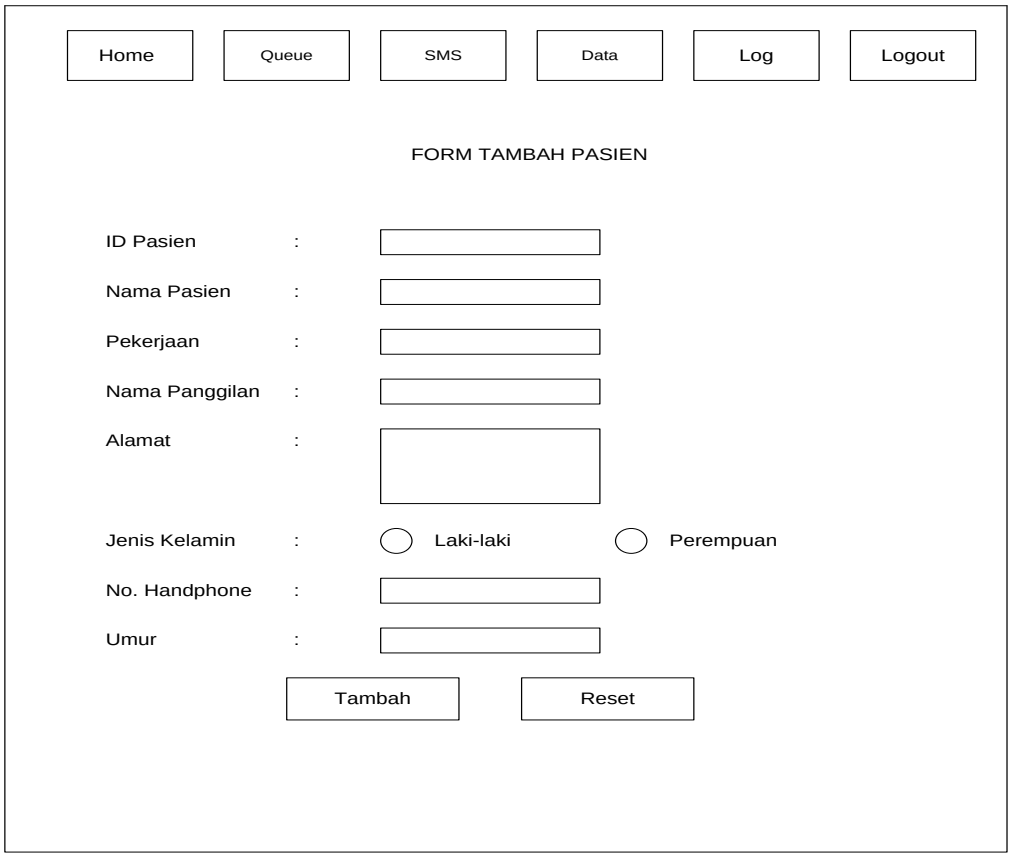
Berikut merupakan form input spesialisasi, digunakan jika ada dokter baru dengan spesialisasi yang belum ada sebelumnya.



Gambar 4.15 Perancangan Input Data Spesialis

5. Perancangan Input Data Pasien

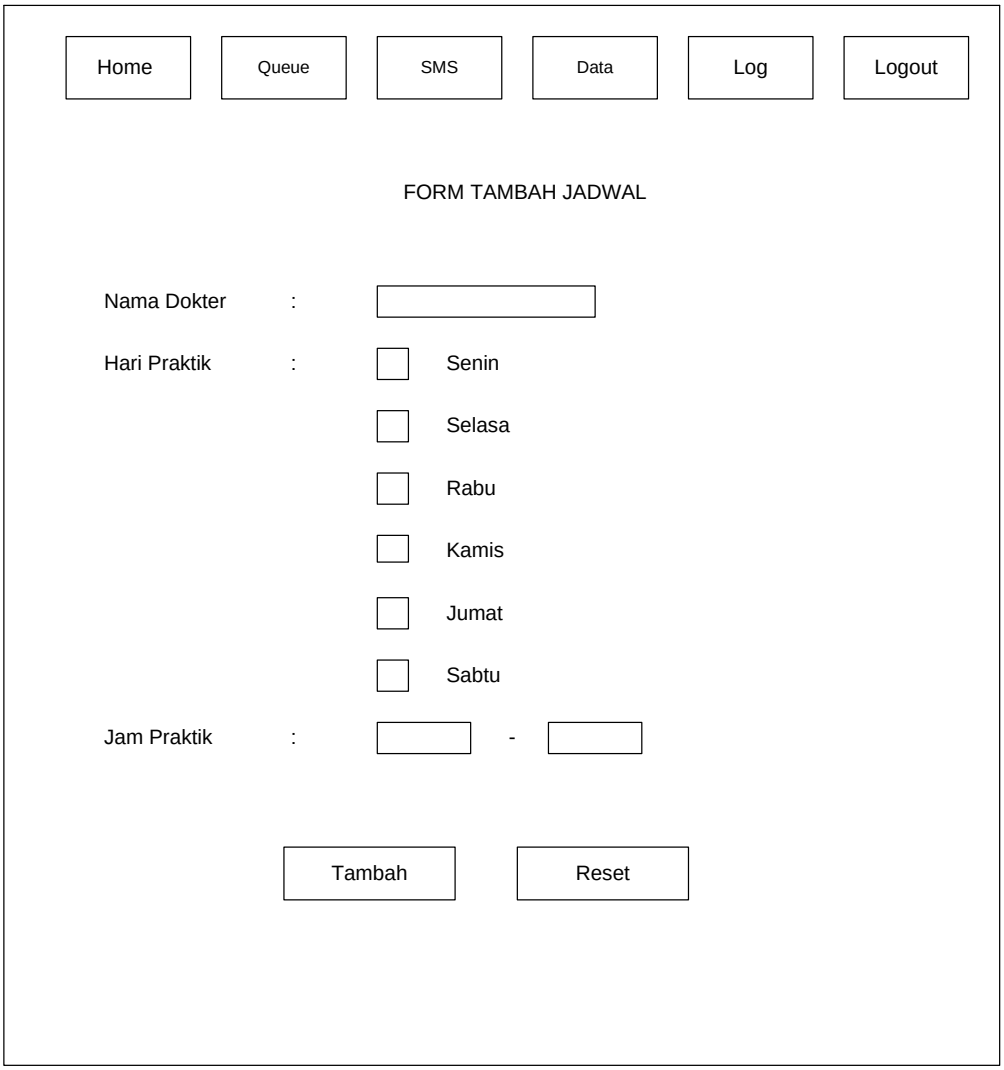
Berikut adalah antarmuka proses pasien baru. Untuk dapat menggunakan fasilitas SMS Gateway, pasien sebelumnya harus mendaftar secara manual langsung ke puskesmas. Melalui form ini akan di inputkan data-data pasien baru beserta nomor *handphone*. Maka dengan ini, pasien tersebt dapat menggunakan fasilitas SMS Gateway.



Gambar 4.16 Perancangan Input Data Pasien

6. Perancangan Input Data Jadwal Dokter

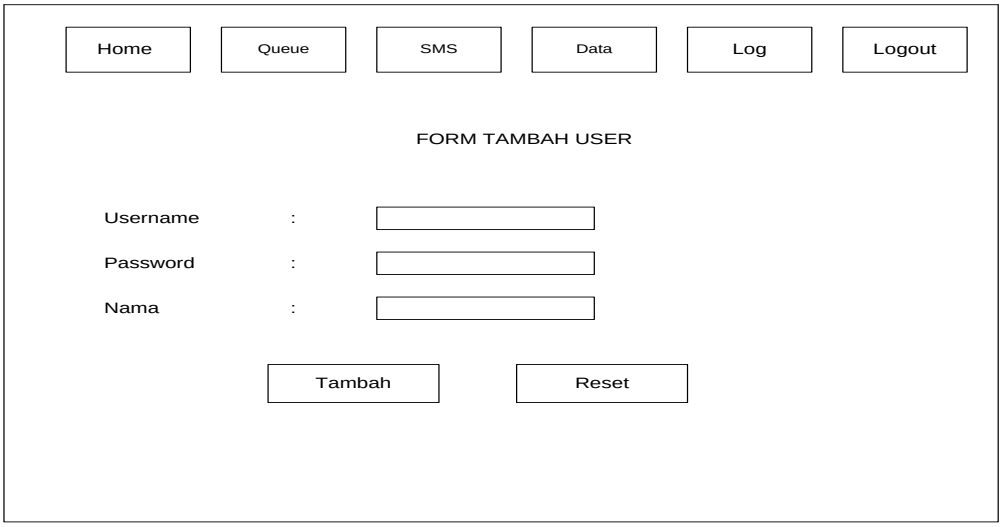
Berikut ini merupakan form input data jadwal dokter. Informasi hari praktek dan jam praktek dokter disimpan melalui form input ini dan kemudian dapat diakses oleh pasien terdaftar melalui fasilitas SMS Gateway



Gambar 4.17 Perancangan Input Data Jadwal Dokter

7. Perancangan Input User

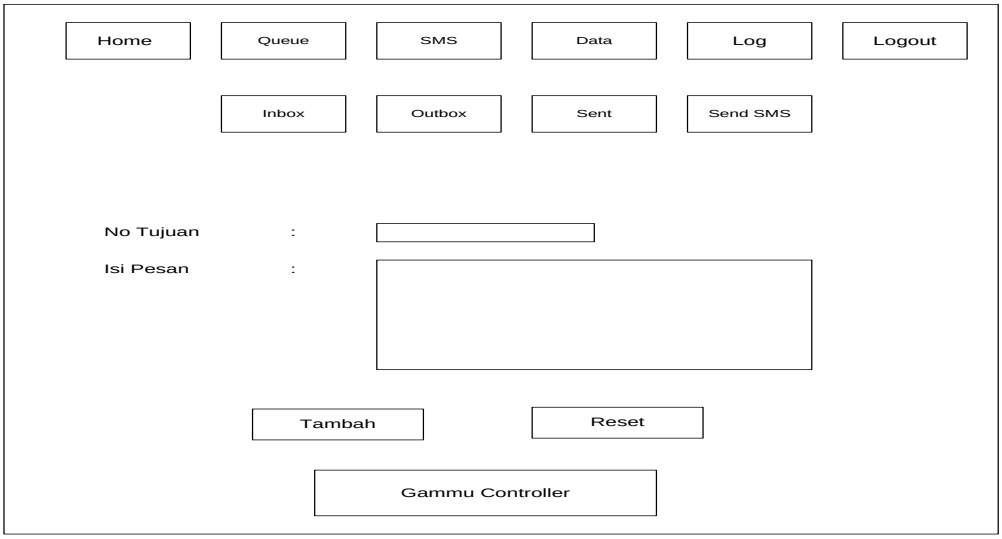
Antarmuka berikut digunakan untuk menambah administrator baru.



Gambar 4.18 Perancangan Input Data User

8. Perancangan Input Kirim SMS

   Berikut adalah tampilan dari input kirim SMS:



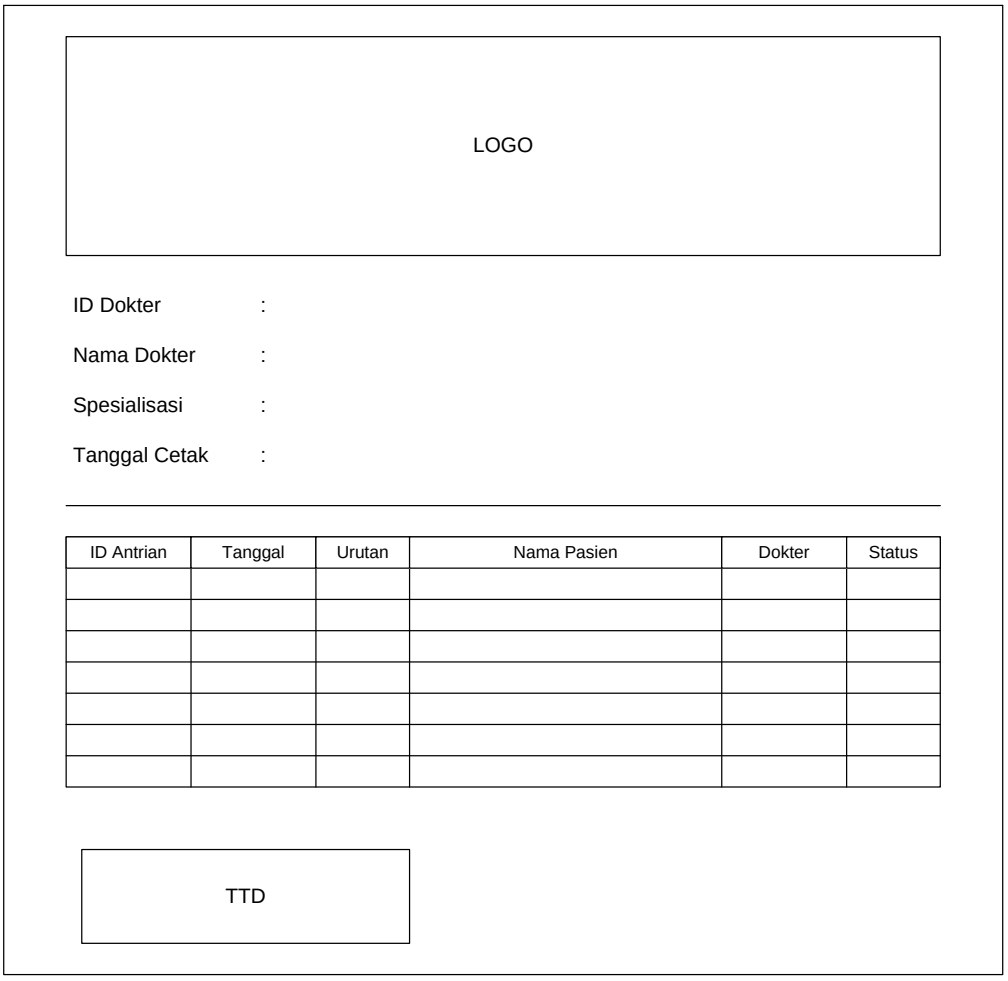
Gambar 4.19 Perancangan Input Kirim SMS

4.2.3 Perancangan Output

Perancangan output merupakan bentuk tampilan keluaran berupa laporan laporan dari sistem. Berikut perancangan output dari sistem yang sudah dibuat :

1. Output Data Antrian

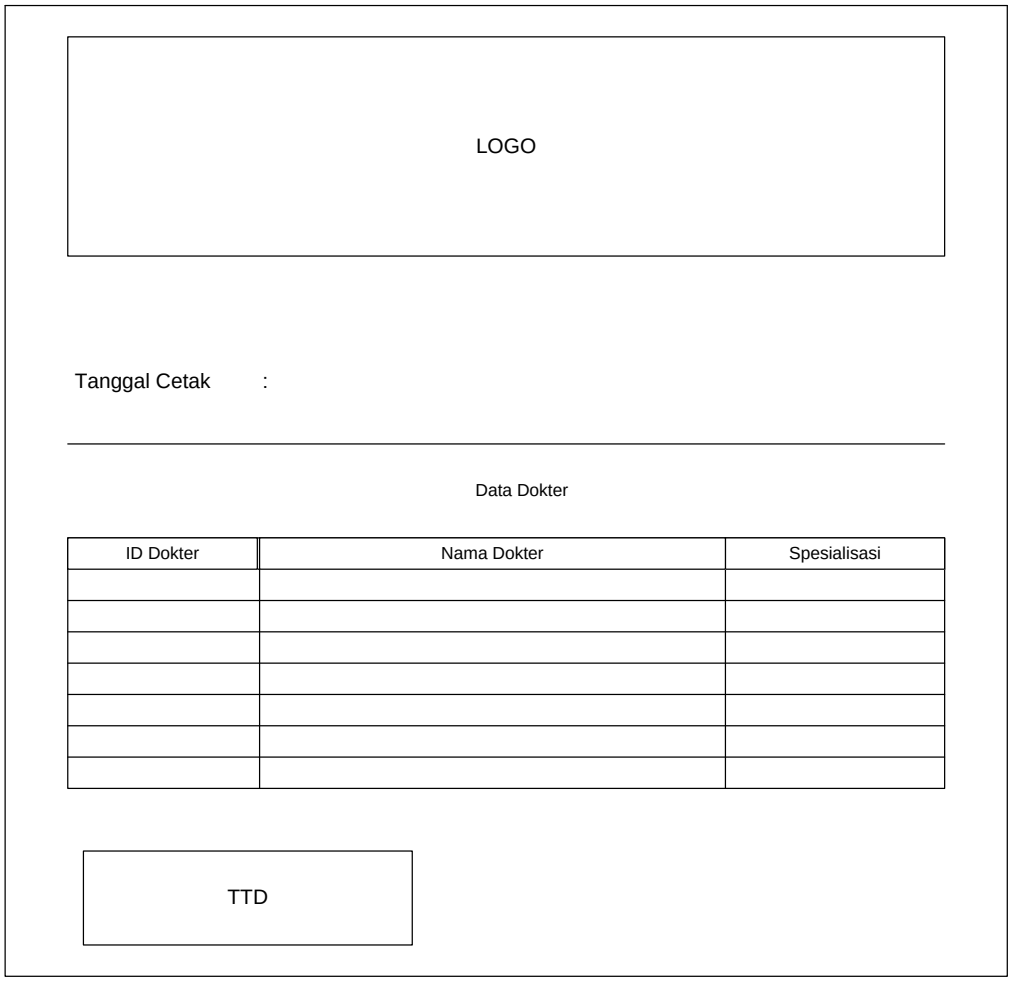
   Adalah hasil cetakan dari data antrian yang masuk, dicetak perhari menurut dokter yang bersangkutan. Berikut adalah rancangannya:



Gambar 4.20 Perancangan Output Data Antrian

2. Output Data Dokter

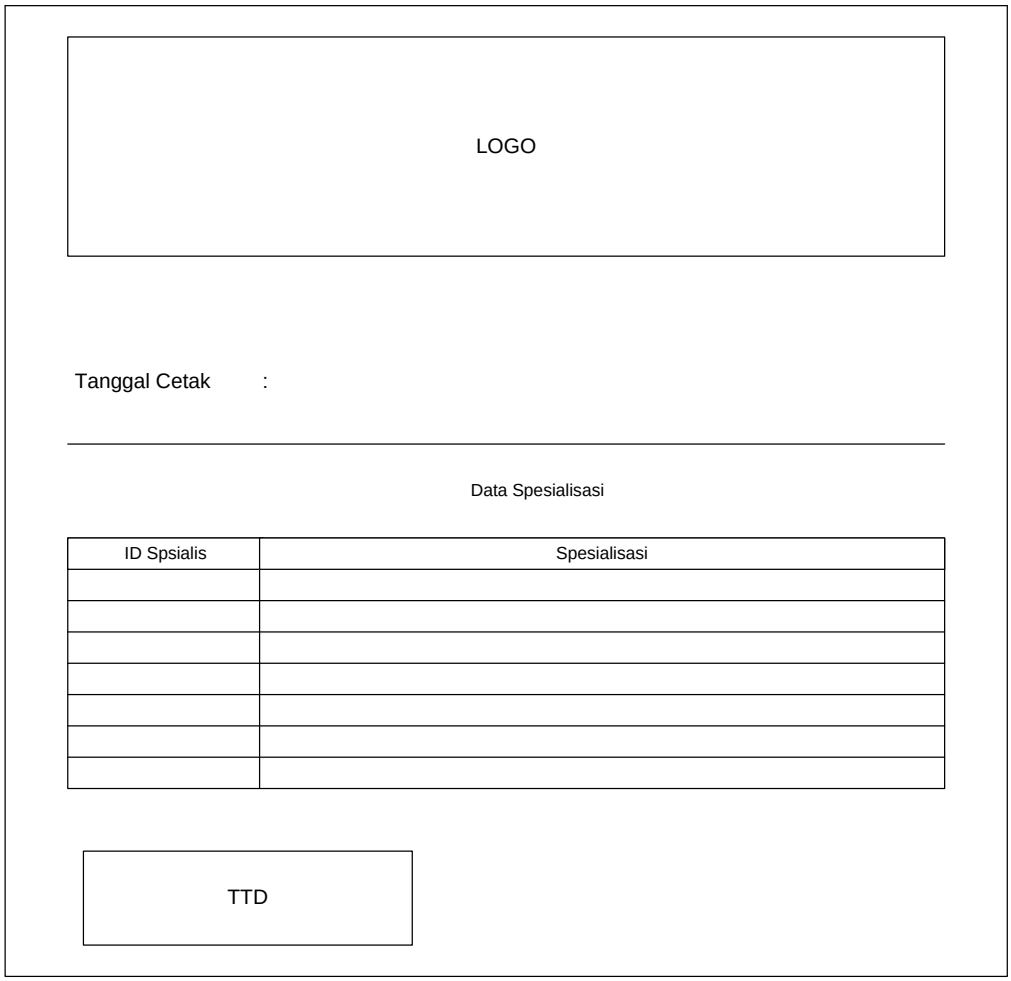
Output data dokter adalah hasil cetakan dari data dokter yang ada di database Puskesmas Cibiru. Berikut adalah rancangannya:



Gambar 4.21 Perancangan Output Data Dokter

3. Output Data Spesialisasi

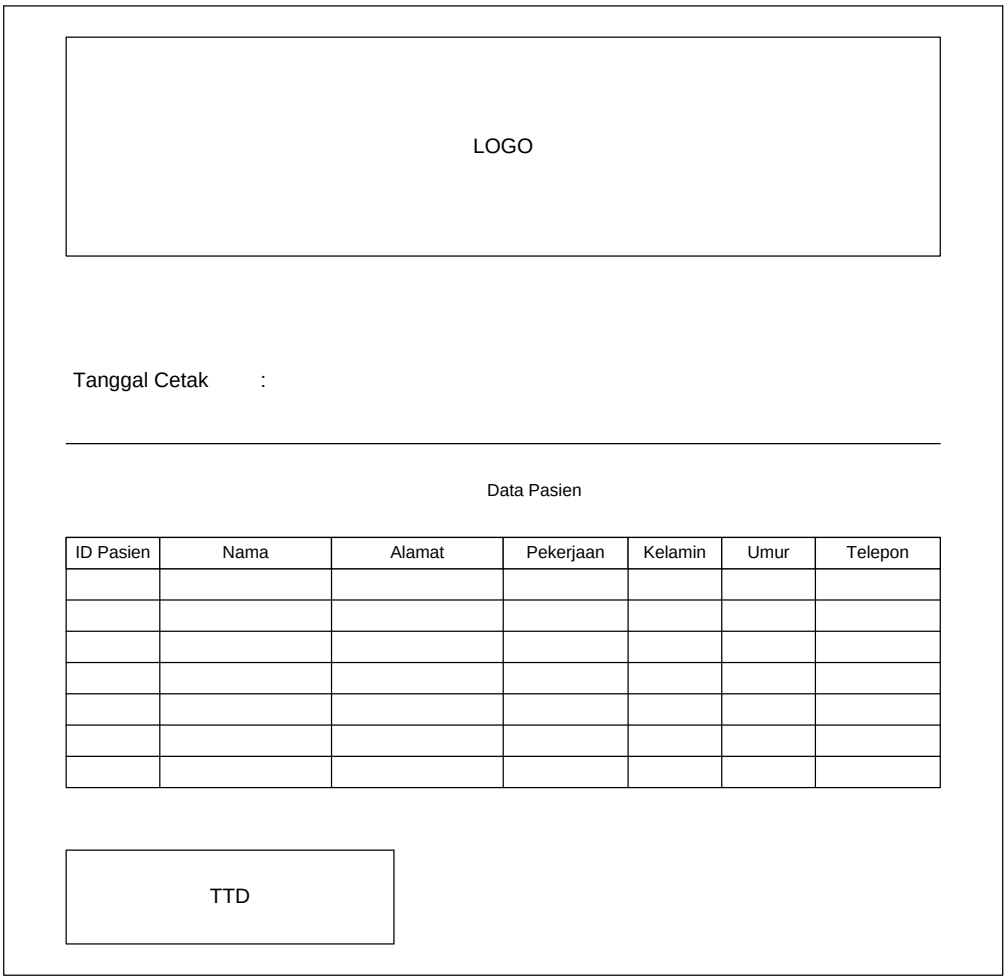
Merupakan hasil cetakan dari data spesialisasi yang ada. Berikut adalah rancangannya:



Gambar 4.22 Perancangan Output Data Spesialis

4. Output Data Pasien

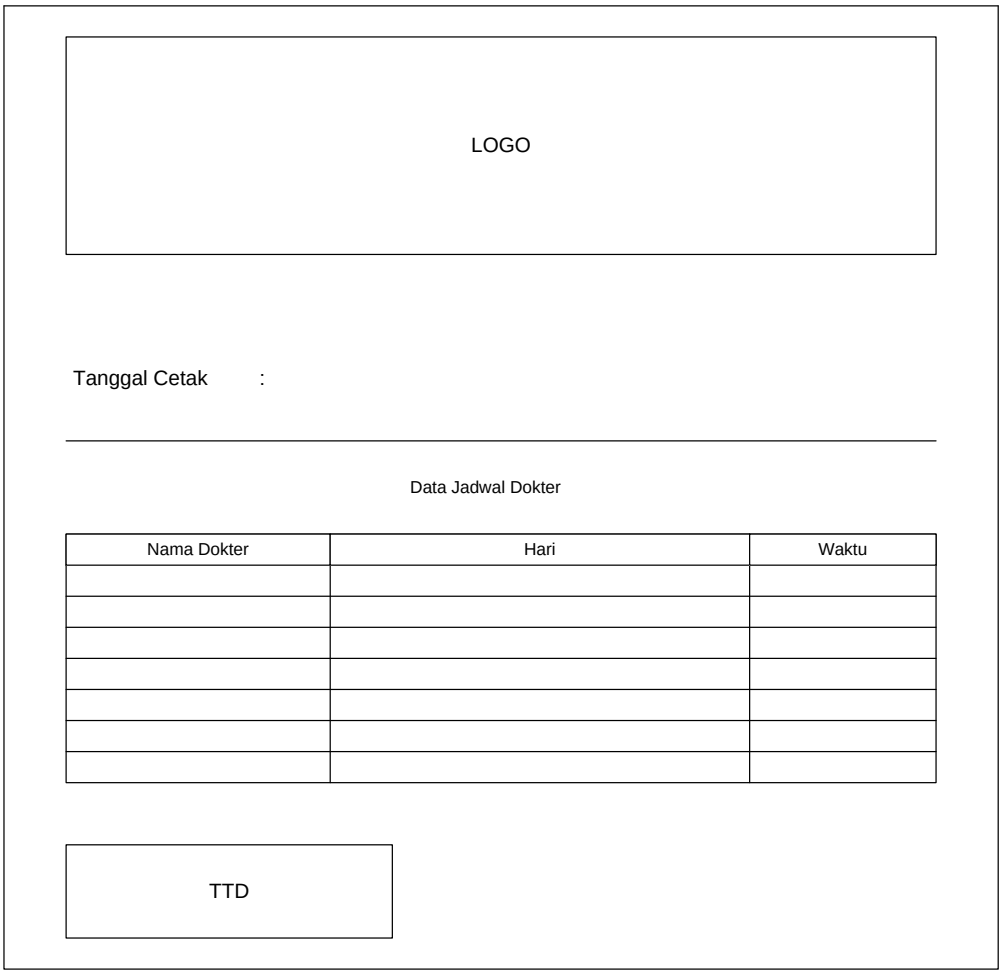
   Adalah hasil cetak dari data pasien yang ada di Puskesmas Cibiru. Berikut adalah rancangannya:



Gambar 4.23 Perancangan Output Data Pasien

5. Output Data Jadwal

   Merupakan output dari jadwal dokter yang ada. Berikut adalah rancangannya:



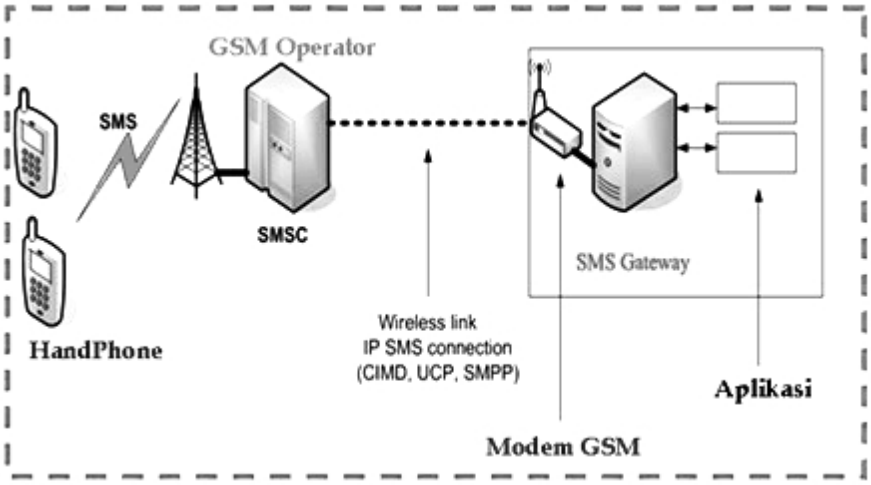
Gambar 4.24 Perancangan Output Data Jadwal Dokter

4.3 Perancangan Arsitektur Jaringan

Pesan SMS dibuat oleh *handphone* atau alat lainnya (komputer). Peralatan ini dapat mengirimkan dan menerima pesan SMS melalui komunikasi jaringan GSM. Peralatan-peralatan tersebut minimal mempunyai satu nomor MSISDN, yang disebut *Short Messaging Entities* (SME). SME merupakan *Starting Points* (*Source*) dan *End Points* (*Receiver*) untuk pesan SMS. Keduanya

akan selalu berkomunikasi dengan SMSC (*Short Message Service Center*) dan tidak akan berkomunikasi langsung antara keduanya.

Berdasarkan aturan dari jalur telekomunikasi, dapat dikategorikan menjadi dua jenis pesan SMS, yaitu Pesan *Mobile-Originated* (MO) dan Pesan *Mobile-Terminated* (MT). Pesan MO dikirimkan oleh *handphon*e kepada SMSC sedangkan pesan MT adalah pesan yang diterima *handphone*. Kedua pesan tersebut telah diset berlainan selama transmisi. Untuk lebih jelasnya secara penggambaran dapat dilihat pada gambar berikut :



Gambar 4.25 Arsitektur Jaringan SMS Gateway

(Sumber : http://aplikasi-software.co.id/product/10/51/Aplikasi-SMS-Gateway/)

4.4 Implementasi

Tahap implementasi adalah tahap yang dilakukan setelah tahap perancangan selesai dilaksanakan. Tujuan yang ingin dicapai dari tahapan

implementasi ini adalah perancangan sistem yang telah dibuat dapat dioperasikan sesuai dengan yang dirancang pada tahapan perancangan sistem.

4.4.1 Batasan Implementasi

Dalam tahapan implementasi perangkat lunak Sistem Informasi Pendaftaran Pasien berbasis SMS Gateway ini terdapat beberapa hal yang menjadi batasan dalam proses implementasi. Adapun batasan-batasan tersebut adalah sebagai berikut :

1. Sistem informasi ini difokuskan hanya pada proses pendaftaran pasien, dan tidak sampai kepada tahapan rekam medisnya. Sistem informasi ini ditujukan untuk bagian pendaftaran pada Puskesmas Cibiru.
2. Bahasa pemograman yang digunakan untuk mengembangkan sistem informasi ini adalah bahasa pemograman PHP, dengan perangkat lunak *web server*-nya adalah Apache. Basis data yang digunakan dalam implementasi sistem ini adalah MySQL.

4.4.2 Implementasi Perangkat Lunak

Untuk mendukung kelancaran pengembangan sistem informasi pendaftaran pasien berbasis SMS gateway ini selain diperlukan perangkat keras juga diperlukan perangkat lunak. Dimana perangkat lunak yang digunakan sebagai pendukung sistem informasi ini adalah sebagai berikut :

1. Sistem Operasi : Microsoft Windows 7 Ultimate

2. Web Server : Apache/2.2.21 (Win32) mod_ssl/2.2.21 OpenSSL/1.0.0e PHP/5.3.8 mod_perl/2.0.4 Perl/v5.10.1
3. Database Server : MySQL 5.5.16
4. Script Server : PHP
5. Design Enterprise : Adobe Dreamwaver CS3, Notepad++
6. Browser : Mozilla Firefox 30.0
7. Mobile Management : GNU All Mobile Management Utilities (GAMMU)

4.4.3 Implementasi Perangkat Keras

Untuk dapat menjalankan sistem yang dirancang maka dibutuhkan suatu perangkat keras sebagai penunjangnya. Adapun beberapa perangkat keras yang dibutuhkan adalah sebagai berikut :

1. Processor Interl Pentium
2. RAM 3 GB
3. Mouse, Keyboard, Monitor sebagai peralatan antarmuka
4. Modem Huwaei E1550

4.1.4 Implementasi Basis Data

Pembuatan basis data menggunakan bahasa SQL, Implementasi basisdatanya dalam SQL adalah sebagai berikut:

1. Struktur Tabel Antrian

```
DROP TABLE IF EXISTS `antrian`;
CREATE TABLE IF NOT EXISTS `antrian` (
`id_antrian` varchar(15) NOT NULL,
`id_dokter` varchar(15) default NULL,
`urutan` int(10) default NULL, `tgl_antrian`
date default NULL, `id_pasien`  varchar(15)
default NULL, `status_antrian` varchar(5)
default NULL, PRIMARY KEY (`id_antrian`), KEY
`id_dokter` (`id_dokter`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

2. Struktur Tabel Antrian Poli

```
CREATE TABLE IF NOT EXISTS `antrian_poli` (
  `id_antrian` varchar(15) NOT NULL,
  `id_spesialis` varchar(15) DEFAULT NULL,
  `urutan` int(10) DEFAULT NULL,
  `tgl_antrian` date DEFAULT NULL,
  `id_pasien` varchar(15) DEFAULT NULL,
  `status_antrian` varchar(5) DEFAULT NULL,
  PRIMARY KEY (`id_antrian`),
  KEY `id_dokter` (`id_spesialis`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

3. Struktur Tabel Dokter

```
CREATE TABLE IF NOT EXISTS `dokter` (
  `id_dokter` varchar(15) NOT NULL,
  `nama_dokter` varchar(50) NOT NULL,
```

```
  `nick` varchar(20) NOT NULL,
  `id_spesialis` int(15) NOT NULL,
  `max_antrian` int(3) DEFAULT NULL,
  PRIMARY KEY (`id_dokter`),
  KEY `id_dokter` (`id_dokter`),
  KEY `id_spesialis` (`id_spesialis`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

4. Struktur Tabel Pasien

```
DROP TABLE IF EXISTS `antrian`;
CREATE TABLE IF NOT EXISTS `pasien` (
`id_pasien` varchar(15) NOT NULL, `nama`
varchar(50) default NULL, `alamat` text,
`pekerjaan` varchar(50) NOT NULL, `kelamin`
varchar(5) default NULL, `umur` varchar(5)
default NULL, `tlp` varchar(20) default NULL,
PRIMARY KEY (`id_pasien`), UNIQUE KEY
`no_pasien` (`id_pasien`) ) ENGINE=MyISAM
DEFAULT CHARSET=latin1;
```

5. Struktur Tabel Spesialis

```
DROP TABLE IF EXISTS `antrian`;
CREATE TABLE IF NOT EXISTS `spesialis` (
`id_spesialis` int(15) NOT NULL
auto_increment, `Spesialis` varchar(50) NOT
NULL, PRIMARY KEY (`id_spesialis`) )
ENGINE=MyISAM  DEFAULT CHARSET=latin1
AUTO_INCREMENT=12 ;
```

6. Struktur Tabel Jadwal

```
DROP TABLE IF EXISTS `antrian`;
CREATE TABLE IF NOT EXISTS `jadwal` (
`id_jadwal` int(5) NOT NULL auto_increment,
`id_dokter` varchar(15) default NULL,
`hari_praktek` varchar(60) default NULL,
`jam_praktek` varchar(15) default NULL,
PRIMARY KEY (`id_jadwal`), UNIQUE KEY
`id_jadwal` (`id_jadwal`) ) ENGINE=MyISAM
DEFAULT CHARSET=latin1 AUTO_INCREMENT=13 ;
```

7. Struktur Tabel Spesialis

```
CREATE TABLE IF NOT EXISTS `konsultasi` (
`id_konsul` int(5) NOT NULL AUTO_INCREMENT,
  `nama_konsul` varchar(30) NOT NULL,
  `hari_konsul` varchar(100) NOT NULL,
  `tempat_konsul` varchar(30) NOT NULL,
  PRIMARY KEY (`id_konsul`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=2 ;
```

Seperti yang telah dijelaskan pada pembahasan sebelumnya bahwa GAMMU memiliki struktur databasenya sendiri. Adapun struktur table-tabel yang diperlukan GAMMU adalah sebagai berikut:

1. Struktur Tabel Daemons

```
DROP TABLE IF EXISTS `daemons`;
CREATE TABLE IF NOT EXISTS `daemons` ( `Start`
text NOT NULL, `Info` text NOT NULL )
ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

2. Struktur Tabel Gammu

```
DROP TABLE IF EXISTS `gammu`;
IF NOT EXISTS `gammu` ( `Version` int(11) NOT
NULL default '0' ) ENGINE=MyISAM DEFAULT
CHARSET=utf8;
```

3. Struktur Tabel Inbox

```
DROP TABLE IF EXISTS `inbox`;
CREATE TABLE IF NOT EXISTS `inbox` (
`UpdatedInDB` timestamp NOT NULL default
CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP,
`ReceivingDateTime` timestamp NOT NULL default
     '0000-00-00 00:00:00', `Text` text NOT
NULL, `SenderNumber` varchar(20) NOT NULL
default '', `Coding`
     enum('Default_No_Compression','Unicode_No_
Compressi
on','8bit','Default_Compression','Unicode_Compr
```

```
essi      on')      NOT      NULL      default
'Default_No_Compression',
     `UDH`   text   NOT   NULL,   `SMSCNumber`
varchar(20)  NOT  NULL  default  '',  `Class`
int(11)  NOT  NULL  default  '-1',  `TextDecoded`
varchar(160) NOT NULL default '', `ID` int(10)
unsigned NOT NULL auto_increment, `RecipientID`
text NOT NULL, `Processed` enum('false','true')
NOT NULL default
     'false',
     PRIMARY KEY (`ID`) ) ENGINE=MyISAM DEFAULT
CHARSET=utf8 AUTO_INCREMENT=94 ;
```

4. Struktur Tabel Outbox

```
DROP TABLE IF EXISTS `outbox`;
CREATE  TABLE  IF  NOT  EXISTS  `outbox`  (
`UpdatedInDB`  timestamp  NOT  NULL  default
CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP,
`InsertIntoDB`  timestamp  NOT  NULL  default
'000000-00 00:00:00',
     `SendingDateTime`   timestamp   NOT   NULL
default
```

```
     '0000-00-00   00:00:00',   `Text`   text,
`DestinationNumber`   varchar(20)   NOT   NULL
default
     '',
     `Coding`
enum('Default_No_Compression','Unicode_No_Compr
essi
on','8bit','Default_Compression','Unicode_Compr
essi      on')      NOT      NULL      default
'Default_No_Compression',
     `UDH` text, `Class` int(11) default '-1',
`TextDecoded` varchar(160) NOT NULL default '',
`ID` int(10) unsigned NOT NULL auto_increment,
`MultiPart`    enum('false','true')    default
'false', `RelativeValidity` int(11) default '-
1',   `SenderID`   varchar(255)   default   NULL,
`SendingTimeOut` timestamp NULL default '0000-
00
     00 00:00:00',
     `DeliveryReport`
enum('default','yes','no')
     default 'default', `CreatorID`
     text NOT NULL, PRIMARY KEY (`ID`),
```

```
     KEY`outbox_date`(`SendingDateTime`,`Sendin
gTimeOut`), KEY `outbox_sender` (`SenderID`) )
ENGINE=MyISAM        DEFAULT        CHARSET=utf8
AUTO_INCREMENT=194 ;
```

5. Struktur Tabel Outbox Multipart

```
DROP TABLE IF EXISTS `outbox_multipart`;
CREATE TABLE IF NOT EXISTS `outbox_multipart` (
`Text` text, `Coding`
enum('Default_No_Compression','Unicode_No_Compr
ession','8bit','Default_Compression','Unicode_C
ompressi     on')     NOT     NULL     default
'Default_No_Compression',
     `UDH` text, `Class` int(11) default '-1',
`TextDecoded` varchar(160) default NULL, `ID`
int(10)   unsigned   NOT   NULL   default   '0',
`SequencePosition`  int(11)  NOT  NULL  default
'1', PRIMARY KEY (`ID`,`SequencePosition`)
     ) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

6. Struktur Tabel Phones

```
CREATE TABLE IF NOT EXISTS `phones` (
`ID` text NOT NULL,
```

```
`UpdatedInDB` timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
`InsertIntoDB` timestamp NOT NULL DEFAULT
'0000-00-00 00:00:00',
`TimeOut` timestamp NOT NULL DEFAULT '0000-00-
00 00:00:00',
`Send` enum('yes','no') NOT NULL DEFAULT 'no',
`Receive` enum('yes','no') NOT NULL DEFAULT
'no',
`IMEI` varchar(35) NOT NULL,
`Client` text NOT NULL,
`Battery` int(11) NOT NULL DEFAULT '0',
`Signal` int(11) NOT NULL DEFAULT '0',
`Sent` int(11) NOT NULL DEFAULT '0',
`Received` int(11) NOT NULL DEFAULT '0',
PRIMARY KEY (`IMEI`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

7. Struktur Tabel Sentitems

```
CREATE TABLE IF NOT EXISTS `sentitems` (
  `UpdatedInDB` timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `InsertIntoDB` timestamp NOT NULL DEFAULT
'0000-00-00 00:00:00',
  `SendingDateTime` timestamp NOT NULL DEFAULT
'0000-00-00 00:00:00',
  `DeliveryDateTime` timestamp NULL DEFAULT
NULL,
  `Text` text NOT NULL,
  `DestinationNumber` varchar(20) NOT NULL
DEFAULT '',
```

```
  `Coding`
enum('Default_No_Compression','Unicode_No_Compr
ession','8bit','Default_Compression','Unicode_C
ompression') NOT NULL DEFAULT
'Default_No_Compression',
  `UDH` text NOT NULL,
  `SMSCNumber` varchar(20) NOT NULL DEFAULT '',
  `Class` int(11) NOT NULL DEFAULT '-1',
  `TextDecoded` varchar(160) NOT NULL DEFAULT
'',
  `ID` int(10) unsigned NOT NULL DEFAULT '0',
  `SenderID` varchar(255) NOT NULL,
  `SequencePosition` int(11) NOT NULL DEFAULT
'1',
  `Status`
enum('SendingOK','SendingOKNoReport','SendingEr
ror','DeliveryOK','DeliveryFailed','DeliveryPen
ding','DeliveryUnknown','Error') NOT NULL
DEFAULT 'SendingOK',
  `StatusError` int(11) NOT NULL DEFAULT '-1',
  `TPMR` int(11) NOT NULL DEFAULT '-1',
  `RelativeValidity` int(11) NOT NULL DEFAULT
'-1',
  `CreatorID` text NOT NULL,
  PRIMARY KEY (`ID`,`SequencePosition`),
  KEY `sentitems_date` (`DeliveryDateTime`),
  KEY `sentitems_tpmr` (`TPMR`),
  KEY `sentitems_dest` (`DestinationNumber`),
  KEY `sentitems_sender` (`SenderID`),
  KEY `SenderID` (`SenderID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf;
```

4.1.5 Implementasi Antar Muka

Implementasi antar muka dilakukan dengan membuat antar muka pada form yang ada dengan Adobe Dreamweaver dan Notepad++. Setiap halaman form yang dibuat akan di bentuk sebuah file yang berekstensi *.php. File -file tersebut selanjutnya dapat di akses dan akan menjadi penghubung antar pengguna dengan sistem. Pada tahap implementasi antar muka ini, akan lebih menitikberatkan pada implementasi antar muka halaman utama yang merupakan sentral penghubung dengan antar muka yang lain.

4.1.5.1 Implementasi Halaman Login

Halaman login merupkan halaman antarmuka yang pertama kali dilihat saat memulai sistem. Antarmuka ini akan menerima inputan data login dari admin dan kemudian akan di teruskan ke halaman utama. Berikut implementasi dari antarmuka halaman login :

Tabel 4.14 Implementasi Halaman Login

Menu	Deskripsi	Nama File
Login	File untuk menangani dan menerima inputan login.	Login.php

4.1.5.2 Implementasi Halaman Utama

Form halaman utama merupakan sentral dari seluruh halaman form yang digunakan. Halaman utama ini berisi menu-menu utama yang dapat mengakses seluruh sub menu dan menjadi penghubung antara pengguna dan system

Tabel 4.15 Implementasi Struktur Menu

Submenu	Deskripsi	Nama File
Home	Menu untuk memanggil antarmuka halaman utama.	index.php
Queue	Menu untuk mengakses form-form yang terdapat pada halaman antrian pasien .	registration.php
SMS	Menu yang memiliki sub menu untuk mengakses form –form pengolahan data SMS.	sms.php
Data	Menu yang memiliki sub menu untuk mengakses data-data yang digunakan oleh sistem	master.php
Report	Menu untuk mengakses halaman report atau log history dari sistem	log.php
Sign Out	Untuk keluar dari sistem	-

4.1.5.3 Implementasi Sub Menu SMS

Sub menu SMS merupakan sub menu dari menu SMS pada halaman utama, untuk mengakses halaman pengolahan data-data SMS. Adapun implementasi sub menu SMS adalah sebagai berikut :

Tabel 4.16 Implementasi Sub Menu SMS

Submenu	Deskripsi	Nama File
Inbox	Sub menu untuk memanggil halaman pengolahan SMS masuk.	sms.php

Submenu	Deskripsi	Nama File
Outbox	Sub menu untuk memanggil halaman pengolahan SMS keluar.	outbox.php
Sent	Sub menu untuk memanggil halaman pengolahan SMS terkirim.	terkirim.php
Send	Sub menu untuk memanggil halaman pengolahan untuk mengirimkan SMS secara manual.	send.php

4.1.5.4 Implementasi Sub Menu Data

Sub menu data merupakan merupakan sub menu yang terdapat pada menu Data, dimana sub menu ini digunakan untuk memanggil dan mengakses halaman-halaman pengolahan data dokter, pasien, jadwal, spesialis, dan informasi keyword. Adapun implementasi sub menu data adalah sebagai berikut :

Tabel 4.17 Implementasi Sub Menu Data

Submenu	Deskripsi	Nama File
Dokter	Sub menu untuk memanggil halaman pengolahan data dokter.	dokter.php
Spesialis	Sub menu untuk memanggil halaman pengolahan data spesialis dokter.	spesialis.php
Pasien	Sub menu untuk memanggil halaman pengolahan pasien.	pasien.php
Jadwal	Sub menu untuk memanggil halaman pengolahan data jadwal dokter	jadwal.php
Keyword	Sub menu untuk memanggil halaman informasi keyword.	keyword.php

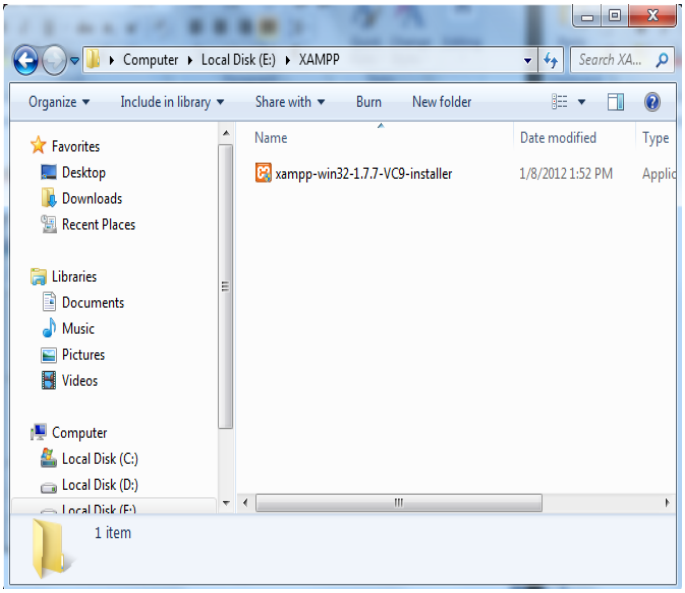
Submenu	Deskripsi	Nama File
User	Sub menu untuk memanggil halaman pengolahan data admin	user.php

4.1.6 Implementasi Instalasi Program

Pada Implementasi Instalasi Program berikut ini akan dijelaskan mengenai langkah -langkah atau prosedur -prosedur yang dilakukan untuk mulai menggunakan sistem informasi pendaftaran pasien berbasis SMS Gateway pada klinik kesehatan Apotek Kimia farma 12 Bandung. Sebelum menggunakan sistem informasi ini, yang wajib dipersiapkan adalah databasenya, berikut akan dijelaskan tahapan instalasinya:

A. Install xampp-win32-1.7.7-VC9-installer

1. Langkah pertama adalah double-click ikon installer xampp-win32-1.7.7-VC9-installer



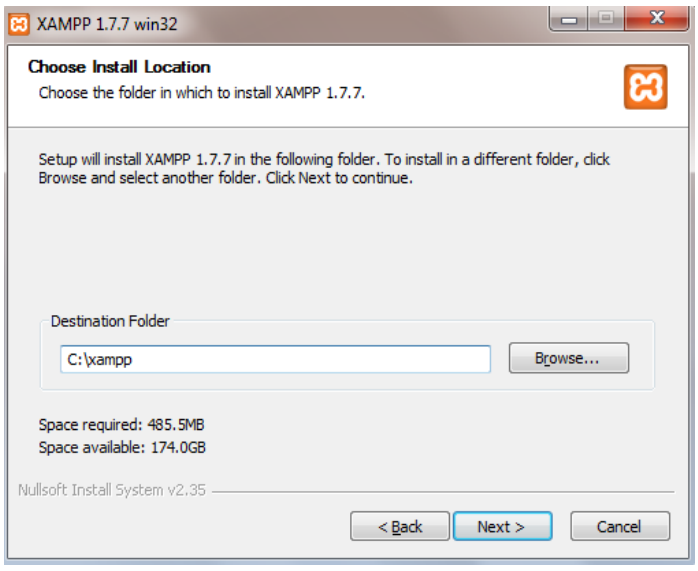
Gambar 4.26 Ikon xampp-win32-1.7.7-VC9 installer

2. Lalu akan muncul dialog box seperti gambar dibawah ini, tekan Next



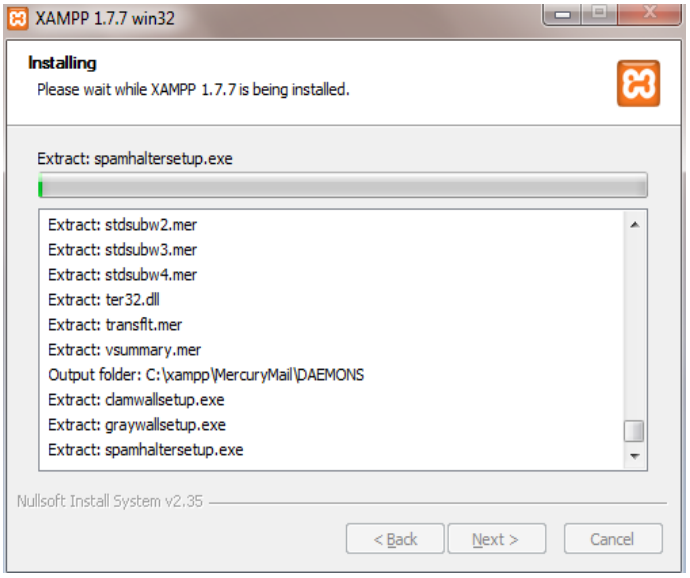
Gambar 4.27 xampp-win32-1.7.7-VC9 Setup Wizard

3. Selanjutnya adalah pemilihan folder instalasi



Gambar 4.28 xampp-win32-1.7.7-VC9 Destination Folder Setting

4. Kemudian proses instalasi akan berjalan, tunggu sampai selesai



Gambar 4.29 Proses Install xampp-win32-1.7.7-VC9

5. Setelah selesai, klik Finish

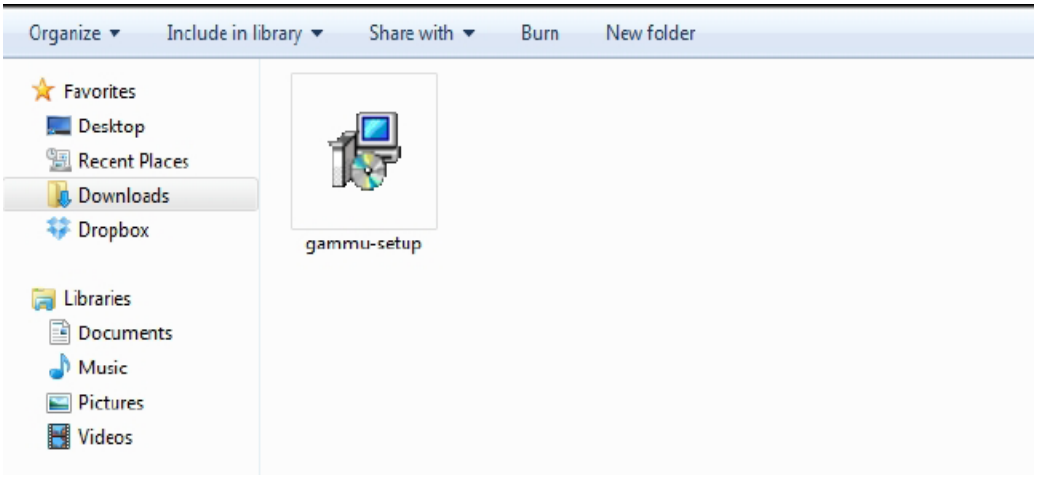


Gambar 4.30 Proses Instal xampp-win32-1.7.7-VC9 Selesai

B. Instalasi GAMUU

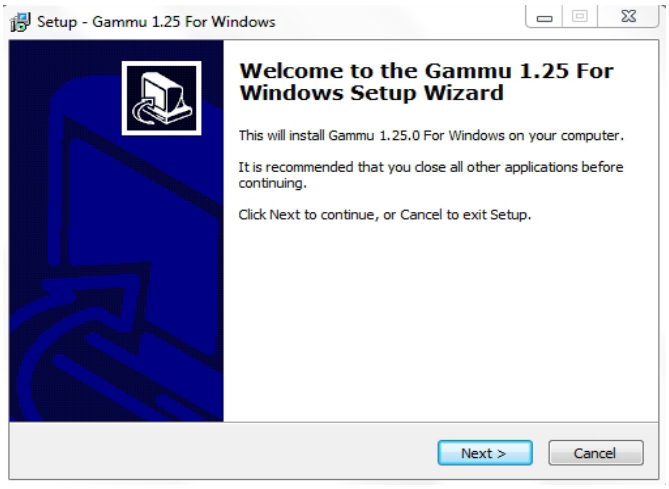
1. Langkah Pertama adalah klik kanan pada icon gammu-setup seperti

gambar dibawah ini:



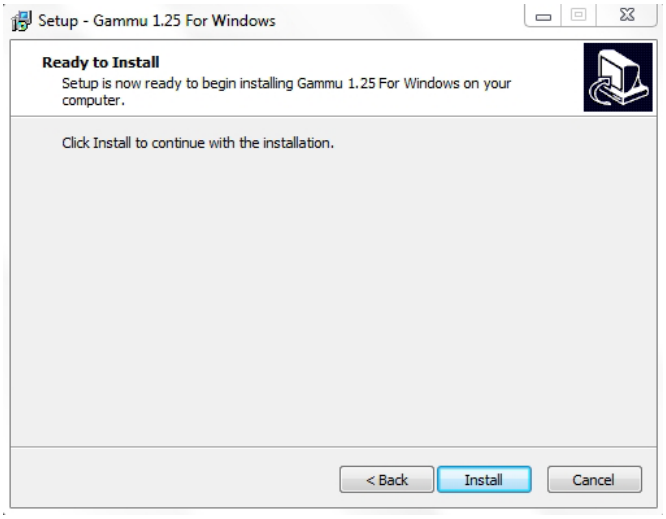
Gambar 4.31 Ikon Gammu-setup

2. Windows installation wizard untuk GAMMU akan muncul seperti dibawah ini, klik next



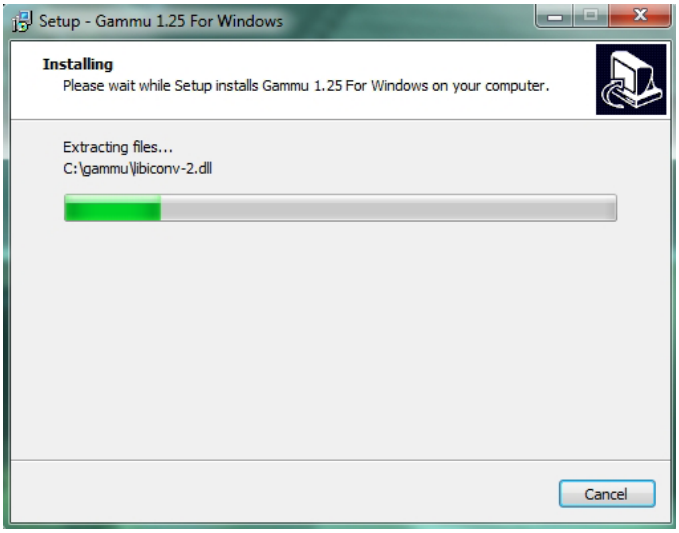
Gambar 4.32 Gammu 1.25 Setup Wizard

3. Setelah itu akan muncul konfirmasi untuk menginstall gammu, klik install



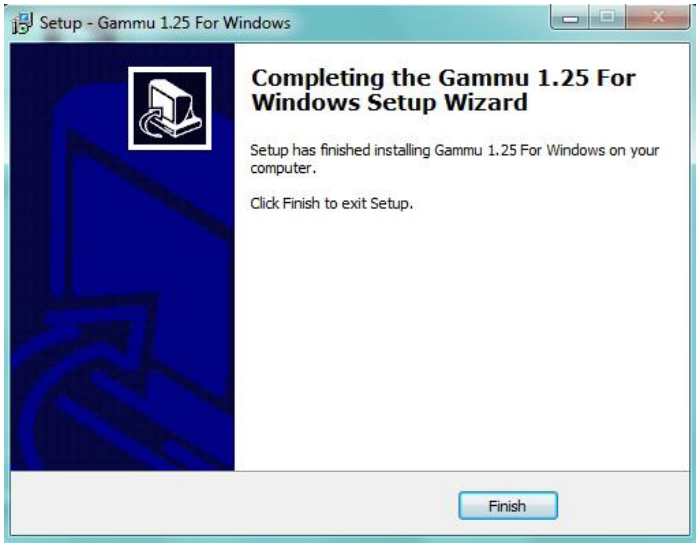
Gambar 4.33 Konfirmasi Instalasi Gammu 1.25

4. Maka Selanjutnya adalah proses instalasi gammu akan berjalan, tunggu prosesnya sampai selesai



Gambar 4.34 Proses Install Gammu 1.25

5. Setelah selesai, klik finish



Gambar 4.35 Proses Instalasi Gammu 1.25 Selesai

C. Setting Konfigurasi Gammu

Setelah terinstall, agara dapat digunakan gammu harus terlebih dahulu dijonfigurasi. Tahapannya adalah sebagai berikut:

1. Masuk Ke folder Gammu yang terdapat di C:/gammu
2. Buka file bernama GAMMURC dengan text editor (notepad):
3. Konfigurasi untuk file GAMMURC adalah sebagai berikut:
4. Selanjutnya adalah test koneksi gammu dengan modem, caranya dengan membuka folder gammu melalui command prompt lalu ketikkan: gammu identify, jika berhasil maka akan muncul reply seperti ini:
5. Berikutnya adalah konfigurasi SMS Daemon Gammu, buka file SMSDRC pada folder instalasi program Gammu menggunakan aplikasi text editor, kemudian isikan konfigurasi seperti berikut ini:

4.4.7 Penggunaan Program

Perangkat lunak yang telah selesai dibuat haruslah dapat dijalankan dengan baik oleh user. Berikut ini akan dijelaskan tata cara dalam menggunakan sistem informasi ini:

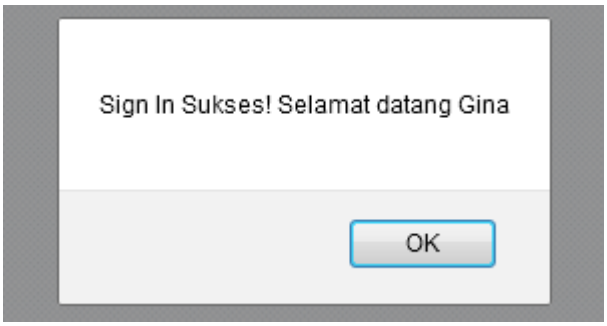
1. Form Login Administrator

Berikut adalah tampilan form login admin, dimana user meenginputkan data login username dan passwordnya saat akan memulai program.



Gambar 4.36 Form Login Admininstrator

Jika password dan username yang dimasukkan salah, maka akan muncul pesan seperti berikut:

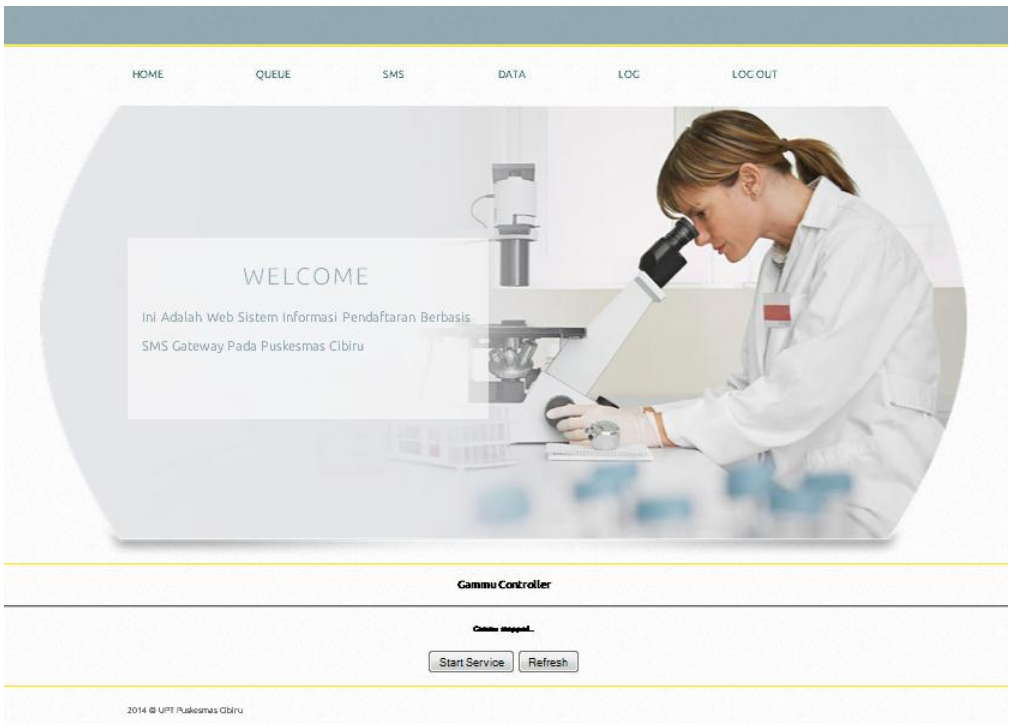


Gambar 4.37 Alert Proses Sign in Berhasil

Dan jika data login yang digunakan benar maka akan memunculkan pesan script seperti berikut ini dan halaman utama akan ditampilkan:

2. Halaman Utama

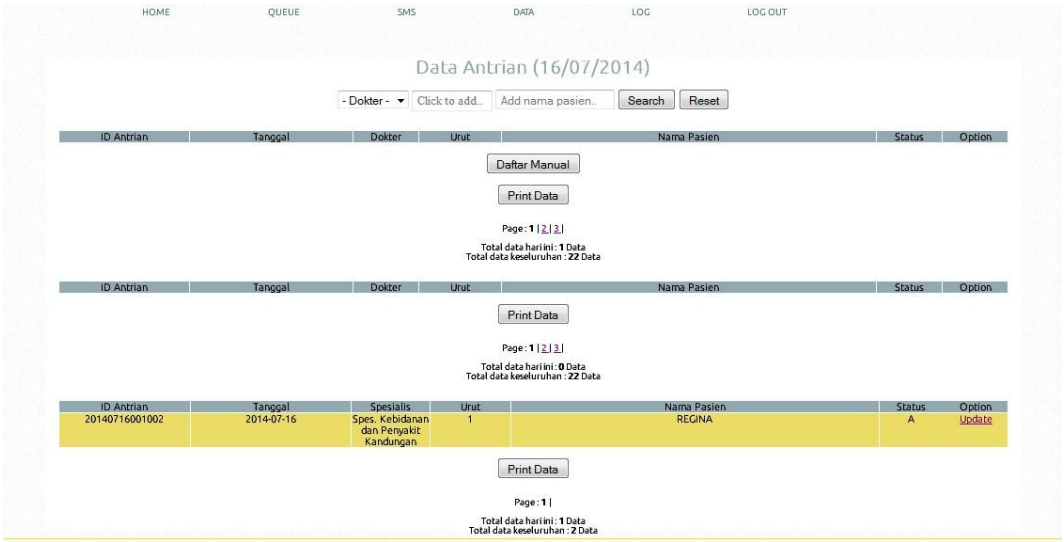
Setelah data login diterima, maka halaman utama program akan ditampilkan. Berikut tampilan halaman utama program :



Gambar 4.38 Tampilan Halaman Utama

3. Form Data Antrian

Berikut adalah tampilan form data antrian dimana data antrian pasien bisa dilihat:

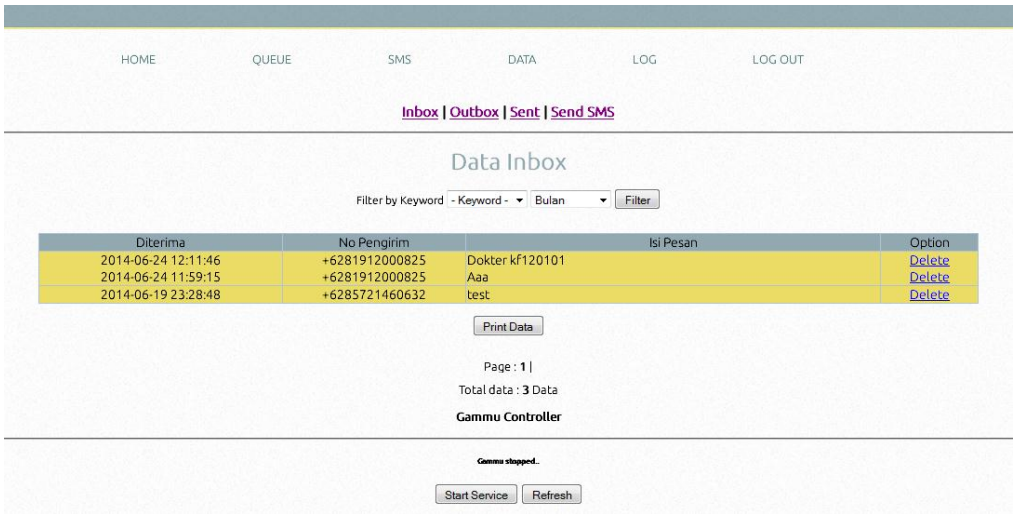


Gambar 4.39 Form Data Antrian

Pada form ini juga terdapat button “Daftar Manual” untuk memanggil halaman pendaftaran pasien yang mendaftar secara langsung ke Puskesmas. Dan juga button “Prrint Data” untuk mencetak absen pasien untuk masing-masing dokter.

4. Form Data SMS

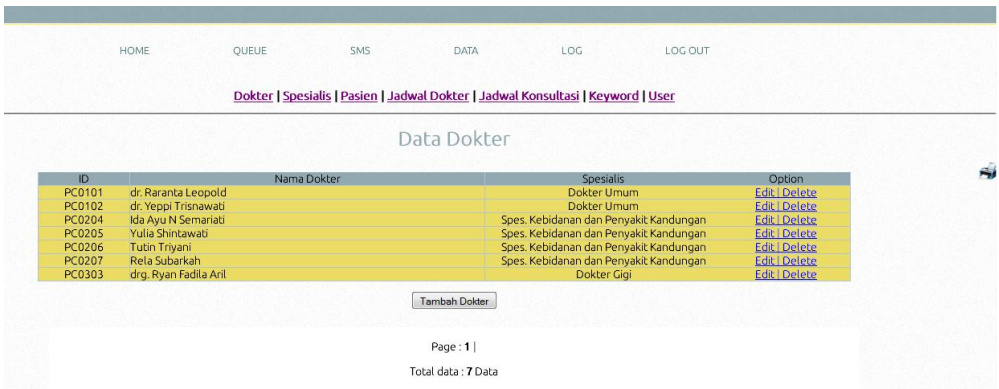
Berikut adalah form data SMS dimana semua SMS masuk, dikirim dan diterima serta submenu untuk mengirim sms dapat dilihat:



Gambar 4.40 Form Data SMS

5. Form Data Dokter

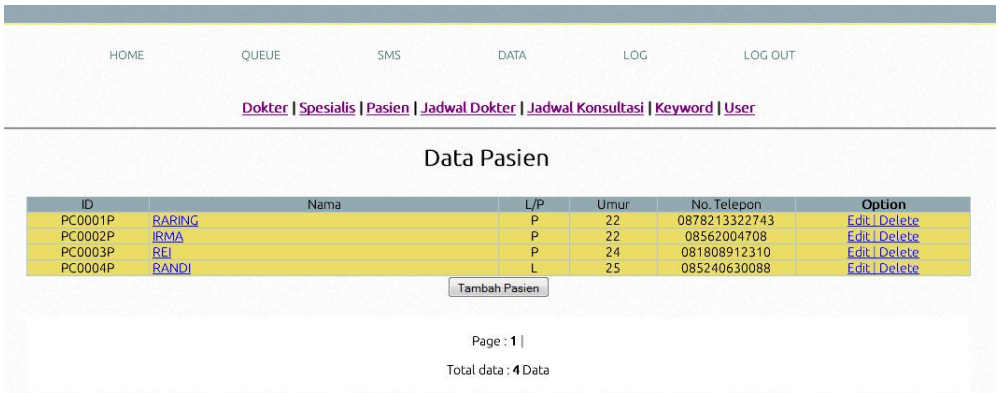
Berikut adalah tampilan form dokter dimana data dokter yang ada disajikan dengan lebih detail dan melalui halaman ini juga dapat memanggil halaman tambah dokter:



Gambar 4.41 Form Data Dokter

6. Form Data Pasien

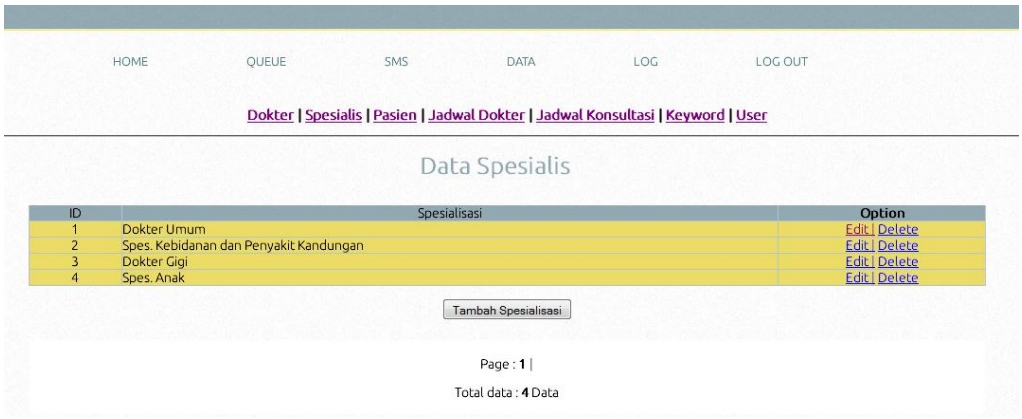
Berikut adalah tampilan form data pasien dimana form ini menampilkan biodata pasien dari Puskesmas Cibiru dengan detail:



Gambar 4.42 Form Data Pasien

7. Form Data Spesialis

   Berikut adalah tampilan form spesialis dokter yang ada di Puskesmas Cibiru:



Gambar 4.43 Form Data Spesialis

8. Form Data Jadwal Dokter

   Berikut adalah form data jadwal harian dokter:



Gambar 4.44 Form Data Jadwal Dokter

4.5 Pengujian

Tahapan pengujian merupakan bagian yang penting dalam siklus pembangunan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangkat lunak. Tujuan dari pengujian ini adalah untuk menjamin bahwa perangkat lunak yang dibangun memiliki kualitas yang baik.

4.5.1 Rencana Pengujian

Adapun rencana pengujian sistem yang akan dilakukan adalah dengan menggunakan teknik pengujian *Black Box*. Pengujian *black box* berfokus pada persyaratan fungsional dari sistem yang dibuat. Berikut rincian pengujian yang akan dilakukan :

Tabel 4.18 Rencana Pengujian

No	Komponen Sistem Yang Diuji	Butir Pengujian	Jenis Pengujian
1	Sign In Admin	Tombol Sign In	*Black Box*
		Informasi kesalahan data login	*Black Box*
2	GAMMU	Integrasi GAMMU dengan sistem	*Black Box*
		Aktivasi GAMMU (On / Off)	*Black Box*
3	Pengolahan SMS	Penerimaan SMS	*Blackbox*
		Pengiriman SMS	*Black Box*
		Eksekusi perintah menurut keyword	*Black Box*
		Pesan validasi keyword (Salah / Benar)	*Black Box*
4	Form Inputan	Pengujian dengan inputan kosong	*Black Box*
		Pengujian dengan tipe inputan yang salah	*Black Box*

4.5.2 Kasus dan Hasil Pengujian

Dalam pengujian ini penulis akan mengambil contoh kasus dari tahap pengujian program terhadap kesesuaian dengan kebutuhan sistem. Adapun kasus pengujian terhadap sistem yang dilakukan adalah sebagai berikut :

Tabel 4.19 Pengujian Kasus

No	Kasus Yang Diuji	Skenario Pengujian	Hasil Yanag Diharapkan	Hasil Pengujian
1	Sign In Admin	Sign In dengan data *username* dan *password* yang BENAR	Ketika tombol *Sign In* ditekan, maka proses validasi akan dijalankan, dengan data yang benar akan muncul pesan "Sign in sukses" dan halaman utama akan terbuka	[ ] Diterima [ ] Ditolak
		Sign In dengan data *username* dan *password* yang SALAH	Ketika tombol *Sign In* ditekan, maka proses validasi akan dijalankan, dengan data yang salah akan muncul pesan "Sign In gagal" dan halaman akana kembali ke halaman login	[ ] Diterima [ ] Ditolak
2	GAMMU	Pengujian integrasi GAMMU dengan Modem via *command prompt*	Dengan membuka gammu folder dari *command prompt*, lalu ketikan "*gammu identify"*, maka akan muncul reply spesifikasi dari modem GSM yang digunakan	[ ] Diterima [ ] Ditolak

No	Kasus yang diuji	Skenario Pengujian	Hasil yang diharapkan	Hasil pengujian
		Menghidupkan atau mematikan GAMUU dari program	Ketika tombol "Start Gammu" ditekan, status GAMMU berubah menjadi "Running" dan pada Windows Service Status "Gammu SMSD Service" akan berubah menjadi "Automatic". Dan jika tombol "Stop Gammu" ditekan akan melakukan proses kebalikannya	[ ] Diterima [ ] Ditolak
3	Form Imputan	Melakukan proses penyimpanan dengan terdapat field yang kosong	Pada saat tombol "Simpan" diklik, akan muncul pesan bahwa ada field yang kosong. Halaman tidak akan tetap dan tidak tersimpan ke database	[ ] Diterima [ ] Ditolak
		Melakukan proses penyimpanan dengan data yang tidak sesuai, seperti inputan dengan tipe huruf dimasukkan tipe angka	Pada saat tombol "Simpan" diklik, akan muncul pesan bahwa tipe inputan tidak sesuai. Halaman tidak akan tetap dan data tidak akan tersimpan ke database	[ ] Diterima [ ] Ditolak
		Melakukan proses penyimpanan dengan data yang sesuai	Pada saat klik tombol "Simpan", maka aka nada pesan bahwa data telah tersimpan	[ ] Diterima [ ] Ditolak
4.	Pengolahan Data SMS	Menguji penerimaan SMS dengan mengirimkan sembarang SMS ke nomor GSM yang digunakan sistem	SMS diterima dan tersimpan ke database secara otomatis	[ ] Diterima [ ] Ditolak

		Menguji pengiriman SMS dengan mengirimkan sembarang SMS ke nomor lain	SMS terkirim dan diterima oleh nomor penerima	[ ] Diterima [ ] Ditolak
5.	Uji keyword Cek ID Dokter	Mengirimkan SMS keyword cek id dokter dengan format yang benar	SMS diterima system dan direspon dengan mengirimkan SMS balasan berisikan ID Dokter yang terdaftar	[ ] Diterima [ ] Ditolak
		Mengirimkan SMS keyword cek id dokter dengan format yang salah	SMS diterima dan direspon dengan mengirimkan SMS balasan berisikan info kesalahan format keyword	
6.	Uji keyword cek info dokter	Mengirimkan SMS keyword cek info dokter dengan keyword yang benar	SMS diterima dan direspon dengan mengirimkan SMS balasan berisikan info dokter yang direquest	[ ] Diterima [ ] Ditolak
		Mengirimkan SMS keyword cek info dokter dengan format yang salah	SMS diterima dan direspon dengan mengirimkan SMS balasan berisikan info kesalahan format keyword atau info dokter yang direquest tidak ditemukan	[ ] Diterima [ ] Ditolak
7.	Uji keyword cek jadwal dokter	Mengirimkan SMS keyword cek jadwal dokter dengan format yang benar	SMS diterima dan direspon dengan mengirimkan SMS balasan berisikan jadwal dokter yang direquest	[ ] Diterima [ ] Ditolak

		Mengirimkan SMS keyword cek jadwal dokter dengan format yang salah	SMS diterima dan direspon dengan mengirimkan SMS balasan berisikan info kesalahan format keyword	[ ] Diterima [ ] Ditolak
8.	Uji keyword Info	Mengirimkan SMS keyword Info dengan format yang benar	SMS diterima dan direspon dengan mengirimkan SMS balasan berisikan informasi keyword yang digunakan	[ ] Diterima [ ] Ditolak
		Mengirimkan SMS keyword Info dengan format yang salah	SMS diterima dan direspon dengan mengirimkan SMS balasan berisikan info kesalahan format	[ ] Diterima [ ] Ditolak
9.	Uji keyword pembatalan registrasi	Mengirimkan SMS keyword pembatalan registrasi dengan format yang benar	SMS diterima dan direspon dengan mengirimkan SMS balasan berisikan info registrasi yang bersangkutan telah dibatalkan	[ ] Diterima [ ] Ditolak
10.	Uji keyword yang tidak terdaftar	Mengirimkan SMS keyword yang tidak dikenali atau salah penulisan	SMS diterima dan direspon dengan mengirimkan SMS balasan berisikan info bahwa keyword tidak dikenali	[ ] Diterima [ ] Ditolak

4.2.3 Kesimpulan Hasil Pengujian

Berdasarkan pengujian kasus terhadap sistem yang dibuat, penulis menyimpulkan bahwa sistem informasi pendaftaran pasien berbasis SMS Gateway ini sudah berfungsi secara fungsionalitas dan berjalan sebagaimana

semestinya. Namun untuk kecepatan SMS terkirim dan diterima bergantung pada *provider* yang digunakan dan kekuatan sinyal.