

**PENGGUNAAN SMS UNTUK OTOMATISASI  
PENGAMBILAN NOMOR ANTRIAN  
PADA KLINIK DOKTER SPESIALIS**

**SKRIPSI**

**UNIVERSITAS BRAWIJAYA**



Disusun oleh :

**RACHMAT BUDIYANTO**

**NIM 0001063347**

**DEPARTEMEN PENDIDIKAN NASIONAL**

**UNIVERSITAS BRAWIJAYA**

**FAKULTAS TEKNIK**

**JURUSAN TEKNIK ELEKTRO**

**MALANG**

**2007**



**PENGUNAAN SMS UNTUK OTOMATISASI  
PENGAMBILAN NOMOR ANTRIAN  
PADA KLINIK DOKTER SPESIALIS**

**SKRIPSI**

Disusun oleh:

**RACHMAT BUDIYANTO**

**NIM 0001063347**

**DOSEN PEMBIMBING:**

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Heru Nurwarsito, M. Kom.  
NIP. 131 879 033

Herman Tolle, ST., MT.  
NIP. 132 283 206

**PENGGUNAAN SMS UNTUK OTOMATISASI  
PENGAMBILAN NOMOR ANTRIAN  
PADA KLINIK DOKTER SPESIALIS**

Disusun oleh :

**RACHMAT BUDIYANTO**  
**NIM. 0001063347**

Skripsi ini telah diuji dan dinyatakan lulus  
pada tanggal 14 Juni 2007

**DOSEN PENGUJI**

**Ir. M. Aswin**  
**NIP. 131 879 045**

**Ir. Sutrisno**  
**NIP. 131 653 479**

**Arief Andy S., ST., M.Kom**  
**NIP. 132 231 567**

**Tri Astoto Kurniawan, ST, MT**  
**NIP. 132 304 630**

Mengetahui,  
Ketua Jurusan Teknik Elektro

**Ir. Heru Nurwasito, M.Kom**  
**NIP. 131 879 033**

## KATA PENGANTAR

Alhamdulillah, puji syukur kehadirat Allah SWT atas rahmat, nikmat, dan karunia-Nya sehingga penyusun dapat menyelesaikan Tugas Akhir yang berjudul **"Penggunaan SMS Untuk Otomatisasi Pengambilan Nomor Antrian Pada Klinik Dokter Spesialis"** dengan baik.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Program Studi Teknik Informatika dan Komputer Fakultas Teknik Universitas Brawijaya Malang.

Tak banyak yang bisa penyusun sampaikan kecuali ungkapan terima kasih, syukur, dan doa kepada berbagai pihak yang telah dengan tulus ikhlas memberikan bimbingan, arahan, dan dukungan hingga penulisan Tugas Akhir ini dapat terselesaikan. Pada kesempatan kali ini, dengan segala kesungguhan dan rasa rendah hati, penyusun mengucapkan banyak terima kasih kepada:

1. Ibu dan Bapakku serta seluruh keluarga besarku Muhammad, Achmad dan Rachmat Nur yang selalu mendoakan dan mendukung dalam menyelesaikan Tugas Akhir ini.
2. Bapak Ir. Heru Nurwarsito, M.Kom selaku Dosen Pembimbing yang telah banyak memberikan masukan dan koreksi dalam penyusunan Tugas Akhir ini.
3. Bapak Herman Tolle, ST.,MT selaku Dosen Pembimbing yang telah banyak memberikan bimbingan dan arahan terhadap penyusunan Tugas Akhir ini.
4. Bapak Ir. Heru Nurwarsito, M. Kom dan Bapak Rudy Yuwono, ST .MSc selaku Ketua dan Sekretaris Jurusan Teknik Elektro dan segenap Staf Pengajar, Administrasi, dan Perpustakaan Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.
5. Rekan-rekan angkatan 2000 (web2000) yang turut membantu kelancaran penyusunan Tugas Akhir ini.
6. Semua pihak yang tidak dapat penyusun sebutkan satu-persatu di sini, sehingga dapat terselesaikannya Tugas Akhir ini.

Semoga Allah SWT memberikan balasan kebahagiaan dan kesuksesan atas segala budi baik yang telah diberikan kepada penyusun.

Penyusun menyadari bahwa Tugas Akhir ini masih banyak kekurangannya dan masih jauh dari sempurna, untuk itu tegur sapa, saran, dan kritik yang membangun sangat penyusun harapkan. Semoga Tugas Akhir ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, Juni 2007

Penyusun



**DAFTAR ISI**

|  | Halaman  |
|--|----------|
| KATA PENGANTAR .....   | i        |
| DAFTAR ISI .....   | iii      |
| DAFTAR TABEL .....   | vii      |
| DAFTAR GAMBAR .....  | viii     |
| DAFTAR LAMPIRAN .....  | xi       |
| DAFTAR ISTILAH .....   | xii      |
| ABSTRAK .....  | xiv      |
| <b>I. PENDAHULUAN .....</b>  | <b>1</b> |
| 1.1 Latar Belakang .....   | 1        |
| 1.2 Rumusan Masalah .....  | 2        |
| 1.3 Batasan Masalah .....  | 2        |
| 1.4 Tujuan .....   | 2        |
| 1.6 Sistematika Penulisan .....                                      | 2        |
| <b>II. TINJAUAN PUSTAKA .....</b>                                    | <b>4</b> |
| 2.1 Sistem Kerja SMS .....   | 4        |
| 2.1.1 PDU ( <i>Protocol Data Unit</i> ) .....                        | 5        |
| 2.2 Sistem Informasi .....   | 11       |
| 2.2.1 <i>Data Flow Diagram Context Level</i> (Diagram Konteks) ..... | 11       |
| 2.2.2 <i>Data Flow Diagram Levelled</i> .....                        | 12       |
| 2.2.3 <i>Entity-Relationship Diagram</i> (Diagram E-R) .....         | 12       |
| 2.3 Visual Basic 6.0 .....   | 14       |
| 2.3.1 Bidang Kerja Visual Basic .....                                | 14       |
| 2.3.1.1 <i>Menu Bar</i> .....  | 15       |
| 2.3.1.2 <i>Tool Bar</i> .....  | 15       |
| 2.3.1.3 <i>Tool Box</i> .....  | 16       |
| 2.3.1.4 <i>Window Project Explorer</i> .....                         | 16       |

|         |  |    |
|---------|--|----|
| 2.3.1.5 | <i>Window Form</i> .....               | 16 |
| 2.3.1.6 | <i>Window Form Layout</i> .....        | 17 |
| 2.3.1.7 | <i>Window Code Editor</i> .....        | 17 |
| 2.3.1.8 | <i>Window Properties</i> .....         | 18 |
| 2.3.1.9 | <i>Window Immediate</i> .....          | 18 |
| 2.3.2   | Dasar Pemrograman Visual Basic .....   | 18 |
| 2.3.2.1 | Penulisan Kode Program.....            | 19 |
| 2.3.2.2 | <i>Procedure</i> .....                 | 20 |
| 2.3.2.3 | Data Dan Variabel.....                 | 21 |
| 2.3.2.4 | Operator .....                         | 23 |
| 2.3.2.5 | Struktur Kontrol.....                  | 24 |
| 2.4     | ODBC .....                             | 26 |
| 2.5     | Microsoft Access .....                 | 26 |
| 2.5.1   | <i>Table</i> .....                     | 27 |
| 2.5.2   | <i>Query</i> .....                     | 28 |
| 2.5.3   | <i>Form</i> .....                      | 28 |
| 2.5.4   | <i>Report</i> .....                    | 29 |
| 2.5.5   | <i>Data Access Page</i> .....          | 29 |
| 2.5.6   | <i>Macro</i> .....                     | 29 |
| 2.5.7   | <i>Module</i> .....                    | 29 |
| 2.6     | Mobile FBUS.....                       | 30 |
| 2.7     | Port Serial.....                       | 31 |
| 2.8     | Sistem Klinik .....                    | 32 |
| III.    | METODE PENELITIAN.....                 | 33 |
| 3.1     | Studi Literatur .....                  | 33 |
| 3.2     | Perancangan Sistem Aplikasi.....       | 33 |
| 3.3     | Implementasi Sistem.....               | 33 |
| 3.4     | Pengujian dan Analisa Program.....     | 33 |
| 3.5     | Pengambilan Kesimpulan dan Saran ..... | 34 |
| IV.     | PERANCANGAN .....                      | 35 |
| 4.1     | Analisis Sistem.....                   | 35 |

|         |   |    |
|---------|---|----|
| 4.1.1   | Analisis Kebutuhan Perangkat Lunak..... | 35 |
| 4.1.2   | Spesifikasi Kebutuhan .....             | 37 |
| 4.1.3   | Diagram Konteks .....                   | 39 |
| 4.1.4   | DFD Level 1 .....                       | 41 |
| 4.1.5   | DFD Level 2 Mengolah SMS .....          | 44 |
| 4.1.6   | DFD Level 2 Memasukkan Data Pasien..... | 46 |
| 4.1.7   | DFD Level 2 Membuat Laporan.....        | 49 |
| 4.2     | Perancangan Sistem .....                | 51 |
| 4.2.1   | Perancangan Desain Arsitektur.....      | 51 |
| 4.2.2   | Perancangan Komponen .....              | 55 |
| 4.2.3   | Perancangan Basis Data.....             | 59 |
| 4.2.3.1 | Struktur Tabel .....                    | 60 |
| 4.2.3.2 | Relasi Tabel .....                      | 63 |
| 4.2.4   | Perancangan User Interface.....         | 64 |
| 4.2.4.1 | Form Login .....                        | 65 |
| 4.2.4.2 | Form Pasien .....                       | 66 |
| 4.2.4.3 | Form Antrian.....                       | 66 |
| 4.2.4.4 | Form Rekam Medis.....                   | 67 |
| 4.2.4.1 | Form Laporan.....                       | 68 |
| V.      | IMPLEMENTASI.....                       | 70 |
| 5.1     | Proses Login.....                       | 70 |
| 5.2     | Proses Pasien.....                      | 71 |
| 5.2.1   | Proses Simpan Data Pasien.....          | 71 |
| 5.2.2   | Proses Edit Data Pasien .....           | 71 |
| 5.2.3   | Proses Delete Data Pasien.....          | 72 |
| 5.3     | Proses Rekam Medis.....                 | 73 |
| 5.3.1   | Proses Simpan Rekam Medis.....          | 73 |
| 5.3.2   | Proses Cetak Rekam Medis .....          | 74 |
| 5.4     | Proses SMS .....                        | 75 |
| 5.4.1   | Proses Koneksi Handphone .....          | 75 |
| 5.4.2   | Proses Membaca Inbox.....               | 76 |

|       |   |     |
|-------|---|-----|
| 5.5   | Proses Mencetak Laporan .....                     | 78  |
| VI.   | PENGUJIAN DAN ANALISIS .....                      | 79  |
| 5.1   | Pengujian Algoritma Dengan <i>White Box</i> ..... | 80  |
| 5.1.1 | Pengujian Proses Login.....                       | 80  |
| 5.1.2 | Pengujian Proses Edit Pasien .....                | 83  |
| 5.1.3 | Pengujian Proses Cetak Rekam Medis .....          | 86  |
| 5.1.4 | Pengujian Proses Cetak Laporan .....              | 88  |
| 5.2   | Validasi Sistem .....                             | 90  |
| 5.2.1 | Pengujian Login .....                             | 90  |
| 5.2.2 | Pengujian Input Data Pasien .....                 | 92  |
| 5.2.3 | Pengujian Antrian Melalui SMS .....               | 94  |
| 5.2.4 | Pengujian Cetak Rekam Medis .....                 | 98  |
| 5.2.5 | Pengujian Cetak Laporan .....                     | 100 |
| VII.  | PENUTUP .....                                     | 102 |
| 7.1   | Kesimpulan .....                                  | 102 |
| 7.2   | Saran .....                                       | 102 |
|       | DAFTAR PUSTAKA .....                              | 103 |
|       | LAMPIRAN.....                                     | 105 |



## ABSTRAK

**RACHMAT BUDIYANTO. 2007. : Penggunaan SMS Untuk Otomatisasi Pengambilan Nomor Antrian Pada Klinik Dokter Spesialis. Skripsi Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya. Dosen Pembimbing : Ir.Heru Nurwarsito, M.Kom. dan Herman Tolle, ST., MT.**

Beberapa upaya untuk membentuk sistem antrian telah banyak dilakukan, salah satunya adalah dengan merancang sistem berbasis mikrokontroler untuk mengatur panggilan pada konsumen berdasarkan urutan antriannya dan ditampilkan melalui peraga angka urutan dengan panggilan menggunakan bel khusus. Alangkah mudahnya jika antrian dapat dilakukan di rumah dengan mengambil nomor antrian terlebih dahulu kemudian datang sesuai dengan jam yang telah ditentukan. Aplikasi pengambilan nomor antrian dengan menggunakan SMS merupakan salah satu solusi untuk meningkatkan kenyamanan dan kemudahan serta untuk meminimalkan waktu tunggu untuk antrian pada dokter spesialis

Perancangan dan pengimplementasian sistem otomatisasi pengambilan nomor antrian pada klinik dokter spesialis menggunakan bahasa pemrograman Visual Basic 6.0 dan basis data MS Access 2000. Penggunaan handphone dengan mengirimkan SMS untuk mengambil nomor antrian dipilih karena kecepatan pengirimannya dan biaya yang lebih murah. Dari pengujian yang dilakukan, proses pengambilan nomor antrian dengan menggunakan SMS dapat mengirimkan balasan SMS secara otomatis berupa informasi nomor antrian dan jam datang.

Kata Kunci : SMS, Antrian, Otomatisasi, Visual Basic, MS Access.

## BAB I PENDAHULUAN

### 1.1. Latar Belakang

Salah satu bentuk pelayanan yang menjadi perhatian adalah layanan yang bisa menimbulkan antrian karena dimanfaatkan oleh banyak konsumen dalam waktu yang bersamaan, misalnya layanan pembayaran tagihan telpon, pembayaran rekening listrik, melakukan transaksi di bank, antrian pada rumah sakit ataupun dokter spesialis.

Pengaturan antrian dengan menggunakan cara manual dengan berdiri berbaris kemudian mengambil nomor antrian setelah itu menunggu untuk dipanggil telah ditinggalkan. Beberapa upaya untuk membentuk sistem antrian telah banyak dilakukan, salah satunya adalah dengan merancang sistem berbasis mikrokontroler untuk mengatur panggilan pada konsumen berdasarkan urutan antriannya dan ditampilkan melalui peraga angka urutan dengan panggilan menggunakan bel khusus.

Pada sistem antrian yang seperti ini waktu tunggu tidak dapat dipastikan dan bergantung pada banyaknya antrian. Dalam kondisi antrian yang seperti ini tentunya sangat tidak menyenangkan bagi yang melakukan antrian, utamanya antrian untuk konsultasi ke dokter spesialis. Alangkah mudahnya jika antrian dapat dilakukan di rumah dengan mengambil nomor antrian terlebih dahulu kemudian datang sesuai dengan jam yang telah ditentukan.

Teknologi telekomunikasi selular yang semakin maju memberikan nilai lebih dari sekedar untuk berkomunikasi. Handphone tidak hanya digunakan untuk berkomunikasi tetapi dapat berfungsi sebagai media untuk mengirim dan menerima pesan singkat atau yang lebih populer dengan SMS (*Short Message Service*). Kondisi tersebut diperkuat dengan munculnya berbagai VAS (*Value Added Services*) seperti m-Banking, m-Kuis dan masih banyak yang lain [IMR-04].

Dengan mempertimbangkan tingkat kenyamanan dan kemudahan serta untuk meminimalkan waktu tunggu untuk antrian pada dokter spesialis, kami merancang sebuah program dengan memanfaatkan SMS sebagai mediana untuk melakukan pengambilan nomor antrian. Penggunaan SMS sebagai media dipilih karena kecepatan pengirimannya dan biaya yang lebih murah.

## 1.2. Rumusan Masalah

Dalam perancangan ini rumusan masalah ditekankan pada:

1. Merancang dan membuat sistem yang dapat mengambil data SMS pada handphone dan menyimpannya dalam database.
2. Merancang dan membuat sistem yang otomatis dapat mengirimkan SMS pada pasien yang mengambil nomor antrian yang berisikan informasi nomor antrian dan waktu untuk datang.

## 1.3. Batasan Masalah

Mengacu pada permasalahan yang ada, maka pembahasan tugas akhir ini dibatasi pada:

1. Pengambilan nomor antrian dengan mengirimkan SMS.
2. Format data yang dikirim dari dan ke handphone berupa teks
3. Handphone yang digunakan adalah Nokia 3315 dan antarmuka dengan komputer menggunakan kabel data serial tipe 3315.
4. Pembuatan software menggunakan tool pengembang Visual Basic 6.0 dan MS Access sebagai Databasenya.

## 1.4. Tujuan

Tujuan dari tugas akhir ini adalah menghasilkan program aplikasi pengambilan nomor antrian dengan menggunakan SMS, sebagai salah satu solusi untuk meningkatkan kenyamanan dan kemudahan serta untuk meminimalkan waktu tunggu untuk antrian pada dokter spesialis.

## 1.5. Sistematika Penulisan

Sistematika penulisan dalam tugas akhir ini sebagai berikut:

### **BAB I**      Pendahuluan

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan dan sistematika penulisan.

### **BAB II**      Tinjauan Pustaka

Membahas teori-teori yang menunjang dalam perancangan dan pembuatan software yang meliputi teori dasar sistem kerja SMS, format data PDU, dasar teori sistem informasi, DFD, ER, ODBC dan bahasa pemrograman Visual Basic 6.0 serta MS Access sebagai databasenya.

**BAB III** Metodologi Penelitian

Membahas tentang metodologi penelitian yang digunakan.

**BAB IV** Perancangan

Membahas perancangan program yang sesuai dengan teori yang ada.

**BAB V** Implementasi

Mengimplementasikan hasil rancangan yang telah dibuat berupa algoritma.

**BAB VI** Pengujian dan Analisis

Membahas pengujian dan analisa program yang dibuat.

**BAB VII** Penutup

Berisi kesimpulan perancangan ini dan saran-saran yang perlu untuk melakukan pengembangan selanjutnya.



## BAB II

### TINJAUAN PUSTAKA

Untuk memudahkan dalam memahami dasar-dasar perencanaan sistem ini, maka perlu penjelasan dan uraian teori penunjang yang digunakan dalam penulisan skripsi ini.

#### 2.1 Sistem kerja SMS

Di balik tampilan menu *message* pada sebuah telepon selular sebenarnya adalah *AT Command* yang bertugas mengirim dan menerima data ke atau dari SMS Centre. *AT Command* tiap SMS *device* bisa berbeda-beda, tetapi pada dasarnya sama. Dalam perjalanan menuju ke SMS-Centre, sebuah SMS akan dikirim dalam bentuk kumpulan bilangan heksa yang berbentuk PDU (*Protocol Data Unit*).

*AT Command* sebenarnya hampir sama dengan perintah  $>$ (prompt) pada DOS. Perintah-perintah yang dimasukkan ke port dimulai dengan kata AT, lalu diikuti oleh karakter lainnya yang mempunyai fungsi unik. Beberapa *AT Command* penting untuk SMS, yaitu sebagai berikut [KHA-03]:

a. AT+CMGS= $n$

Digunakan untuk mengirim SMS di mana  $n$  adalah jumlah pasangan heksa PDU SMS dimulai setelah nomor SMS-Centre (maksimal seratus empat puluh).

b. AT+CMGR =  $n$

Digunakan untuk membaca SMS di mana  $n$  adalah indeks memori tempat SMS masuk disimpan.

c. AT+CMGL =  $n$

Digunakan untuk memeriksa SMS. Nilai  $n$  yang dapat digunakan antara lain:

- $n = 0$  untuk SMS baru di *inbox*.
- $n = 1$  untuk SMS lama di *inbox*.
- $n = 2$  untuk SMS *unsent* di *outbox*.
- $n = 3$  untuk SMS *sent* di *outbox*.
- $n = 4$  untuk semua SMS.

d. AT+CMGD =  $n$

Digunakan untuk menghapus SMS  $n$  adalah nomor referensi SMS yang akan dihapus.

Berikut ini ditunjukkan beberapa *AT Command* set yang berhubungan dengan pengoperasian SMS seperti ditunjukkan Tabel 2.1, karena *AT command* tidak hanya digunakan pada operasi SMS.

**Tabel 2.1** Contoh Perintah *AT Command*

| Perintah | Fungsi  |
|----------|---|
| AT       | Awalan untuk semua perintah lainnya                 |
| ATE0     | Mendeaktivasi perintah echo                         |
| ATE1     | Mengaktifasi perintah echo                          |
| AT+CMGC  | Mengirim sebuah perintah SMS                        |
| AT+CMGD  | Menghapus sebuah SMS dalam SMS memori               |
| AT+CMGF  | SMS Format  |
| AT+CMGL  | Daftar SMS  |
| AT+CMGR  | Membaca dalam sebuah SMS                            |
| AT+CMGS  | Mengirim sebuah SMS                                 |
| AT+CMGW  | Mengisi sebuah SMS ke SMS memori                    |
| AT+CMSS  | Mengirim sebuah SMS dari SMS memori                 |
| AT+CNMA  | Pemberitahuan pesan singkat langsung pengeluarannya |
| AT+CNMI  | Menampilkan datangnya SMS baru                      |
| AT+CSCA  | Alamat dari pusat SMS servis                        |

Sumber: [KHA-03]

### 2.1.1 PDU (*Protocol Data Unit*)

Data yang mengalir ke atau dari SMS-Centre harus berbentuk PDU (*Protocol Data Unit*). PDU berisi bilangan-bilangan heksadesimal yang mencerminkan bahasa I/O. PDU terdiri atas beberapa bagian entitas. Untuk keperluan SMS, terdapat dua jenis PDU yaitu PDU untuk arah kirim ke SMS-Centre dan PDU untuk arah terima dari SMS-Centre. Bagian untuk kirim SMS ke SMS-Centre berbeda dengan SMS yang diterima dari SMS-Centre. PDU adalah gabungan dari isi SMS yang dikirimkan dengan data-data lainnya yang telah diolah standart PDU.

- **PDU Untuk Arah Kirim Dan Terima Ke/Dari SMS-Centre.**

Untuk PDU arah kirim terdapat 8 entiti yang harus ada, yaitu:

1. No SMS-Centre
2. Tipe SMS
3. No referensi SMS
4. No ponsel penerima
5. Bentuk SMS
6. Skema *encoding* data I/O
7. Jangka waktu sebelum SMS *expired*
8. Isi SMS

Untuk PDU arah terima dari SMS-Centre terdiri dari:

1. No SMS-Centre
2. Tipe SMS
3. No ponsel pengirim
4. Bentuk SMS
5. Skema *encoding*
6. Tanggal dan waktu SMS di stamp oleh SMS-Centre
7. Batas waktu validitas
8. Isi SMS

**A. PDU untuk kirim SMS ke SMS-Centre**

PDU untuk mengirim SMS terdiri atas delapan entitas

**1. Nomor SMS-Centre**

Entitas pertama terdiri atas tiga subentitas, yaitu:

- a. Jumlah pasangan heksadesimal SMS-Centre dalam bilangan heksa
- b. Kode nasional atau internasional
- c. No SMS-Centrenya sendiri, dalam pasangan heksa dibalik-balik. Jika tertinggal satu angka heksa yang tidak memiliki pasangan, angka tersebut akan dipasangkan dengan huruf F didepannya.

Sebagai contoh untuk SMS-Centre Exelcom dapat ditulis dengan dua cara sebagai berikut:

**Cara 1 (Nasional)**

0818445009 diubah menjadi

a.06 : ada 6 pasang

b.81 : 1 pasang

c.80-81-44-05-90 : 5 pasang

digabung menjadi : 06818081440590

**Cara 2 (Internasional)**

0818445009 diubah menjadi

a.07 : ada 7 pasang

b.91 : 1 pasang

c.26-18-48-54-00-f9 : 6 pasang

digabung menjadi : 07912618485400f9

Berikut ini beberapa nomor SMS-Centre operator selular di indonesia

**Tabel 2.2** PDU Format Nomor SMS-Centre Nasional

| No | Operator selular | No SMS-Centre | Kode PDU       |
|----|------------------|---------------|----------------|
| 1  | Telkomsel        | 0811000000    | 06818011000000 |
| 2  | Satelindo        | 0816125       | 0581806121f5   |
| 3  | Excelcom         | 0818445009    | 06818081440590 |
| 4  | Indosat-m3       | 0855000000    | 06818055000000 |

**Tabel 2.3** PDU Format Nomor SMS-Centre Internasional

| No | Operator selular | No SMS-Centre | Kode PDU         |
|----|------------------|---------------|------------------|
| 1  | Telkomsel        | 0811000000    | 07912618010000f0 |
| 2  | Satelindo        | 0816125       | 059126181652     |
| 3  | Excelcom         | 0818445009    | 07912618485400f9 |
| 4  | Indosat-m3       | 0855000000    | 07912658050000f0 |

**2. Tipe SMS**

Untuk send tipe SMS=1 jadi bilangan heksanya adalah 01

**3. Nomor referensi SMS**

Nomor referensi ini dibiarkan dulu 0, jadi bilangan heksanya adalah 00. Nanti akan diberikan sebuah nomor referensi otomatis oleh ponsel/alat SMS-gateway.

#### 4. Nomor ponsel penerima

Sama seperti menulis PDU entitas untuk SMS-Centre, entitas ini juga terdiri atas tiga bagian, sebagai berikut:

- a. Jumlah bilangan desimal nomor ponsel yang dituju dalam bilangan heksa
  - b. Kode nasional/internasional
    - Untuk nasional, kode subentitasnya 81
    - Untuk internasional, kode subentitasnya 91
  - c. Nomor ponsel yang dituju, dalam pasangan heksa dibalik-balik. jika tertinggal satu angka heksa yang tidak memiliki pasangan, angka tersebut dipasangkan dengan huruf F didepannya
- contoh :

Untuk nomor ponsel yang dituju = 628129573337 dapat ditulis:

**Cara 1 (Nasional)** 08129573337 diubah menjadi

a.0B : ada 11 angka

b.81

c.80-21-59-37-33-f7

digabung menjadi 0B818021593733F7

**Cara 2 (Internasional)** 08129573337 diubah menjadi

a.0C : ada 11 angka

b.91

c.26-18-92-75-33-73

digabung menjadi 0C91261892753373

#### 5. Bentuk SMS, antara lain:

0 00 dikirim sebagai SMS

1 01 dikirim sebagai telex

2 02 dikirim sebagai fax

untuk mengirim dalam bentuk SMS tentu saja memakai 00

#### 6. Skema encoding data I/O

Ada dua skema, yaitu :

a. Skema 7 bit: ditandai dengan angka 0

b. Skema 8 bit: ditandai dengan angka lebih besar dari 0

Kebanyakan ponsel atau SMS gateway yang ada dipasaran sekarang menggunakan skema 7 bit sehingga menggunakan kode 00

**Tabel 2.4** Skema Encoding Data 7 bit

| b4 | b3 | b2 | b1 |    |    |   |    |   |   |   |    |   |
|----|----|----|----|----|----|---|----|---|---|---|----|---|
| 0  | 0  | 0  | 0  | 0  | @  | Δ | SP | 0 | - | P | .. | p |
| 0  | 0  | 0  | 1  | 1  |    |   | !  | 1 | A | Q | a  | q |
| 0  | 0  | 1  | 0  | 2  | \$ | Φ | "  | 2 | B | R | b  | r |
| 0  | 0  | 1  | 1  | 3  |    | Γ | #  | 3 | C | S | c  | s |
| 0  | 1  | 0  | 0  | 4  |    | Λ |    | 4 | D | T | d  | t |
| 0  | 1  | 0  | 1  | 5  |    | Ω | %  | 5 | E | U | e  | u |
| 0  | 1  | 1  | 0  | 6  |    | Π | &  | 6 | F | V | f  | v |
| 0  | 1  | 1  | 1  | 7  |    | Ψ | '  | 7 | G | W | g  | w |
| 1  | 0  | 0  | 0  | 8  |    | Σ | (  | 8 | H | X | h  | x |
| 1  | 0  | 0  | 1  | 9  |    | Θ | )  | 9 | I | Y | i  | y |
| 1  | 0  | 1  | 0  | 10 | LF | Ε | *  | : | J | Z | j  | z |
| 1  | 0  | 1  | 1  | 11 |    |   | +  | ; | K | Ä | k  | ä |
| 1  | 1  | 0  | 0  | 12 |    |   | ,  | < | L | Ö | l  | ö |
| 1  | 1  | 0  | 1  | 13 | CR |   | -  | = | M |   | m  |   |
| 1  | 1  | 1  | 0  | 14 |    | β | .  | > | N | Ü | n  | ü |
| 1  | 1  | 1  | 1  | 15 |    |   | /  | ? | O |   | o  |   |

Sumber: [KHA-03]

## 7. Jangka waktu sebelum SMS expired

Jika bagian ini dilewatkan berarti tidak membatasi waktu berlakunya SMS. Sedangkan jika kita isi dengan suatu bilangan integer yang kemudian diubah ke pasangan heksa tertentu, bilangan tersebut akan mewakili jumlah waktu validitas SMS tersebut. Untuk menghitung jangka waktu validitas SMS adalah sebagai berikut :

**Tabel 2.5** Jangka Waktu Validitas SMS

| Integer ( INT) | Jangka waktu validitas SMS                        |
|----------------|---|
| 0-143          | (int+1) x 5 menit (berarti : 5 menit s/d 12 jam ) |
| 144-167        | 12 jam + (int-143) x 30 menit                     |
| 168-196        | (int- 166) x 1 hari                               |
| 197-255        | (int- 192) x 1 minggu                             |

Agar SMS pasti terkirim sampai ke ponsel penerima, sebaiknya tidak memberikan batasan waktu validnya.

### 8. Isi SMS

Entitas ini terdiri atas dua sub entitas, yaitu:

- Panjang isi (jumlah huruf dari isi)
- Isi berupa bilangan heksa, untuk ponsel/SMS gateway berskema *encoding* 7 bit.

### B. PDU untuk SMS terima dari SMS Centre

- No SMS-Centre
- Tipe SMS, untuk SMS terima = 4
- No ponsel pengirim
- Bentuk SMS
- Skema *encoding*
- Tanggal dan waktu SMS distamp di SMS-Centre
- Batas waktu validitas
- Isi SMS

contoh untuk PDU :

**07912658050000F0 | 04 | 0C91265816107398 | 00 | 00 | 207022512380 | 00 |**  
                   1                  2                  3                  4 5                  6                  7  
**5E8329BFD06**  
                   8

Dapat diartikan sebagai berikut:

- SMS tersebut dikirim lewat SMS-Centre: 62855000000
- SMS tersebut merupakan SMS terima
- SMS tersebut dikirim dari ponsel no :6285661013789
- SMS tersebut diterima dalam bentuk SMS
- SMS tersebut memiliki skema encoding 7 bit
- SMS tersebut sampai di SMS-Centre pada tanggal 22-07-02. pukul 15:32:08 WIB
- SMS tersebut tidak memiliki batas waktu valid
- SMS tersebut isinya “hello”

## 2.2 Sistem Informasi

Sistem informasi dapat didefinisikan sebagai kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses, dan menyimpan serta mendistribusikan informasi. Sistem informasi merupakan kesatuan elemen-elemen yang saling berinteraksi secara sistematis dan teratur untuk menciptakan dan membentuk aliran informasi.

Pembuatan sistem informasi memerlukan perangkat pemodelan sistem informasi. Perangkat pemodelan berfungsi sebagai media yang memberikan penjelasan tentang sistem informasi yang dibuat. Perangkat pemodelan dapat berupa diagram, *flowchart*, maupun gambar.

Tiga alasan yang menyebabkan kita sebaiknya melakukan pemodelan sistem, yaitu [POH-97] :

- Dapat memfokuskan perhatian pada hal-hal penting dalam sistem tanpa mesti terlibat terlalu jauh.
- Mendiskusikan perubahan dan koreksi terhadap kebutuhan pemakai dengan resiko dan biaya minimal.
- Menguji pengertian penganalisa sistem terhadap kebutuhan pemakai dan membantu pendesain sistem dan pemrogram membangun sistem.

### 2.2.1 Data Flow Diagram Context Level (Diagram Konteks)

Diagram konteks adalah sebuah diagram sederhana yang menggambarkan hubungan dengan entitas luar, masukan dan keluaran dari sistem. Diagram konteks direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem [POH-97]

Diagram konteks mempunyai karakteristik penting dari sebuah sistem, yaitu:

1. Kelompok pemakai, organisasi atau sistem lain melakukan komunikasi yang disebut juga sebagai terminator.
2. Data masuk yaitu data yang diterima sistem dari lingkungan dan harus diproses dengan cara-cara tertentu.
3. Data keluar yaitu data yang dihasilkan sistem dan diberikan ke pihak luar.
4. Penyimpanan data yang digunakan secara bersama-sama antara sistem dan terminator. Data ini dapat dibuat oleh sistem dan digunakan oleh lingkungan atau sebaliknya.

5. Batasan antara sistem dan lingkungan.

### 2.2.2 *Data Flow Diagram Levelled*

*Data flow diagram levelled* menggambarkan sistem sebagai jaringan kerja antar fungsi yang berhubungan satu sama lain dengan proses penyimpanan serta aliran data. Empat komponen dalam model *Data flow diagram levelled* antara lain [POH-97]:

1. Proses, fungsi dan tranformasi. Proses ini menunjukkan tranformasi dari masukan menjadi keluaran.
2. Aliran data. Aliran data digunakan untuk menggambarkan gerakan paket data informasi dari satu bagian ke bagian lain..
3. Penyimpanan. komponen ini digunakan untuk memodelkan kumpulan data atau paket data.
4. Terminator. komponen ini mewakili entitas luar sistem berkomunikasi.

### 2.2.3 *Entity-Relationship Diagram (Diagram E-R)*

Diagram *Entity-Relationship* menjelaskan semesta data yang ada pada dunia nyata. Diagram *Entity-Relationship* menterjemahkan atau mentransformasikan data dengan memanfaatkan sejumlah perangkat konseptual [POH-97]. Komponen-komponen pembentuk diagram *Entity-Relationship*, antara lain:

1. Entitas dan Himpunan Entitas

Entitas merupakan individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Sekelompok entitas yang sejenis dan berada dalam lingkup yang sama membentuk sebuah himpunan entitas. Entitas menunjuk pada individu suatu objek, sedangkan himpunan entitas menunjuk pada rumpun dari individu tersebut. Himpunan entitas digambarkan dengan persegi panjang dalam diagram *Entity-Relationship*.

2. Atribut

Entitas pasti memiliki atribut yang mendeskripsikan karakteristik dari entitas tersebut. Penentuan atau pemilihan atribut-atribut yang relevan bagi sebuah entitas merupakan hal penting lain dalam pembentukan model data. Atribut digambarkan dengan elips dalam diagram *Entity-Relationship*.

### 3. Relasi dan Himpunan Relasi

Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Kumpulan semua relasi di antara entitas-entitas yang terdapat pada himpunan entitas-himpunan entitas tersebut membentuk suatu himpunan relasi. Himpunan relasi digambarkan dengan belah ketupat dalam diagram *Entity-Relationship*.

### 4. Kardinalitas atau Derajat Relasi

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari himpunan entitas yang satu ke himpunan entitas yang lain. Kardinalitas relasi yang terjadi di antara dua himpunan entitas (misal A dan B) dapat berupa:

- Satu ke Satu (*One to One*)

Entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B dan sebaliknya.

- Satu ke Banyak (*One to Many*)

Entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B tetapi tidak sebaliknya. Setiap entitas pada himpunan entitas B berhubungan paling banyak dengan satu entitas pada himpunan entitas A.

- Banyak ke Satu (*Many to One*)

Entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B tetapi tidak sebaliknya. Setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.

- Banyak ke Banyak (*Many to Many*)

Entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B dan sebaliknya.

Diagram E-R selalu dibuat bertahap. Tahapan pertama yang dapat dilakukan adalah dengan membuat diagram E-R awal. Tujuan dari pentahapan ini adalah untuk mendapatkan sebuah rancangan basis data minimal yang dapat mengakomodasi

kebutuhan penyimpanan data terhadap sistem yang sedang ditinjau. Langkah-langkah yang harus dilakukan antara lain:

1. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat.
2. Menentukan atribut-atribut key dari masing-masing himpunan entitas.
3. Mengidentifikasi dan menetapkan seluruh himpunan relasi di antara himpunan entitas-himpunan entitas yang ada beserta *foreign key*-nya.
4. Menentukan derajat atau kardinalitas relasi untuk setiap himpunan relasi.
5. Melengkapi himpunan entitas dan himpunan relasi dengan atribut-atribut deskriptif (*non-key*).

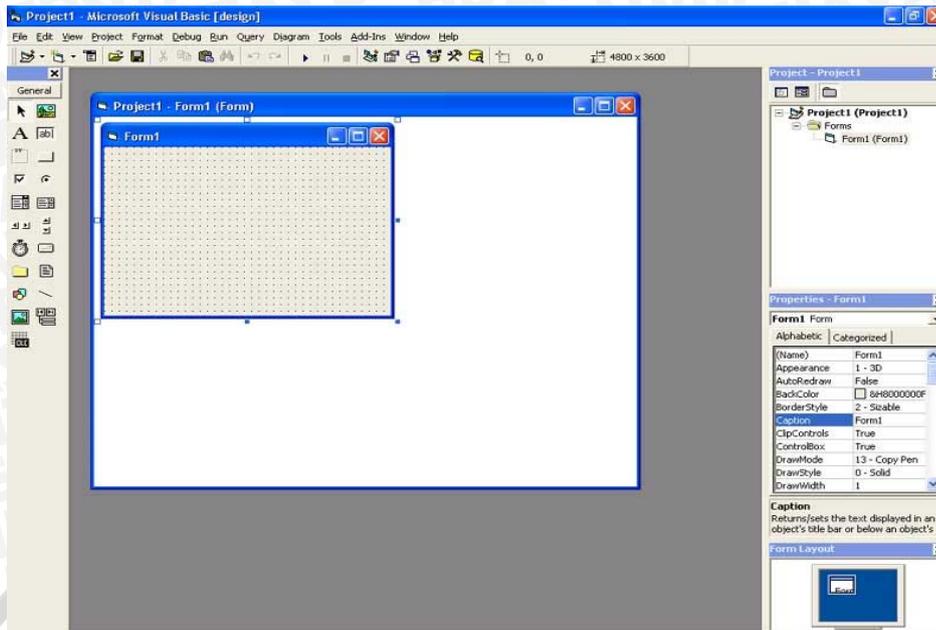
### 2.3 Visual Basic 6.0

Visual Basic adalah salah satu bahasa pemrograman komputer. Bahasa pemrograman adalah perintah-perintah yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Bahasa pemrograman Visual Basic, yang dikembangkan oleh Microsoft sejak tahun 1991, merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC (*Beginner's All-purpose Symbolic Instruction Code*) yang dikembangkan pada era 1950-an.

Visual Basic merupakan salah satu *Development Tool* yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi Windows. Visual Basic merupakan salah satu bahasa pemrograman komputer yang mendukung object (*Object Oriented Programming* = OOP) [SUR-03].

#### 2.3.1 Bidang Kerja Visual Basic

Setelah kita membuka/menjalankan Visual Basic 6.0. kita akan masuk ke dalam jendela *Integrated Development Environment* (IDE). IDE merupakan bidang kerja tempat kita membangun dan menghasilkan program aplikasi, seperti terlihat pada Gambar 2.1.

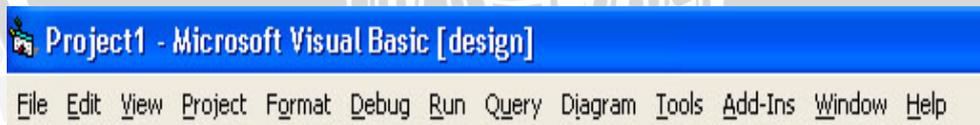


**Gambar 2.1** Tampilan *Integrated Development Environment (IDE)*

Jendela kerja tersebut terbagi atas beberapa jendela spesifik yang nantinya akan sangat berperan dalam proses pembuatan *project*.

### 2.3.1.1 Menu Bar

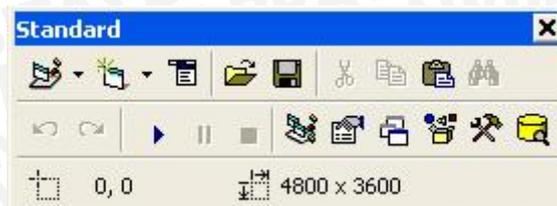
*Menu Bar* (Baris Menu) merupakan salah satu elemen dari *Integrated Development Environment (IDE)*. *Menu Bar* terdiri dari perintah-perintah yang dikelompokkan dalam kriteria operasi yang dihasilkan. *Menu Bar* dapat digambarkan seperti pada Gambar 2.2.



**Gambar 2.2** Menu Bar

### 2.3.1.2 Tool Bar

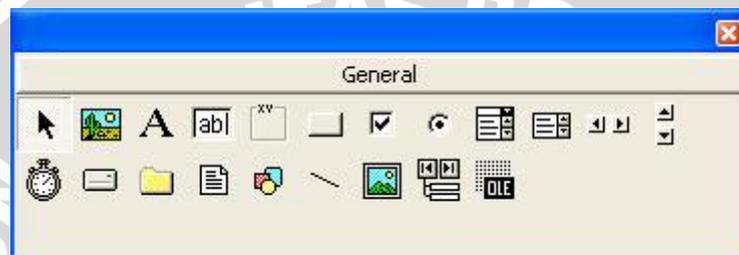
*Tool Bar* berisi *icon-icon* yang berfungsi mempersingkat pemilihan perintah pada menu bar. Secara *default*, *tool bar* yang akan muncul adalah seperti Gambar 2.3. dibawah.



Gambar 2.3 Tool Bar

### 2.3.1.3 Tool Box

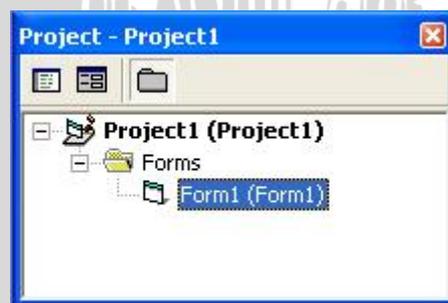
*Tool Box* merupakan kumpulan kontrol yang banyak digunakan dalam pemrograman Visual Basic 6.0. Secara *default* toolbox berisi satu tab (*general*) yang terdiri atas beberapa kontrol.



Gambar 2.4 Tool Box

### 2.3.1.4 Window Project Explorer

*Window Project Explorer* merupakan windows yang berisi bagian-bagian pembentuk *project*. Pada window ini ditampilkan form, dan object-object yang ada dalam suatu *project*.



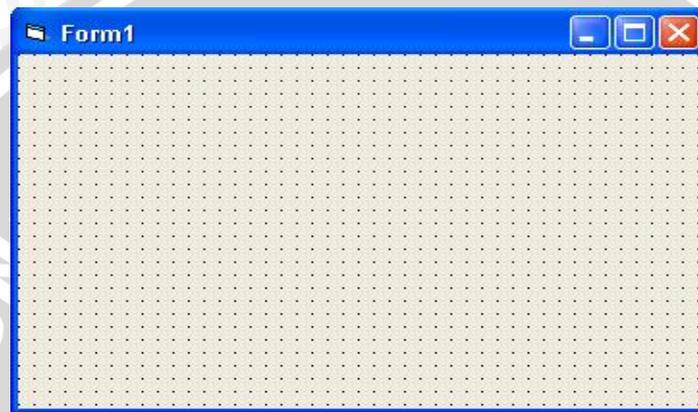
Gambar 2.5 Window Project Explorer

### 2.3.1.5 Window Form

*Form* atau yang biasa juga disebut *window* (jendela) merupakan area kerja anda akan merancang suatu program aplikasi dengan Visual Basic. Pada form berbagai macam kontrol (obyek) dapat diletakkan seperti *command button*, *textbox*, *label*, *checkbox*, dan lainnya. Ukuran suatu *form* dapat diubah dengan melakukan *Drag*

dan *Drop* pada titik-titik yang ada disisi form, disebelah atas, bawah, kiri, dan kanan.

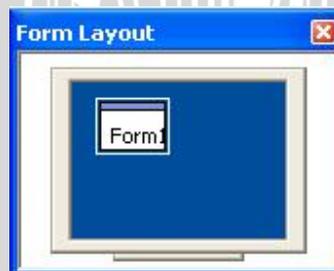
Saat program dijalankan, *form* akan menjadi latar belakang dari obyek-obyek yang menaunginya. Terdapat tiga tombol pada sebuah form standar, yaitu: *Minimize*, *Restore* dan *Close*. Pada form akan terdapat titik-titik yang tertata rapi secara vertikal dan horisontal yang disebut *Grid*. Fungsi dari *grid* adalah untuk membantu anda dalam menempatkan obyek ke dalam *form* agar posisinya serasi. Pada Gambar 2.6 ditampilkan bentuk suatu *form* yang belum diisi oleh komponen.



Gambar 2.6 Window Form

#### 2.3.1.6 Window Form Layout

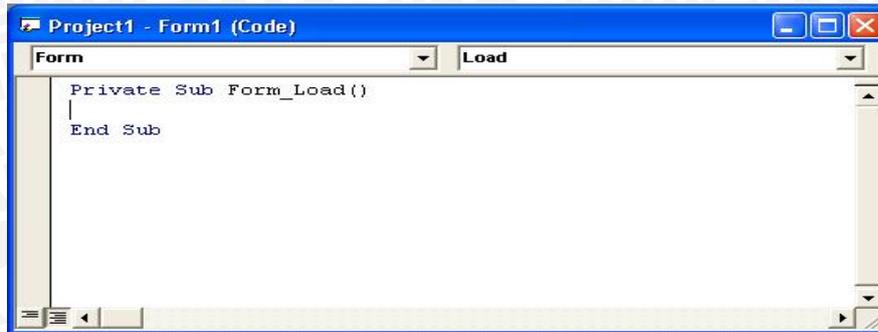
*Window Form Layout* digunakan untuk menentukan posisi suatu *form* yang akan diaktifkan saat program dijalankan. Untuk mengubah posisi pengaktifan tersebut, *drag form* ke arah posisi yang diinginkan



Gambar 2.7 Window Form Layout

#### 2.3.1.7 Window Code Editor

*Window Code Editor* berfungsi untuk menuliskan semua kode program pada *project* yang dibuat.



Gambar 2.8 Window Code Editor

### 2.3.1.8 Window Properties

*Window Properties* digunakan untuk mengatur segala properti dari obyek yang diperlukan, baik dalam perancangan *interface* maupun pemrograman.



Gambar 2.9 Window Properties

### 2.3.1.9 Window Immediate

*Window Immediate* berfungsi untuk menguji instruksi program. Saat pengujian program tersebut, window ini dapat dijadikan sebagai *windows debug* (pencari kesalahan).

## 2.3.2 Dasar Pemrograman Visual Basic

Visual Basic merupakan salah satu bahasa pemrograman komputer yang mendukung object (*Object Oriented Programming* = OOP). Dalam OOP ini dikenal beberapa istilah, yaitu:

- Object** : komponen di dalam sebuah program
- Property** : karakteristik yang dimiliki object
- Method** : aksi yang dapat dilakukan oleh object
- Event** : kejadian yang dapat dialami oleh object

Implementasinya dalam sebuah aplikasi misalnya anda membuat form, maka form tersebut memiliki *property*, *method*, dan *event*. Sebagaimana pemrograman visual lain seperti Delphi dan Java, VB juga bersifat *event driven programming*. Artinya anda dapat menyisipkan kode program pada *event* yang dimiliki suatu obyek.

### 2.3.2.1 Penulisan Kode Program

Untuk menuliskan kode program kita harus masuk pada *Window Code Editor* kemudian menuliskan kode program. *Window Code Editor* dapat ditunjukkan pada Gambar 2.10 dibawah ini.



**Gambar 2.10** Properties *Window Code Editor*  
Sumber : [MAN-04]

Program yang berbasis Windows bersifat *event-driven*, artinya program bekerja berdasarkan event yang terjadi pada object di dalam program tersebut. Misalnya, jika seorang user meng-klik sebuah tombol maka program akan memberikan “reaksi” terhadap event klik tersebut. Program akan memberikan “reaksi” sesuai dengan kode-kode program yang dibuat untuk suatu event pada object tertentu. Pilih object Command1 (tombol) pada bagian Object Selector. Otomatis pada bagian *Code Editor* akan muncul blok kode program berikut:

```
Private Sub Command1_Click()  
End Sub
```

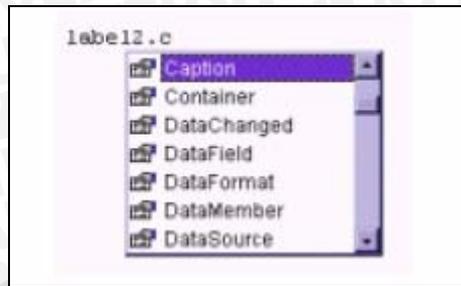
Event Click merupakan event *default* (standar) dari object Command1, Anda bisa melihat event lainnya pada bagian Event Selector.

Cara penulisan (*syntax*) kode program untuk mengatur property sebuah object adalah sebagai berikut :

```
nama_object.property = value
```

Perhatikan, setelah Anda mengetikkan tanda titik (dot) dibelakang nama\_object maka VB 6 akan menampilkan daftar property dan method yang dimiliki oleh

nama\_object , seperti ditunjukkan pada Gambar 2.11 dibawah ini. Fasilitas ini disebut *Intellisense*.



**Gambar 2.11** Fasilitas *Intellisense*  
Sumber : [MAN-04]

### 2.3.2.2 Procedure

*Procedure* adalah blok kode program yang berisi perintah-perintah untuk mengerjakan tugas tertentu. Bila di dalam kode program yang kita buat ada perintah-perintah untuk melakukan tugas yang sama di beberapa tempat, maka akan lebih baik perintah-perintah tersebut dibuat dalam sebuah *procedure*. Kemudian, *procedure* itu bisa di panggil bila diperlukan.

Penggunaan *procedure* sangat menghemat penulisan kode program, karena kode-kode program yang sama di beberapa tempat cukup dibuat pada satu bagian saja. Selain itu, *procedure* akan memudahkan perbaikan kode program bila terjadi perubahan atau kesalahan, karena perbaikan cukup dilakukan pada satu bagian saja.

- **Sub Procedure**

*Sub Procedure* merupakan kumpulan kode yang dilaksanakan sebagai tanggapan atas suatu kejadian (*event*), sehingga program lebih mudah dibaca dan dianalisa. Sub prosedur merupakan prosedur yang tidak mengembalikan nilai setelah ‘tugas’-nya selesai.

Rumus untuk perintah *sub procedure* secara umum adalah sebagai berikut :

```
[Private/Public] Sub <nama_procedure> (argumen)
    pernyataan 1
    pernyataan 2
    ...
End Sub
```

Bila *procedure* dipanggil maka pernyataan di antara Sub dan End Sub akan dilaksanakan. Untuk memanggil suatu *procedure* digunakan *Call* atau dengan langsung menuliskan nama *procedure* tersebut.

```
Call nama_procedure (argumen)
```

Atau :

```
nama_procedure argumen
```

- **Function Procedure**

*Function Procedure* (fungsi) merupakan *procedure* yang mampu menghasilkan nilai atau prosedur yang mengembalikan nilai setelah ‘tugas’-nya selesai.

Penulisan kode program, yaitu :

```
[Private/Public] Function <nama_function>(argumen) As <tipe>
    pernyataan 1
    pernyataan 2
    pernyataan 3
    pernyataan 4
    ...
End Function
```

Sedangkan untuk memanggil fungsi, menggunakan *Call* atau langsung tulis nama fungsinya, seperti pemanggilan *procedure* diatas.

### 2.3.2.3 Data dan Variabel

Ketika seorang *user* (pengguna) menggunakan sebuah program komputer, seringkali computer memintanya untuk memberikan informasi. Informasi ini kemudian disimpan atau diolah oleh komputer. Informasi inilah yang disebut dengan **DATA** .

Visual Basic 6 mengenal beberapa type data, antara lain :

- **String** adalah type data untuk teks (huruf, angka dan tanda baca).
- **Integer** adalah type data untuk angka bulat.
- **Single** adalah type data untuk angka pecahan.
- **Currency** adalah type data untuk angka mata uang.
- **Date** adalah type data untuk tanggal dan jam.
- **Boolean** adalah type data yang bernilai TRUE atau FALSE.

Data yang disimpan di dalam memory komputer membutuhkan sebuah wadah. Wadah inilah yang disebut dengan **VARIABEL**. Setiap variabel untuk menyimpan data dengan type tertentu membutuhkan alokasi jumlah memory (*byte*) yang berbeda.

Variabel dibuat melalui penulisan deklarasi variabel di dalam kode program :

**Dim <nama\_variabel> As <type\_data>**

Contoh : Dim nama\_program As String

Aturan di dalam penamaan variabel :

- Harus diawali dengan huruf.
- Tidak boleh menggunakan spasi. Spasi bisa diganti dengan karakter underscore ( \_ ).
- Tidak boleh menggunakan karakter-karakter khusus (seperti : +, -, \*, /, <, dll).
- Tidak boleh menggunakan kata-kata kunci yang sudah dikenal oleh Visual Basic 6 (seperti : dim, as, string, integer, dll).

Sebuah variabel hanya dapat menyimpan satu nilai data sesuai dengan type datanya. Cara mengisi nilai data ke dalam sebuah variabel :

**<nama\_variabel> = <nilai\_data>**

Contoh : nama\_program = "konversi"

Untuk type data tertentu nilai\_data harus diapit tanda pembatas. Type data string dibatasi tanda petik ganda: "nilai\_data". Type data date dibatasi tanda pagar : #nilai\_data#. Type data lainnya tidak perlu tanda pembatas.

Sebuah variabel mempunyai ruang-lingkup (*scope*) dan waktu-hidup (*lifetime*):

- **Variabel global** adalah variabel yang dapat dikenali oleh seluruh bagian program. Nilai data yang tersimpan didalamnya akan hidup terus selama program berjalan.
- **Variabel lokal** adalah variabel yang hanya dikenali oleh satu bagian program saja. Nilai data yang tersimpan didalamnya hanya hidup selama bagian program tersebut dijalankan.

Variabel yang nilai datanya bersifat tetap dan tidak bisa diubah disebut **KONSTANTA**. Penulisan deklarasi konstanta di dalam kode program :

**Const <nama\_kontanta> As <type\_data> = <nilai\_data>**

Contoh: Const tgl\_gajian As Date = #25/09/2003#

### 2.3.2.4 Operator

Operator di dalam bahasa pemrograman bisa diartikan sebagai simbol yang digunakan untuk melakukan suatu operasi terhadap nilai data. Simbol operator bisa berupa karakter ataupun kata khusus .

Visual Basic 6 mengenal tiga jenis operator, yaitu:

1. **Operator Aritmatika** digunakan untuk operasi matematis terhadap nilai data. Simbol-simbol yang digunakan :

**Tabel 2.6** Operator Aritmatika

| Simbol | Operasi Matematis         | Contoh             |
|--------|---------------------------|--------------------|
| ^      | Pemangkatan               | 5 ^ 2 hasilnya 25  |
| *      | Perkalian                 | 5 * 2 hasilnya 10  |
| /      | Pembagian (hasil pecahan) | 5 / 2 hasilnya 2.5 |
| \      | Pembagian (hasil bulat)   | 5 \ 2 hasilnya 2   |
| Mod    | Sisa pembagian            | 5 Mod 2 hasilnya 1 |
| +      | Penjumlahan               | 5 + 2 hasilnya 7   |
| -      | Pengurangan               | 5 - 2 hasilnya 3   |
| &      | Penggabungan string       | 5 & 2 hasilnya 52  |

Sumber : [MAN-04]

2. **Operator Perbandingan** digunakan untuk operasi yang membandingkan nilai data. Simbol-simbol yang digunakan:

**Tabel 2.7** Operator Perbandingan

| Simbol | Operasi Perbandingan         | Contoh                |
|--------|------------------------------|-----------------------|
| <      | Lebih besar                  | 5 < 2 hasilnya FALSE  |
| >      | Lebih kecil                  | 5 > 2 hasilnya TRUE   |
| <=     | Lebih kecil atau sama dengan | 5 <= 2 hasilnya FALSE |
| >=     | Lebih besar atau sama dengan | 5 >= 2 hasilnya TRUE  |
| =      | Sama dengan                  | 5 = 2 hasilnya FALSE  |
| <>     | Tidak sama dengan            | 5 <> hasilnya TRUE    |

Sumber : [MAN-04]

3. **Operator Logika** digunakan untuk operasi yang membandingkan suatu perbandingan. Simbolsimbol yang digunakan:

Tabel 2.8 Operator Logika

| Simbol | Operasi Logika | Contoh                         |
|--------|----------------|--------------------------------|
| Or     | Atau           | (5<2) Or (5>2) hasilnya TRUE   |
| And    | Dan            | (5<2) And (5>2) hasilnya FALSE |
| Not    |                | Not (5<2) hasilnya TRUE        |

Sumber : [MAN-04]

### 2.3.2.5 Struktur Kontrol

Struktur kontrol di dalam bahasa pemrograman adalah perintah dengan bentuk (struktur) tertentu yang digunakan untuk mengatur (mengontrol) jalannya program .

Visual Basic 6 mengenal dua jenis struktur kontrol, yaitu:

1. **Struktur kontrol keputusan** - digunakan untuk memutuskan kode program mana yang akan dikerjakan berdasarkan suatu kondisi.

Ada dua bentuk struktur kontrol keputusan, yaitu :

- Struktur **IF ... THEN**

Bentuk penulisan (*syntax*) struktur IF...THEN :

- **IF <kondisi> THEN <kode program>**

Bila <kondisi> bernilai True maka <kode program> akan dikerjakan.

- **IF <kondisi> THEN**

**<blok kode program 1>**

**ELSE**

**<blok kode program 2>**

**END IF**

Bila <kondisi> bernilai True maka <blok kode program 1> akan dikerjakan, tetapi bila <kondisi> bernilai False maka <blok kode program 2> yang akan dikerjakan.

- Struktur **SELECT ... CASE**

Bentuk penulisan (*syntax*) struktur SELECT...CASE :

**SELECT CASE <pilihan>**

**CASE <pilihan 1>**

**<blok kode program 1>**

**CASE <pilihan 2>**

**<blok kode program 2>**

**CASE <pilihan n>**

```

    <blok kode program n>
[CASE ELSE
    <blok kode program x>]
END SELECT

```

Bila <pilihan> sesuai dengan <pilihan 1> maka <blok kode program 1> akan dikerjakan, dst. Tetapi bila <pilihan> tidak ada yang sesuai dengan <pilihan1> s/d <pilihan n> maka <blok kode program x> yang akan dikerjakan.

2. **Struktur kontrol pengulangan** - digunakan untuk melakukan pengulangan kode program.

Ada dua bentuk struktur kontrol pengulangan (*looping*), yaitu:

- Struktur **FOR ... NEXT**

Bentuk penulisan (*syntax*) struktur For...Next:

```

FOR <pencacah> = <awal> TO <akhir> [STEP <langkah>]
    <blok kode program>
NEXT <pencacah>

```

<pencacah> adalah variabel (tipe: integer) yang digunakan untuk menyimpan angka pengulangan. <awal> adalah nilai awal dari <pencacah>. <akhir> adalah nilai akhir dari <pencacah>. <langkah> adalah perubahan nilai <pencacah> setiap pengulangan. Sifatnya optional (boleh ditulis ataupun tidak). Bila tidak ditulis maka nilai <langkah> adalah 1.

- Struktur **DO ... LOOP**

Bentuk penulisan (*syntax*) struktur Do...Loop:

- **DO WHILE <kondisi>**  
     <blok kode program>

**LOOP**

<blok kode program> akan diulang **selama** <kondisi> bernilai TRUE. Pengulangan berhenti bila <kondisi> sudah bernilai FALSE.

- **DO UNTIL <kondisi>**  
     <blok kode program>

**LOOP**

<blok kode program> akan diulang **sampai** <kondisi> bernilai TRUE. Pengulangan berhenti bila <kondisi> sudah bernilai TRUE.

## 2.4 ODBC

*Open Database Connectivity* (ODBC) merupakan antarmuka dari Microsoft yang telah menjadi standar mekanisme aplikasi *client* untuk mengakses data dari berbagai sumber data yang berbeda dengan hanya melalui satu antarmuka. Antarmuka Microsoft ODBC adalah suatu standard industri saat ini dan merupakan komponen dari Microsoft *Windows Open Services Architecture* (WOSA). Antarmuka ODBC membuat aplikasi-aplikasi dapat mengakses data dari berbagai macam database management system (DBMS). ODBC memungkinkan interoperabilitas secara maksimal terhadap berbagai macam DBMS hanya dengan melalui satu interface. Ini dapat dikatakan bahwa suatu aplikasi akan berjalan secara independen. Pengguna aplikasi dapat menambah suatu software komponen yang dinamakan driver, untuk menciptakan suatu interface antara suatu aplikasi dan suatu DBMS spesifik. ODBC juga merupakan suatu protokol atau *Application Programming Interface* (API) database yang khusus digunakan untuk mengakses database relasional SQL. Secara singkat, begitu kita menginstall suatu driver ODBC dari suatu produk database, maka kita bisa mengakses isi dari database tersebut tanpa harus mengetahui kerumitan dibalikinya.

Untuk bisa diakses dari aplikasi client ODBC, Database *Environment* harus menyediakan software yang biasa disebut *driver* pada sistem client aplikasi tersebut berada. Driver tersebut mentranslasi fungsi-fungsi ODBC standard ke dalam bentuk yang dapat dimengerti oleh sumber data yang dituju, sehingga dapat diproses untuk direspon dengan pengiriman data ke aplikasi. Setiap sumber data menyediakan driver pada sistem client untuk digunakan dalam mengakses data dari sumber datanya.

ODBC biasanya di gunakan pada software aplikasi, seperti Visual basic, Delphi, dan lain-lain yang bertujuan untuk mengambil dan menggunakan data yang diinginkan dari database tertentu. Intinya, ODBC adalah media penghubung antara software aplikasi dengan database tertentu [MAN-04].

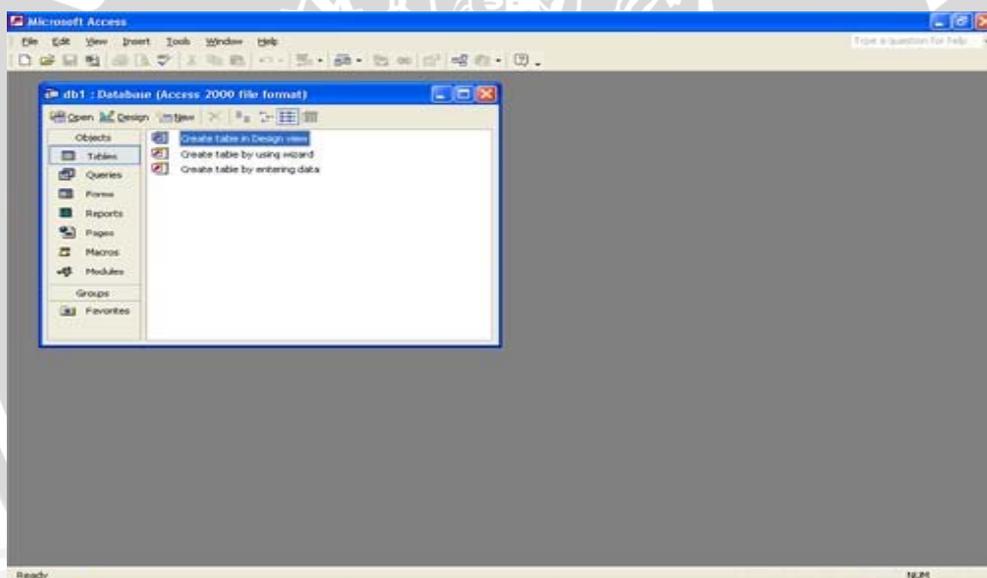
## 2.5 Microsoft Access

*Microsoft Access* merupakan perangkat lunak sistem manajemen database relasional (DBMS) dan berorientasi visual, dan berbasis *Windows*. Fungsi kinerja utamanya adalah digunakan untuk membuat dan memodifikasi tabel, formulir entri data, *query*, laporan, dan database. Pada kasus tertentu, dibutuhkan pengembangan aplikasi yang lebih rumit misalnya untuk validasi input yang lebih ekstensif atau *error handling*

yang lebih baik. Untuk situasi-situasi seperti ini, Access menyediakan bahasa pemrograman yang disebut *Visual Basic for Application* (VBA) yang mampu meningkatkan kemampuan dari *Access*.

Berikut beberapa fitur yang ditawarkan *Access*:

- *Database management system* (DBMS) yang benar-benar relasional
- *Wizard* yang mudah digunakan
- *Importing, exporting, dan linking* tabel
- *Form* dan laporan dengan fitur WYSIWYG
- *Multiple table queries and relationships*
- Kemampuan DDE dan OLE
- Dukungan modul VBA



**Gambar 2.12** Tampilan Jendela Kerja Microsoft Access

Di dalam *Access*, ada tujuh buah obyek yang biasa digunakan dalam pengoperasian sebuah database yaitu:

### 2.5.1 *Table*

Tabel adalah suatu obyek yang berfungsi untuk mendefinisikan dan menyimpan data menurut aturan tertentu. Sebuah tabel terdiri dari:

- *Fields* atau suatu unsur tabel yang berisi suatu item informasi yang spesifik dan diwakili oleh suatu kolom atau sel di dalam suatu datasheet.
- *Records* atau berupa baris pada tampilan Acces, yaitu yang memuat sebuah isi tabel data.
- *Primary Key* atau kunci primer satu atau lebih nama *field* yang berfungsi sebagai acuan tabel untuk dapat menghubungkan dua tabel atau lebih.
- *Indeks* atau indeks yang berfungsi untuk mempercepat penampilan data.

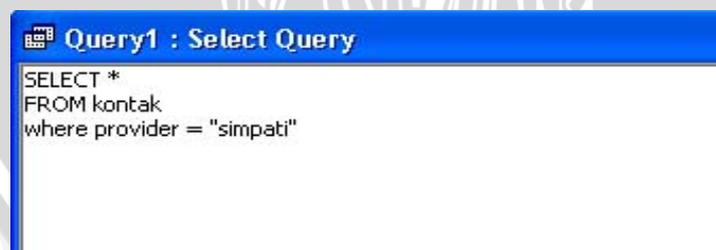


| kontak : Table |              |        |                  |          |
|----------------|--------------|--------|------------------|----------|
|                | nohp         | nama   | alamat           | provider |
| ▶              | 085633258987 | wachid | ijen no 4        | im3      |
|                | 081805000584 | yayan  | kembang kertas 9 | proxl    |
|                | 081334307368 | budi   | griyasahnta 320  | simpati  |
| *              |              |        |                  |          |

Gambar 2.13 Contoh Tampilan Tabel

### 2.5.2 Query

*Query* adalah perintah untuk mendefinisikan fungsi kerja data pada database. Query memanipulasi data dan mengendalikan manipulasi data tersebut melalui bahasa SQL (*Structured Query Language*). Suatu *query* dapat membawa bersama-sama data dari berbagai tabel untuk bertidak sebagai sumber data untuk suatu format, laporan atau halaman akses data. Contoh penggunaan *query* adalah untuk merelasikan record antara tabel, membuat *field* baru dan mengisi data pada *record*.



```

Query1 : Select Query
SELECT *
FROM kontak
where provider = "simpati"

```

Gambar 2.14 Contoh Tampilan Query

### 2.5.3 Form

*Form* atau formulir berfungsi untuk memanipulasi data seperti mengisi atau menampilkan data pada tabel. Jika tampilan yang ada pada tabel bersifat standar dan

kaku, maka *form* bisa mengubahnya sehingga tampilan lebih menarik serta ditambah dengan berbagai alat kontrol untuk mengeksekusi berbagai aplikasi.

Gambar 2.15 Contoh Tampilan *Form*

#### 2.5.4 *Report*

*Report* atau laporan berfungsi untuk memformat, mengkalkulasi, menampilkan dan mencetak data-data yang berasal dari tabel atau *query*. *Report* merupakan hasil akhir dalam pengolahan database. Isi *report* tidak dapat dimodifikasi, tetapi hanya menampilkan isi *tabel* dan atau *query*.

#### 2.5.5 *Data Access Page*

*Data Access Page* adalah suatu jenis khusus halaman *web* yang dirancang untuk bekerja dengan data dari suatu internet atau intranet. Data bisa berasal dari suatu Microsoft Access, SQL Server atau dari sumber lainnya, seperti Microsoft Excel.

#### 2.5.6 *Macro*

*Macro* berfungsi membantu dalam mengotomasikan suatu tugas. *Macro* berisi suatu set atau lebih tindakan yang masing-masing melakukan operasi tertentu, seperti membuka suatu *form* atau mencetak suatu laporan (*report*). Sebagai contoh dengan menggunakan *macro* untuk mencetak suatu laporan ketika seorang meng-klik suatu tombol.

#### 2.5.7 *Module*

*Module* adalah objek yang berisi sekumpulan *declaration*, *statement*, dan *procedure* yang dibuat pada satu file dengan menggunakan kode bahasa *Visual Basic*, yang akan menjalankan suatu operasi tertentu pada *Access*.

## 2.6 Mobile FBUS

Mobile FBUS merupakan komponen kontrol ActiveX *Freeware* yang akan digunakan dalam visual basic untuk membuat aplikasi otomatisasi pengambilan nomor antrian pada klinik dokter spesialis melalui SMS, komponen ini akan memudahkan kita untuk melakukan koneksi dengan handphone sehingga kita dapat melakukan pengendalian terhadap handphone, seperti; mengirim SMS, Monitoring, dan lain sebagainya [PEA-05].

Mobile FBUS adalah komponen yang cukup baik dan memadai untuk digunakan menjembatani antara aplikasi dan handphone. Mobile FBUS merupakan komponen yang dapat diperoleh melalui internet. Mobile FBUS terdiri dari dua versi, yaitu versi *Lite* dan *Professional*. Untuk versi *professional* kita harus membelinya dahulu ke perusahaan yang membuatnya, sedangkan untuk versi *lite* disediakan secara gratis sehingga kita dapat men-download-nya tanpa perlu membayar. Untuk dapat menggunakan komponen ini, pertama kita harus meng-copy file:

- gifbus15.dll
- MFBUS15.ocx
- MFBUS15.oca
- Regsvr32.exe

Ke dalam direktori aktif “<Windows>/System” untuk windows 98 atau “<Windows>/System32” untuk windows NT.

Setelah keempat file itu dicopy, kemudian daftarkan MFBUS15.ocx kedalam registry windows dengan mengetikkan perintah berikut di command prompt : regsvr32 MFBUS15.ocx

Proses instalasi komponen Mobile FBUS selesai setelah kita melakukan langkah terakhir tersebut. Sekarang komponen ini dapat digunakan di dalam aplikasi yang kita buat dengan menggunakan visual basic. Komponen ini akan muncul dengan nama “Mobile FBUS control v1.5” didalam visual basic komponen list.

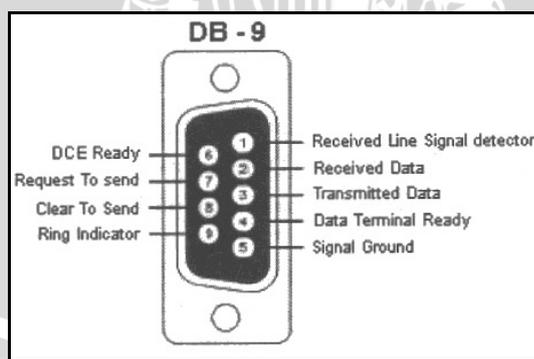
## 2.7 Port Serial

Standar sinyal komunikasi serial yang banyak digunakan adalah Standar RS232 yang dikembangkan oleh *Electronic Industry Association and the Telecommunications Industry Association* (EIA/TIA) yang pertama kali dipublikasikan pada tahun 1962. Ini terjadi jauh sebelum IC TTL populer sehingga sinyal ini tidak ada hubungan sama sekali dengan level tegangan IC TTL. Peralatan yang menggunakan media port serial untuk berkomunikasi dibagi dalam dua kelompok: *Data Communication Equipment* (DCE), dan *Data Terminal Equipment* (DTE). Contoh DCE adalah modem, ploter, sedang contoh DTE adalah terminal di komputer.

Berikut ini spesifikasi elektronis dari port serial dengan merujuk kepada *Electronic Industry Association* (EIA).

- "Space" (= logika 0) adalah tegangan antara +3 hingga +25 Volt.
- "Mark" (= logika 1) adalah tegangan antara -3 hingga -25 Volt.
- Daerah antara +3 Volt dan -3 Volt tidak didefinisikan.
- Tegangan rangkaian terbuka tidak boleh melebihi 25 Volt karena tegangan tersebut dapat merusak line driver pada saluran RS232 (terhadap ground).
- Arus hubungan singkat tidak boleh melebihi 500mA.

Konektor port serial tersedia dalam dua jenis, yaitu konektor 25 pin (DB-25) dan 9 pin (DB-9). Keduanya memiliki konektor jantan dan betina. Konektor port serial pada komputer (DTE) adalah jenis jantan. Bentuk dan ukuran DB-25 identik dengan port paralel, sedang DB-9 tampak seperti ditunjukkan Gambar 2.16.



**Gambar 2.16** Konektor Serial DB-9 Pada Bagian Belakang CPU

## 2.8 Sistem Klinik

Untuk lebih memberikan pelayanan kesehatan kepada masyarakat biasanya dokter membuka sendiri praktek di luar jam kerja rumah sakit, biasanya praktek di buka mulai jam 17: 00 sampai jam 21:00 malam. Dewasa ini praktek secara mandiri yang dilakukan oleh dokter mulai berkembang. Praktek yang biasanya dibuka di rumah pribadi kini berganti pada praktek bersama dalam suatu tempat.

Setiap pasien yang akan berobat atau berkonsultasi harus terdaftar. Karena tiap pasien akan diberikan kartu berobat dan nantinya akan dibuatkan rekam medis. Dalam satu hari tiap-tiap dokter spesialis umumnya menangani 20 sampai 30 pasien. Ketika berobat tiap pasien membutuhkan waktu yang sangat bervariasi tergantung dari sakit dan pemeriksaan yang dilakukan dokter. Dari pengamatan secara langsung di klinik praktek bersama rata-rata waktu pelayanan yang diperlukan ketika pasien berobat adalah 8 sampai 15 menit.

Selain dibatasi oleh banyaknya jumlah pasien untuk jam tutup praktek juga dibatasi oleh waktu. Praktek sedapat mungkin berakhir ketika jam 21:00, dengan demikian jika rata-rata pasien yang berobat berobat dalam 1 hari untuk dokter spesialis tertentu sebanyak 25 pasien maka waktu keseluruhan untuk menagani seluruh pasien tersebut adalah  $(25 \times 8) / 60 = 3 \text{ jam } 33 \text{ menit}$ .

Sistem antrian memberikan waktu 10 menit untuk pergantian antara pasien yang satu dengan yang lain dengan acuan tiap pasien berobat memerlukan waktu 8 menit. Hal ini akan memberikan suatu keuntungan yaitu jika pasien yang berobat memerlukan waktu lebih lama (lebih dari 8 menit) maka pasien sudah dalam keadaan menunggu dan siap untuk berobat.

## BAB III METODOLOGI PENELITIAN

Untuk merealisasikan tujuan penulisan Tugas akhir ini, langkah-langkah yang dilakukan adalah sebagai berikut:

### 3.1 Studi Literatur

Studi literatur yang dilakukan bertujuan untuk mengkaji hal-hal yang berhubungan dengan teori-teori yang mendukung dalam perencanaan dan perealisasiian aplikasi yaitu:

1. Kajian pustaka mengenai sistem kerja SMS, AT-Command, format data PDU (*Protocol Data Unit*).
2. Kajian pustaka mengenai struktur dan bahasa pemrograman yang digunakan pada *software* pengembang Visual Basic 6.0.
3. Kajian pustaka mengenai Microsoft Access yang digunakan sebagai RDBMS (*Relational Database Management System*).
4. Kajian antarmuka Handphone dan Komputer.
5. Kajian mengenai sistem klinik.dokter spesialis praktek bersama.

### 3.2 Perancangan Sistem Aplikasi

Pada tahap perancangan dibuat suatu blok diagram sistem aplikasi secara keseluruhan. Perancangan aplikasi dilakukan pada tiap-tiap blok untuk mempermudah analisis algoritma, perancangan, dan pemrograman. Perancangan didasarkan pada teori-teori yang ada yang nantinya akan diaplikasikan pada sistem yang dibuat.

### 3.3 Implementasi Sistem

Implementasi disusun sesuai dengan perancangan yang telah dibuat. Dalam implementasi perancangan ini disajikan dalam bentuk algoritma untuk lebih mudah dipahami struktur logika dari program yang dibuat.

### 3.4 Pengujian dan Analisa Program

Tahap ini adalah pengujian dan analisa baik itu sistem maupun program yang telah dibuat secara keseluruhan apakah telah sesuai dengan yang diharapkan.



### 3.5 Pengambilan Kesimpulan dan Saran

Tahap berikutnya dari penulisan adalah pengambilan kesimpulan dari program aplikasi yang dibuat. Pengambilan kesimpulan ini didasarkan pada kesesuaian antara teori dan praktek. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi serta menyempurnakan penulisan.



## BAB IV PERANCANGAN

### 4.1 Analisis Sistem

Analisis sistem dibutuhkan untuk menggambarkan kebutuhan yang dibutuhkan oleh pengguna, untuk membangun dasar bagi pembuatan desain perangkat lunak, dan untuk membatasi serangkaian persyaratan yang dapat divalidasi begitu perangkat lunak dibuat. Proses analisis meliputi analisis kebutuhan perangkat lunak dan pemodelan analisis perangkat lunak yaitu dengan membuat DFD.

#### 4.1.1 Analisis Kebutuhan Perangkat Lunak

Agar terpenuhinya kebutuhan kesehatan masyarakat yang merata maka pada suatu daerah umumnya di daerah tingkat kabupaten terdapat sebuah rumah sakit umum. Rumah sakit umum ini melayani hampir seluruh masalah kesehatan masyarakat yang berada pada daerah tersebut mulai dari masalah kesehatan yang ringan sampai dengan yang berat misalnya operasi. Agar lebih mudah dalam pengelolaan maka dari permasalahan yang ada di kelompokkan dalam beberapa kategori. Untuk permasalahan kesehatan yang khusus tentunya ditangani oleh dokter spesialis sehingga dapat teratasi dengan baik. Untuk lebih memberikan pelayanan kesehatan kepada masyarakat biasanya dokter membuka sendiri praktek di luar jam kerja rumah sakit, biasanya praktek di buka mulai jam 17: 00 sampai jam 21:00 malam.

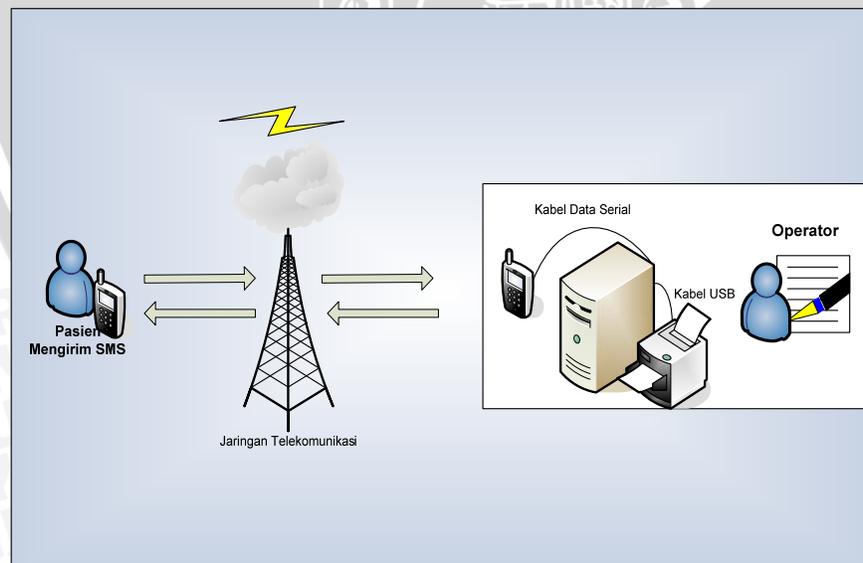
Dewasa ini praktek secara mandiri yang dilakukan oleh dokter mulai berkembang. Praktek yang biasanya dibuka di rumah pribadi kini berganti pada praktek bersama dalam suatu gedung. Jadi dalam satu gedung tersebut terdapat beberapa dokter spesialis dengan ruang praktek sesuai dengan spesialisasi masing masing dokter. Sebelum berobat ataupun berkonsultasi umumnya pasien harus memesan atau mengambil nomor antrian dahulu. Setelah pasien mendapatkan nomor antrian, pasien menunggu waktu tiba untuk berobat.

Sistem pengambilan nomor antrian pada klinik dokter spesialis dilakukan secara manual memiliki beberapa kekurangan diantaranya pasien tidak dapat mengetahui banyaknya antrian sehingga jika antrian banyak maka waktu tunggu untuk tiba waktu

berobat menjadi lama. Untuk memperbaiki sistem antrian yang lama maka dirancang sebuah program pengambilan nomor antrian dengan menggunakan SMS sebagai mediana. Dari penelitian yang dilakukan di klinik dokter spesialis praktek bersama dan pasien maka didapatkan kebutuhan sebagai berikut:

1. Pasien dapat mengambil nomor antrian sesuai dengan klinik yang dituju dengan menggunakan SMS. Nomor antrian yang diambil memberikan informasi nomor antrian dan jam untuk datang ke klinik.
2. Pihak dokter spesialis mendapatkan informasi data medis pasien yang berobat atau berkonsultasi. Dokter spesialis juga mendapatkan laporan jumlah pasien yang ditangani dalam hari tersebut.
3. Operator harus dapat memasukkan data, mengedit, menghapus data. Data yang dimaksud meliputi data pasien, data antrian, data hasil pemeriksaan dan laporan.
4. Sistem yang dibuat menggunakan *database* yang sudah dinormalisasi sebelumnya.
5. Sistem diimplementasikan dengan bahasa pemrograman Visual Basic 6.0 dan basis data MS Access 2000.
6. Sistem operasi yang digunakan adalah sistem operasi Windows.

Sistem pengambilan nomor antrian dengan menggunakan SMS [SIPENAS] seperti ditunjukkan dalam Gambar 4.1



**Gambar 4.1** Diagram Blok SIPENAS  
Sumber: [Perancangan]

Proses bisnis yang terjadi dalam klinik dokter spesialis praktek bersama adalah sebagai berikut .Pasien yang akan berobat dan mengambil nomor antrian harus terdaftar terlebih dahulu. Bagi pasien yang memiliki handphone dapat memanfaatkan layanan pengambilan nomor antrian dengan menggunakan SMS. Pasien diharapkan datang sesuai dengan jam yang telah ditentukan dan masuk dalam ruangan tunggu, Jika pasien datang tetapi terlambat maka ia harus menunggu 2 nomor antrian masuk berobat, setelah itu dipanggil untuk masuk. Pasien kemudian masuk dalam ruangan dokter spesialis untuk berobat. Hasil dari pemeriksaan dokter kemudian dicatat dalam rekam medis. Setelah pasien selesai berobat, pasien melakukan pembayaran. Biaya meliputi biaya berobat ditambah dengan biaya SMS bagi yang memanfaatkan pengambilan nomor antrian dengan SMS.

Program aplikasi pengambilan nomor antrian dengan menggunakan SMS diharapkan dapat memudahkan pasien mengambil nomor antrian yang akan berobat ke klinik dokter spesialis dengan menggunakan SMS. Sistem informasi rekam medis yang dibuat diharapkan mampu memberikan data yang lebih akurat dan cepat sehingga dapat memberikan pelayanan yang lebih cepat dan baik yang akhirnya meningkatkan kinerja sistem secara keseluruhan

#### **4.1.2 Spesifikasi Kebutuhan**

Spesifikasi kebutuhan dibutuhkan untuk menjelaskan kebutuhan perangkat lunak yang telah didefinisikan sebelumnya secara lebih detail dan tepat yang akan menjadi dasar bagi perancangan dan implementasi.

##### **Definisi :**

1. Pasien dapat mengambil nomor antrian sesuai dengan klinik yang dituju dengan menggunakan SMS. Nomor antrian yang diambil memberikan informasi nomor antrian dan jam untuk datang ke klinik [PRJ-001].

##### **Spesifikasi :**

- 1.1 Pasien harus dapat memperoleh informasi nomor antrian dan jam datang ke klinik dokter spesialis [PRJ-002].
- 1.2 Pasien harus terdaftar terlebih dulu [PRJ-003].

- 1.3 Pasien mengirimkan SMS dengan format yang telah ditentukan untuk dapat mengambil nomor antrian [PRJ-004].

**Definisi :**

2. Pihak dokter spesialis mendapatkan informasi data medis pasien yang berobat atau berkonsultasi. Dokter spesialis juga mendapatkan laporan jumlah pasien yang ditangani dalam hari tersebut [PRJ-005].

**Spesifikasi :**

- 2.1 Dokter menyerahkan hasil pemeriksaan kepada operator untuk dimasukkan datanya [PRJ-006].
- 2.2 Dokter mendapatkan data rekam medis pasien. Data ini diberikan oleh operator [PRJ-007].
- 2.3 Dokter mendapat laporan jumlah pasien yang ditanganinya dalam hari tersebut atau periode tertentu [PRJ-008].

**Definisi :**

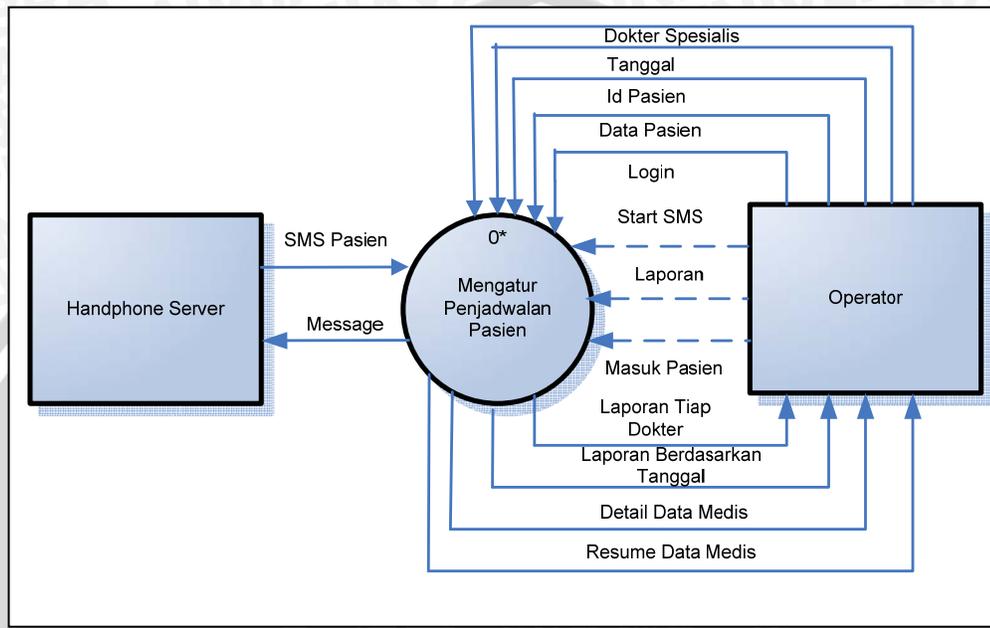
3. Operator harus dapat memasukkan data, mengedit, menghapus data. Data yang dimaksud meliputi data pasien, data hasil pemeriksaan dan laporan [PRJ-009].

**Spesifikasi :**

- 3.1 Operator harus login terlebih dahulu untuk dapat menjalankan aplikasi [PRJ-010].
- 3.2 Operator harus dapat melakukan pengisian data pasien dengan lengkap kedalam sistem, meliputi data :id\_pasien, nama\_pasien, alamat\_pasien, tanggal lahir, no\_handphone, no\_telephone, jenis\_kelamin [PRJ-011].
- 3.3 Operator harus dapat melakukan pengisian data hasil pemeriksaan kedalam sistem, meliputi data :id\_pasien, nama\_pasien, nama\_dokter, hasil pemeriksaan dan jenis\_obat [PRJ-012].
- 3.4 Operator harus dapat mencetak data rekam medis pasien untuk diberikan kepada dokter [PRJ-013].
- 3.5 Operator harus dapat mencetak laporan sesuai dengan hari tersebut atau sesuai dengan periode tertentu berdasarkan masing-masing dokter spesialis [PRJ-014].

### 4.1.3 Diagram Konteks

DFD yang pertama kali dibuat adalah DFD level 0 atau *Context Diagram*. Diagram konteks menjelaskan hubungan sistem dengan lingkungan atau kesatuan luar. *Context Diagram* seperti ditunjukkan dalam Gambar 4.2.



**Gambar 4.2** *Context Diagram* SIPENAS  
Sumber: [Perancangan]

Berdasarkan Gambar 4.2 proses SIPENAS mempunyai data masukan berupa:

1. login

*login* diberikan oleh operator untuk dilakukan proses validasi login.

Parameter yang digunakan adalah:

- a. *username*

Berisi masukan *username* yang diberikan oleh operator. Tipe data dari *username* adalah text dengan lebar *field* 15 karakter.

- b. *password*

Berisi masukan *password* yang diberikan oleh operator. Tipe data dari *password* adalah text dengan lebar *field* 15 karakter.

2. Data pasien  
Berisi data pasien meliputi nama, alamat,tanggal lahir,jenis kelamin data ini dimasukkan ketika proses daftar pasien.
3. Id Pasien  
Id pasien diberikan ketika pasien mendaftar dan diberikan secara otomatis oleh program
4. Tanggal  
Tanggal merupakan parameter masukan yang digunakan untuk mencetak laporan berdasarkan tanggal.
5. Dokter Spesialis  
Dokter spesialis merupakan parameter yang digunakan untuk mencetak laporan untuk tiap-tiap dokter spesialis
6. SMS Pasien  
Masukan dari pasien berupa SMS untuk pengambilan nomor antrian
7. Kontrol SMS  
Kontrol ini diberikan oleh operator untuk mengaktifkan proses terima dan kirim SMS secara otomatis.
8. Kontrol Masuk Pasien  
Kontrol ini diberikan oleh operator untuk mengaktifkan proses memasukkan data pasien
9. Kontrol Laporan  
Kontrol ini diberikan oleh operator untuk mengaktifkan proses mencetak laporan.

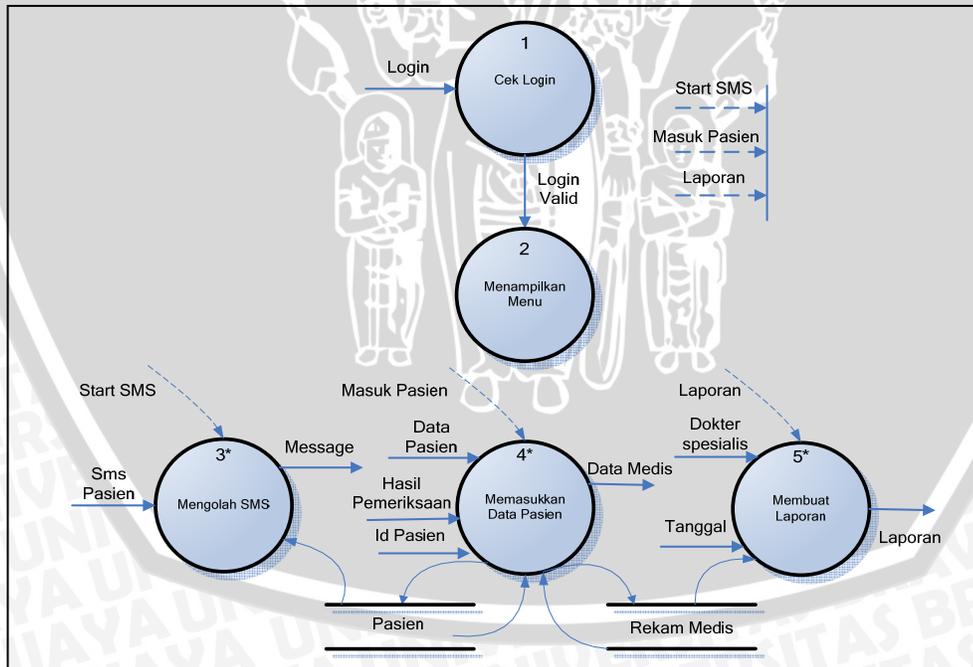
Berdasarkan Gambar 4.2 proses SIPENAS mempunyai data keluaran berupa:

1. *Message*  
Berisi data antrian dan jam datang ke klinik yang dikirimkan oleh program yang memiliki dua parameter utama yaitu no handphone pengirim SMS dan teks yang berisi informasi nomor antrian dan jam datang atau informasi kesalahan.
2. Laporan Tiap Dokter  
Laporan yang dicetak oleh operator berdasarkan parameter masukan dokter spesialis dan diberikan kepada masing-masing dokter spesialis.

3. Laporan Berdasarkan Tanggal  
Laporan yang dicetak oleh operator berdasarkan parameter masukan tanggal dan diberikan kepada dokter spesialis yang membutuhkan laporan tersebut.
4. Detail Data Medis  
Berisi data riwayat sakit pasien sebelumnya, data ini dicetak oleh operator kemudian diberikan kepada dokter spesialis yang membutuhkan.
5. Resume Data Medis  
Resume data medis merupakan tampilan data secara langsung berupa data grid ketika operator memasukkan data rekam medis.

#### 4.1.4 DFD Level 1

DFD Level 1 menggambarkan proses-proses yang lebih detail dari *context diagram* untuk Sistem pengambilan nomor antrian dengan menggunakan SMS seperti ditunjukkan dalam Gambar 4.3. Pada DFD Level 1 tersebut dijabarkan lebih detail proses-proses sebagai berikut:



Gambar 4.3 DFD Level 1 SIPENAS

Sumber: [Perancangan]

#### 1. Proses Cek Login ( Proses 1 )

Proses Validasi Login Operator akan terjadi setelah operator memasukkan username dan password. Proses ini merupakan proses untuk menentukan apakah Operator berhak untuk mengakses menu-menu yang ada pada sistem SIPENAS.

#### 2. Proses Menampilkan Menu ( Proses 2 )

Setelah login operator valid maka program akan menampilkan keseluruhan menu utama yang dapat diaktifkan dengan melakukan klik pada menu *pull down* ataupun pada *toolbar*.

#### 3. Proses Mengolah SMS ( Proses 3 )

Proses ini merupakan proses untuk menerima dan mengirim SMS.

Proses ini mempunyai masukan:

- SMS Pasien

SMS yang dikirimkan oleh pasien untuk mengambil nomor antrian. Ketika SMS masuk ada dua parameter yang akan diolah oleh program yaitu nomor handphone pengirim dan teks.

Proses ini mempunyai keluaran:

- Message

Message atau pesan yang diterima oleh pengambil nomor antrian memiliki dua status jika SMS yang dikirimkan memiliki status valid maka keluaran dari program akan menginformasikan nomor antrian dan jam datang. Jika status SMS tidak valid maka akan dikirimkan SMS yang menginformasikan pesan kesalahan.

Proses ini menggunakan *data store* :

- Pasien

Data pasien yang berasal dari data masukan pasien ketika pasien didaftarkan.

#### 4. Proses Memasukkan Data Pasien ( Proses 4 )

Proses ini digunakan untuk memasukkan data pasien. Data pasien yang dimasukkan meliputi data pasien ketika pasien didaftarkan dan data rekam medis pasien hasil dari pemeriksaan.

Proses ini mempunyai masukan:

- Data Pasien
- Hasil Pemeriksaan

- Id Pasien

Proses ini mempunyai keluaran:

- Data Medis

Proses ini menggunakan *data store* :

- Pasien
- Rekam Medis

*Data store* rekam medis digunakan untuk menyimpan data medis pasien hasil pemeriksaan.

#### 5. Proses Membuat Laporan ( Proses 5 )

Proses ini merupakan proses untuk membuat dan mencetak laporan dilakukan oleh operator. Operator memasukkan parameter untuk mencetak sebuah laporan. Laporan ini nantinya akan diberikan kepada dokter spesialis yang membutuhkan.

Proses ini mempunyai masukan:

- Dokter spesialis
- Tanggal

Proses ini mempunyai keluaran:

- Laporan

Laporan yang dicetak nantinya memiliki dua jenis yaitu laporan tiap dokter yang berisi banyaknya jumlah pasien yang ditangani oleh masing-masing dokter spesialis dan laporan berdasarkan tanggal

Proses ini menggunakan *data store* :

- Rekam Medis

Tabel 4.1 memperlihatkan proses – proses yang dilakukan ketika suatu kontrol diberikan kepada perangkat lunak

**Tabel 4. 1** Tabel Aktivasi Proses DFD level 1

| Kontrol             | PROSES |   |   |
|---------------------|--------|---|---|
|                     | 3      | 4 | 5 |
| <i>Start SMS</i>    | 1      |   |   |
| <i>Masuk Pasien</i> |        | 1 |   |
| <i>Laporan</i>      |        |   | 1 |

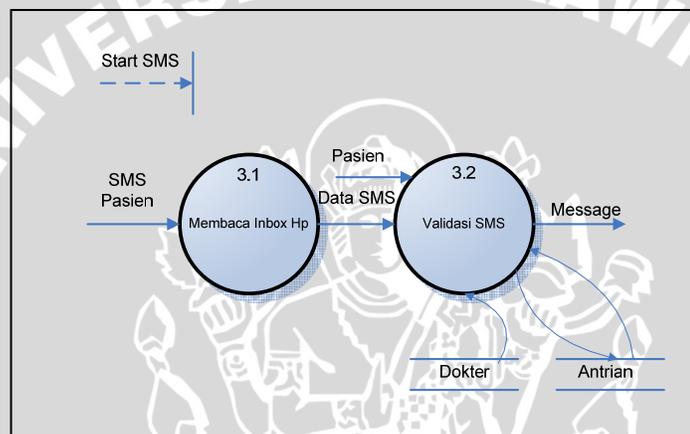
Sumber: [Perancangan]

Keterangan

- 1 = Proses aktif
- Kosong = Proses tidak aktif

#### 4.1.5 DFD Level 2 Mengolah SMS

Proses mengolah SMS dapat dijelaskan menjadi lebih detail lagi melalui DFD Level 2 dimana dimana proses tersebut dibagi menjadi proses-proses yang lebih spesifik, yaitu : proses membaca inbox dan proses validasi SMS. DFD level 2 mengolah SMS seperti ditunjukkan dalam Gambar 4.4.



**Gambar 4.4** DFD Level 2 mengolah SMS  
Sumber: [Perancangan]

Dalam DFD level 2 mengolah SMS terdapat kontrol start SMS yang berfungsi untuk mengaktifkan proses membaca inbox handphone untuk selanjutnya akan dilakukan validasi dan akan dikirimkan SMS balasan.

Dari Gambar 4.4, maka dapat dijabarkan lebih detail proses-proses tersebut sebagai berikut:

##### 1. Proses membaca inbox

Proses membaca inbox digunakan untuk mengambil SMS yang ada di inbox handphone server kemudian dibaca is SMS nya.

Proses ini mempunyai masukan:

- SMS pasien  
SMS pasien yang masuk dalam handphone server memiliki dua parameter utama yaitu nomor handphone pengirim dan teks SMS. Untuk teks yang dikirimkan memiliki format yang telah ditentukan yaitu [Klinik] Spasi [Nama Klinik] contoh “Kinik THT” untuk melakukan antrian pada klinik dokter spesialis THT sedangkan besar kecilnya huruf tidak berpengaruh.

## 2. Proses validasi SMS

Proses validasi SMS digunakan untuk memvalidasi data SMS yang telah dibaca. Validasi yang dilakukan meliputi nomor handphone pengirim SMS dan isi teks SMS. Setelah selesai divalidasi akan dikirimkan pesan kepada pengirim SMS. Jika proses validasi valid maka data yang dikirimkan kepada pengirim SMS berupa informasi nomor antrian dan jam datang. Jika proses validasi tidak valid maka akan dikirimkan pesan kesalahan.

Proses ini mempunyai keluaran:

- *Message*  
*Message* yang dikirimkan memiliki dua parameter yaitu no pengirim dan teks. Jika proses validasi SMS valid maka akan dikirimkan no antrian dan jam datang. Jika proses validasi SMS tidak valid maka akan dikirimkan pesan kesalahan.

Proses ini menggunakan *data store* :

- Dokter
- Pasien
- Antrian

*Data store* antrian digunakan untuk menyimpan data hasil validasi SMS valid berupa no antrian dan jam datang yang dikirimkan ke pasien.

Tabel 4.2 memperlihatkan proses – proses yang dilakukan ketika suatu kontrol diberikan kepada perangkat lunak

**Tabel 4.2** Tabel Aktivasi Proses DFD level 2 mengolah SMS

| Kontrol   | PROSES |     |
|-----------|--------|-----|
|           | 3.1    | 3.2 |
| Start SMS | 1      | 2   |

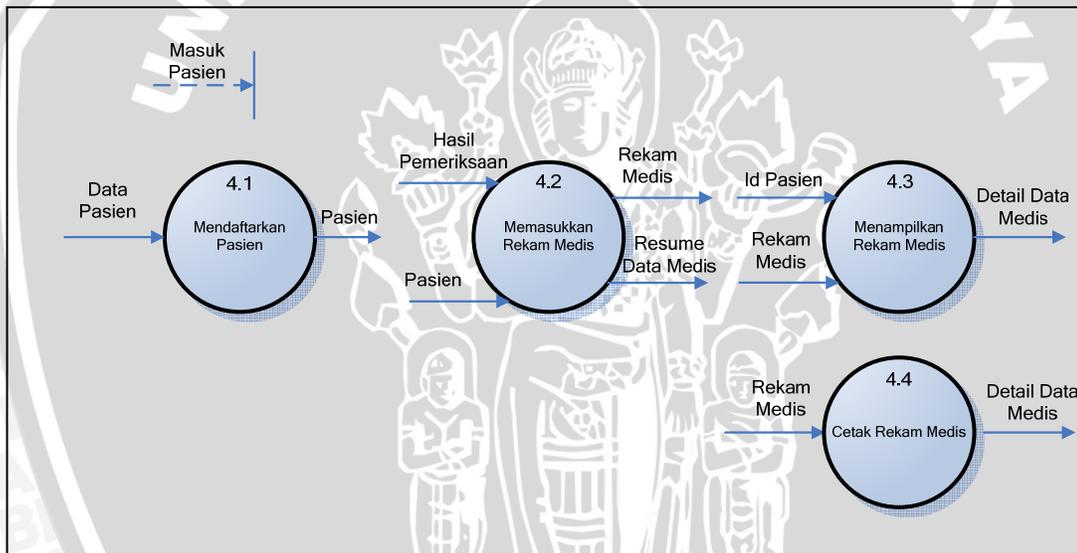
Sumber: [Perancangan]

Keterangan

- 1 = Proses aktif
- 2 = Proses berikutnya dikerjakan

**4.1.6 DFD Level 2 Memasukkan Data Pasien**

Proses memasukkan data pasien dapat dijelaskan menjadi lebih detail lagi melalui DFD Level 2 dimana dimana proses tersebut dibagi menjadi proses-proses yang lebih spesifik, yaitu : proses mendaftarkan pasien, memasukkan rekam medis, menampilkan rekam medis dan mencetak rekam medis. DFD level 2 proses memasukkan data pasien seperti ditunjukkan dalam Gambar 4.5.



**Gambar 4.5** DFD Level 2 memasukkan data pasien  
 Sumber: [Perancangan]

Dalam DFD level 2 memasukkan data pasien terdapat kontrol masuk pasien untuk mengaktifkan proses memasukkan data pasien .Dalam kontrol masuk pasien terdapat option pilih: daftar, rekam medis, tampil, cetak untuk mengaktifkan masing-masing proses tersebut.

Dari Gambar 4.5, maka dapat dijabarkan lebih detail proses-proses tersebut sebagai berikut:

1. Proses mendaftarkan pasien

Setiap pasien yang berobat ke klinik harus terdaftar. Pada proses ini dilakukan pendaftaran pasien dengan memasukkan data pasien secara lengkap.

Proses ini mempunyai masukan:

- Data pasien  
Data pasien yang dimasukkan meliputi nama pasien, alamat, tanggal lahir, nomor handphone, no telepon dan jenis kelamin.

Proses ini menggunakan *data store* :

- Pasien

2. Proses memasukkan rekam medis

Setelah pasien berobat, dokter akan memberikan hasil pemeriksaan pasien kepada operator untuk dimasukkan dalam rekam medis.

Proses ini mempunyai masukan:

- Hasil Pemeriksaan.  
Data hasil pemeriksaan dokter meliputi no transaksi, nama pasien, dokter, hasil pemeriksaan dan jenis obat akan dimasukkan dalam rekam medis.

Proses ini mempunyai keluaran:

- Resume data medis  
Resume data medis merupakan tampilan data setelah hasil pemeriksaan dimasukkan dalam rekam medis.

Proses ini menggunakan *data store* :

- Pasien
- Rekam medis

3. Proses menampilkan rekam medis

Proses menampilkan rekam medis digunakan untuk menampilkan detail data medis pasien sebelum data ini dicetak. Untuk menampilkan detail data pasien operator harus memasukkan parameter id pasien.

Proses ini mempunyai masukan:

- Id pasien

Id pasien merupakan nomor id yang bersifat unik dan dimiliki oleh pasien yang telah terdaftar.

Proses ini mempunyai keluaran:

- Detail data medis

Detail data medis merupakan data hasil proses seleksi dalam *data store* yang ditampilkan meliputi keseluruhan data pasien beserta riwayat sakit pasien.

Proses ini menggunakan *data store* :

- Rekam medis

#### 4. Proses cetak rekam medis

Proses ini digunakan untuk mencetak detail rekam medis pasien untuk diberikan ke dokter spesialis yang membutuhkan

Proses ini mempunyai keluaran:

- Detail data medis

Detail data medis yang telah dicetak dalam kertas.

Proses ini menggunakan *data store* :

- Rekam medis

Tabel 4.3 memperlihatkan proses – proses yang dilakukan ketika suatu kontrol diberikan kepada perangkat lunak.

**Tabel 4.3** Tabel Aktivasi Proses DFD level 2 memasukkan data pasien

| Kontrol<br>Masuk Pasien | PROSES |     |     |     |
|-------------------------|--------|-----|-----|-----|
|                         | 4.1    | 4.2 | 4.3 | 4.4 |
| <i>Daftar</i>           | 1      |     |     |     |
| <i>Rekam Medis</i>      |        | 1   |     |     |
| <i>Tampil</i>           |        |     | 1   |     |
| <i>Cetak</i>            |        |     |     | 1   |

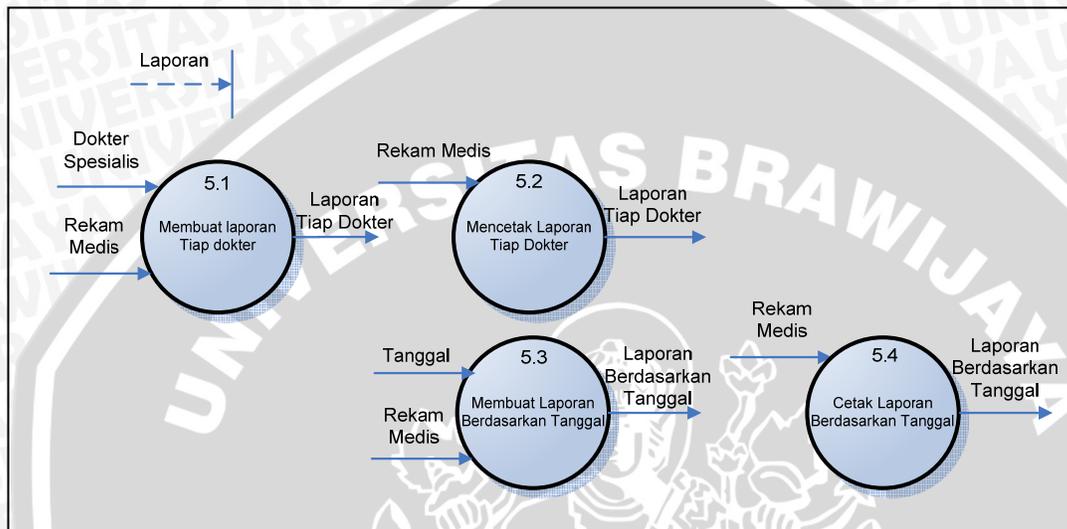
Sumber: [Perancangan]

Keterangan

- 1 = Proses aktif
- Kosong = Proses tidak aktif

#### 4.1.7 DFD Level 2 Membuat Laporan

Proses membuat laporan digunakan untuk melakukan proses seleksi dalam *data store* kemudian dari hasil seleksi tersebut data akan dicetak. DFD Level 2 membuat laporan dibagi menjadi proses-proses yang lebih spesifik seperti ditunjukkan dalam Gambar 4.6.



**Gambar 4.6** DFD Level 2 membuat laporan  
Sumber: [Perancangan]

Dalam DFD level 2 membuat laporan terdapat kontrol laporan untuk mengaktifkan proses membuat laporan. Dalam kontrol laporan terdapat option pilih: laporan tiap dokter, cetak laporan tiap dokter, laporan sesuai tanggal dan cetak laporan berdasarkan tanggal untuk mengaktifkan masing-masing proses tersebut.

Dari Gambar 4.6, maka dapat dijabarkan lebih detail proses-proses tersebut sebagai berikut:

1. Proses membuat laporan tiap dokter.

Proses membuat laporan tiap dokter digunakan untuk melakukan seleksi dalam *data store* rekam medis sesuai dengan parameter masukan yaitu dokter spesialis hasilnya akan langsung ditampilkan.

Proses ini mempunyai masukan:

- Dokter spesialis

Parameter dokter spesialis yang dimasukkan adalah nama kliniknya. Jadi untuk membuat laporan tiap dokter cukup memasukkan nama klinik contoh klinik THT.

Proses ini mempunyai keluaran:

- Laporan tiap dokter

Hasil seleksi dalam *data store* akan ditampilkan sesuai dengan parameter masukan dokter spesialis.

Proses ini menggunakan *data store* :

- Rekam medis

## 2. Proses mencetak laporan tiap dokter

Proses ini digunakan untuk mencetak hasil dari proses membuat laporan tiap dokter

Proses ini mempunyai keluaran:

- Laporan tiap dokter

Hasil proses membuat laporan tiap dokter dicetak dalam kertas.

Proses ini menggunakan *data store* :

- Rekam medis

## 3. Proses membuat laporan berdasarkan tanggal

Proses membuat laporan berdasarkan tanggal digunakan untuk melakukan seleksi dalam *data store* rekam medis sesuai dengan parameter masukan yaitu tanggal hasilnya akan langsung ditampilkan.

Proses ini mempunyai masukan:

- Tanggal

Tanggal yang dimasukkan adalah tanggal sistem sekarang atau rentang tanggal tertentu.

Proses ini mempunyai keluaran:

- Laporan berdasarkan tanggal

Hasil seleksi dalam *data store* akan ditampilkan sesuai dengan parameter masukan berupa tanggal sistem sekarang atau rentang tanggal tertentu

Proses ini menggunakan *data store* :

- Rekam medis

#### 4. Proses mencetak laporan berdasarkan tanggal

Proses ini digunakan untuk mencetak hasil dari proses membuat laporan berdasarkan tanggal.

Proses ini mempunyai keluaran:

- Laporan berdasarkan tanggal

Hasil proses membuat laporan tiap dokter dicetak dalam kertas

Proses ini menggunakan *data store* :

- Rekam medis

Tabel 4.4 memperlihatkan proses – proses yang dilakukan ketika suatu kontrol diberikan kepada perangkat lunak.

**Tabel 4.4** Tabel Aktivasi Proses DFD level 2 membuat laporan

| Kontrol<br>Laporan     | PROSES |     |     |     |
|------------------------|--------|-----|-----|-----|
|                        | 5.1    | 5.2 | 5.3 | 5.4 |
| Laporan tiap dokter    | 1      |     |     |     |
| Cetak lpr tiap dokter  |        | 1   |     |     |
| Laporan sesuai tanggal |        |     | 1   |     |
| Cetak lpr tanggal      |        |     |     | 1   |

Sumber: [Perancangan]

Keterangan

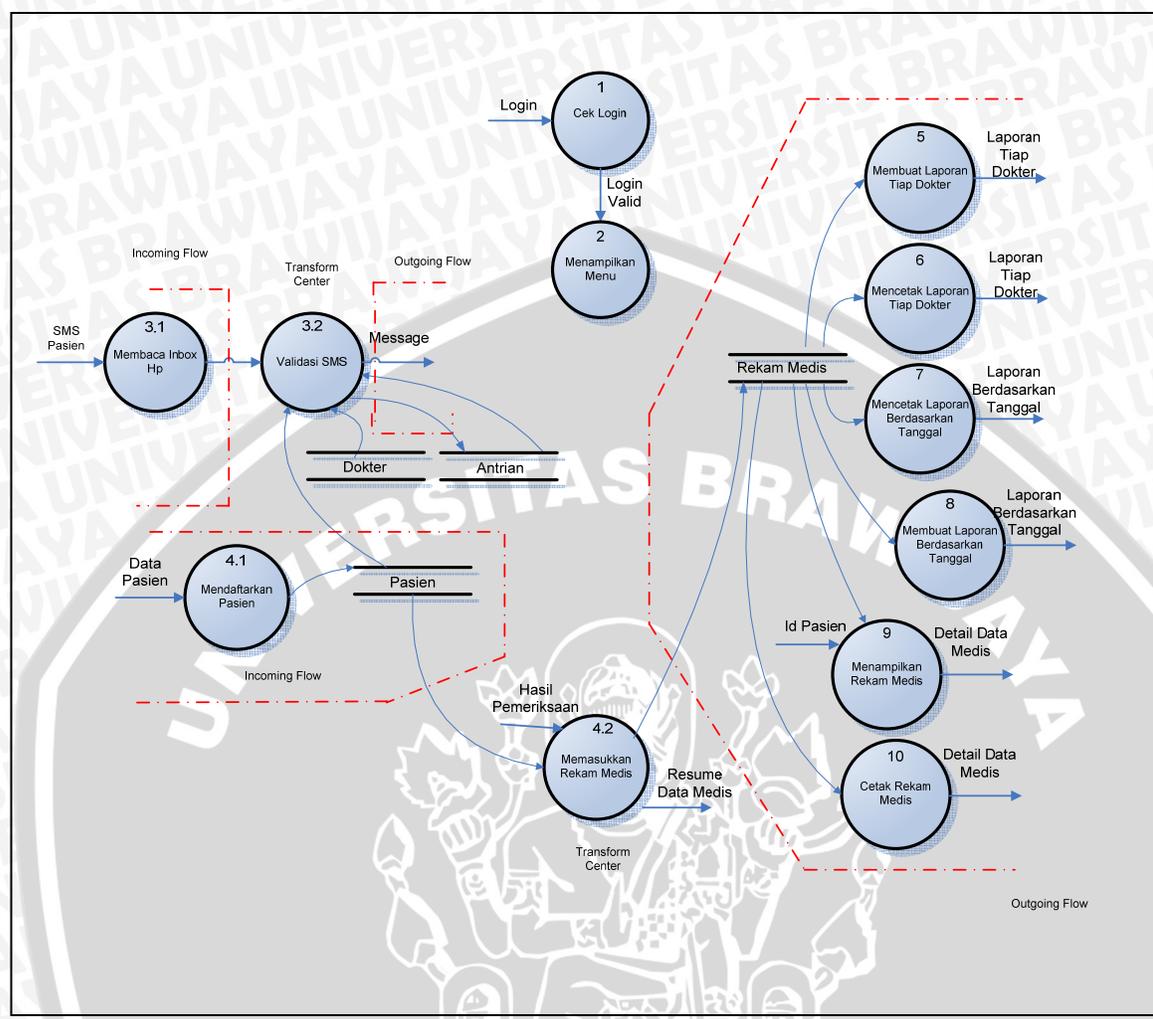
- 1 = Proses aktif
- Kosong = Proses tidak aktif

## 4.2 Perancangan Sistem

Perancangan sistem digunakan sebagai representasi rekayasa dari perangkat lunak yang akan dibangun. Perancangan yang dibuat meliputi desain arsitektur, perancangan *user interface*, perancangan basis data dan perancangan koneksi.

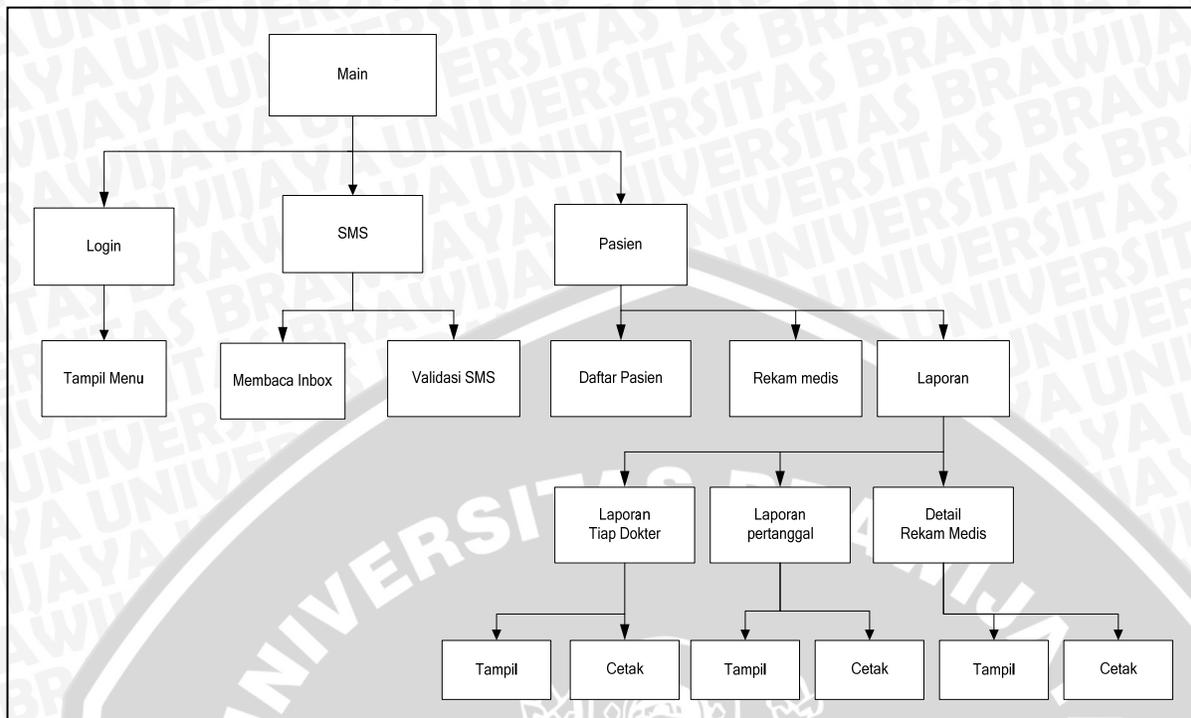
### 4.2.1 Perancangan Desain Arsitektur

Dari DFD Level 1 SIPENAS yang telah dibuat dapat disimpulkan bahwa DFD mempunyai karakteristik tipe *transform flow*. Desain yang berorientasi pada aliran data merupakan suatu metode desain arsitektur yang mengijinkan transisi yang baik dari model analisis ke deskripsi desain dari struktur program.



**Gambar 4.7** Aliran transformasi pada DFD Level 1  
 Sumber: [Perancangan]

Dari pembagian batas *incoming flow*, *transform center*, *outgoing flow* yang telah dibuat dibangun pemfaktoran tingkat pertama seperti pada gambar 4.8.



**Gambar 4.8** Pemfaktoran tingkat pertama SIPENAS  
Sumber: [Perancangan]

Dari gambar 4.8, Pemfaktoran tingkat pertama terdiri dari :

1. Modul Login

Modul login digunakan untuk proses login operator sebelum menjalankan sistem.

2. Modul Tampil menu

Setelah operator melakukan login dan valid maka semua menu dapat diakses.

3. Modul SMS

Modul SMS menangani fungsi terima dan kirim SMS.

4. Modul membaca inbox

Modul ini berfungsi untuk mengambil SMS pada handphone server kemudian membacanya.

5. Modul validasi SMS

Modul ini berfungsi untuk melakukan proses searching dan validasi SMS dan kemudian mengirim SMS balasan.

6. Modul pasien

Modul ini berfungsi untuk menangani data pasien.

7. Modul daftar pasien

Modul ini berfungsi untuk memasukkan data lengkap pasien ketika pasien didaftarkan.

8. Modul rekam medis

Modul ini berfungsi untuk mencatat hasil pemeriksaan pasien.

9. Modul laporan

Modul ini berfungsi untuk mencetak laporan.

10. Modul laporan tiap dokter

Modul ini berfungsi untuk mencetak laporan tiap dokter..

11. Modul laporan pertanggal

Modul ini berfungsi untuk mencetak laporan pertanggal.

12. Modul detail rekam medis

Modul ini berfungsi untuk mencetak detail rekam medis pasien.

13. Modul tampil

Modul ini berfungsi untuk melakukan seleksi dalam database dan hasilnya ditampilkan

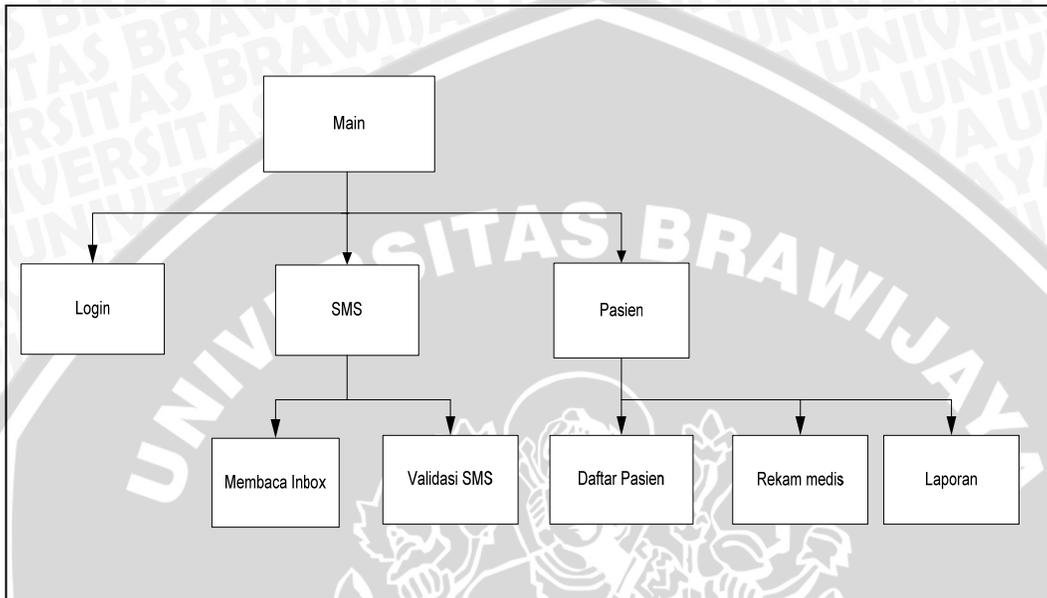
14. Modul cetak

Modul ini berfungsi untuk mencetak data ditampilkan oleh modul tampil.

Untuk menghasilkan struktur program yang lebih baik maka dilakukan proses penyaringan dari pemfaktoran tingkat pertama. Beberapa proses dapat digabungkan menjadi satu modul untuk mendapatkan struktur yang lebih baik dan efisien serta dapat diimplementasikan dengan mudah.

Modul login dan modul tampil menu dapat digabungkan menjadi satu. Modul tampil menu merupakan modul sederhana yang ditampilkan setelah proses login berhasil dilakukan. Untuk modul membaca inbox dan validasi SMS dimasukkan menjadi satu dalam modul SMS. Sedangkan modul tampil, modul cetak dan modul di atasnya dapat digabungkan menjadi satu modul yaitu modul laporan. Setelah dilihat sebenarnya modul laporan yang menyeleksi menjadi keluaran beberapa laporan tergantung dari parameter masukan dari operator.

Untuk modul detail rekam medis dilakukan penggabungan dengan modul rekam medis melalui pertimbangan praktis dan logis karena detail rekam medis sangat erat hubungannya dengan modul rekam medis dan bukan merupakan jenis laporan. Hasil pemfaktoran tingkat II untuk SIPENAS seperti ditunjukkan dalam Gambar 4.9.



**Gambar 4.9** Pemfaktoran tingkat dua SIPENAS  
Sumber: [Perancangan]

#### 4.2.2 Perancangan Komponen

Perancangan Komponen menjelaskan tentang prosedur dan variabel global yang ada pada setiap modul. Perancangan Komponen meliputi:

##### 1. Modul Login

- Prosedur Login Click

Prosedur Login click berfungsi untuk melakukan validasi login terhadap username dan password yang diberikan oleh operator.

Return type : Void

Argumen : String username, String password

- Prosedur Tampil Main Menu

Prosedur ini akan dipanggil ketika prosedur login memberikan nilai valid.

Return type : Void

## 2. Modul Main

- Prosedur Koneksi Database

Prosedur konek database digunakan untuk melakukan koneksi database dengan menggunakan driver database yang sesuai dan telah disetting dalam ODBC. Ketika *form* main sudah aktif maka koneksi database akan dibuka dan ketika *form* main ditutup maka koneksi database akan ditutup.

Return type : Void

- Prosedur Tampil click

Prosedur tampil digunakan untuk menampilkan *form-form* yang ada dalam main menu seperti *form* pasien, antrian, rekam medis dan laporan.

Return type : Void

Argumen : Int *form*

## 3. Modul SMS

- Prosedur Koneksi Handphone

Prosedur ini digunakan untuk melakukan koneksi antara handphone dengan komputer melalui kode program.

Return type : Void

- Prosedur Membaca Inbox

Prosedur membaca inboks digunakan untuk membaca SMS yang masuk pada inbox handphone.

Return type : Void

Argumen : Int SMS

- **Prosedur Validasi SMS**

Prosedur validasi SMS digunakan untuk memvalidasi SMS yaitu nomor handphone dan isi teks SMS. Dari dua parameter tersebut akan dilakukan proses pencarian dalam *data store*.

Return type : Void

- **Prosedur Membalas SMS**

Prosedur balas SMS digunakan untuk mengirimkan balasan SMS yang isinya tergantung dari hasil validasi. Jika validasi bernilai valid maka akan dikirimkan nomor antrian dan jam datang, jika hasil validasi tidak valid akan dikirimkan pesan kesalahan.

Return type : Void

Argumen : String no\_pengirim, String isi SMS

#### 4. Modul Pasien

- **Prosedur Simpan click**

Prosedur ini digunakan untuk menyimpan data lengkap pasien yang telah dimasukkan dalam *form* pasien

Return type : Void

- **Prosedur Edit click**

Prosedur ini digunakan untuk mengambil data pasien yang telah dimasukkan dan menampilkannya untuk kemudian dilakukan perubahan terhadap salah satu data pasien. Operator harus memasukkan id pasien untuk dapat menampilkan data pasien.

Return type : Void

- **Prosedur Delete click**

Prosedur ini digunakan untuk menghapus data pasien sesuai dengan id pasien yang dimasukkan oleh operator.

Return type : Void

## 5. Modul Rekam Medis

- Prosedur Simpan Rekam medis click

Prosedur ini digunakan untuk menyimpan data rekam medis pasien.

Return type : Void

- Prosedur Lihat click

Prosedur ini digunakan untuk mengambil data rekam medis dalam *data store* untuk kemudian ditampilkan hasilnya sesuai dengan id pasien yang dimasukkan oleh Operator.

Return type : Void

- Prosedur Cetak click

Prosedur ini digunakan untuk mencetak detail rekam medis pasien yang telah ditampilkan.

Return type : Void

## 6. Modul Laporan

- Prosedur tampil laporan click

Prosedur ini digunakan untuk mengambil parameter laporan yang dimasukkan oleh operator untuk kemudian dilakukan proses seleksi dalam data store dan hasilnya akan ditampilkan untuk kemudian siap dicetak..

Return type : Void

- Prosedur cetak click

Prosedur ini digunakan untuk mencetak laporan.

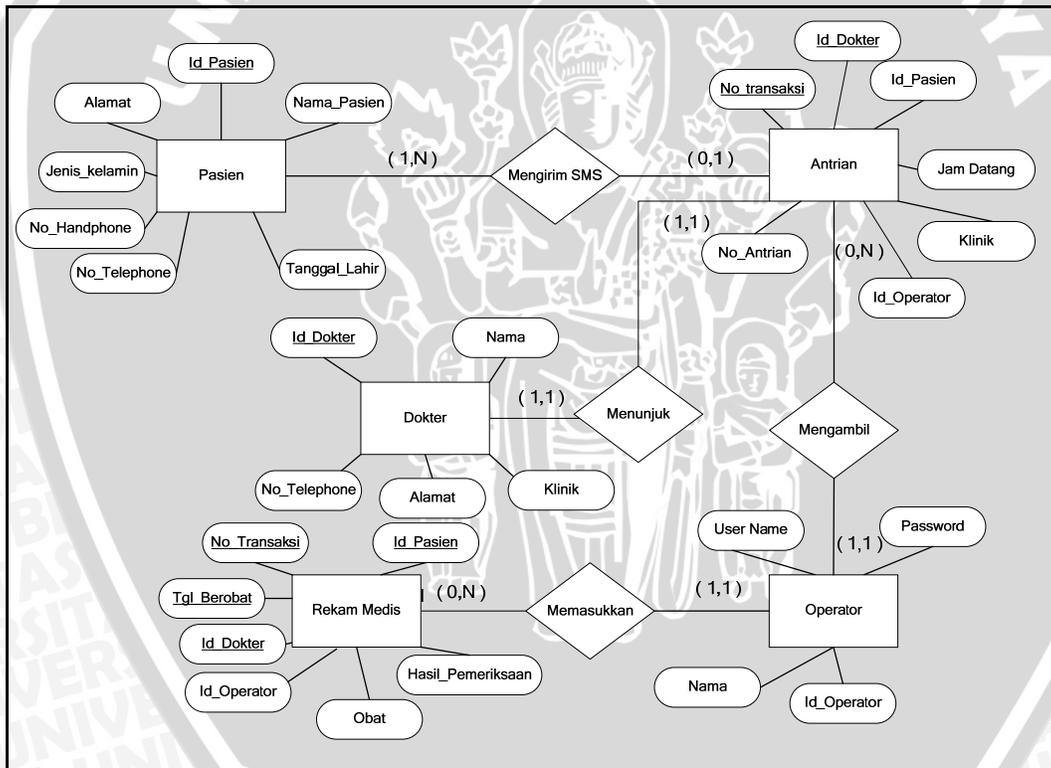
Return type : Void

### 4.2.3 Perancangan Basis Data

Perancangan basis data dilakukan agar Sistem pengambilan nomor antrian dengan menggunakan SMS dan sistem rekam medisnya menjadi basis data yang efisien dalam penggunaan ruang penyimpanan, cepat dalam pengaksesan dan mudah dalam manipulasi data. Perancangan basis data dapat dilakukan dengan menggunakan diagram ER, *data object description* dan normalisasi data

#### Entity Relationship Diagram (Diagram ER)

*Entity relationship diagram* (Diagram ER) mendefinisikan hubungan antar tabel yang ada didalam basis data melalui sebuah hubungan entitas. Hubungan ini juga memiliki tingkat derajat seperti satu ke satu, satu ke banyak, atau banyak ke satu. *Entity Relationship Diagram* dari basis data SIPENAS seperti ditunjukkan dalam Gambar 4.10.



Gambar 4.10. Entity-Relationship Diagram  
 Sumber: [Perancangan]

#### 4.2.3.1 Struktur Tabel

Struktur tabel menjelaskan secara rinci mengenai atribut-atribut yang dimiliki oleh masing-masing tabel yang ada pada basis data sesuai dengan *Entity Relationship Diagram*. Tabel – tabel yang dibuat dalam basis data adalah sebagai berikut :

- **Pasien.** Tabel pasien berisi data-data pasien yang dimasukkan ketika pasien pertama kali terdaftar. Data-data pada tabel pasien meliputi nama, alamat, tanggal lahir, nomor handphone, jenis kelamin disimpan pada tabel ini. Sebagai *primary key* nya adalah *id\_pasien*. Struktur Tabel pasien ditunjukkan dalam Tabel 4.5.

**Tabel 4.5** Struktur Tabel Pasien

| NamaTabel: Pasien |                       |        |             |             |
|-------------------|-----------------------|--------|-------------|-------------|
| Nama Field        | Deskripsi Field       | Tipe   | Lebar       | Keterangan  |
| Id_Pasien         | Nomor Id Pasien       | Number | (5)         | Primary Key |
| Nama_Pasien       | Nama Lengkap Pasien   | Text   | ( 35)       | Not Null    |
| Alamat            | Alamat Rumah Pasien   | Text   | (90 )       | Not Null    |
| Tanggal_Lahir     | Tanggal Lahir Pasien  | Date   | Simple Date | Not Null    |
| No_Handphone      | No Handphone Pasien   | Number | (15)        | Allow Null  |
| No_Telephone      | No Telephone Rumah    | Number | (15)        | Allow Null  |
| Jenis_Kelamin     | Laki-Laki / Perempuan | Text   | (1)         | Not Null    |

- **Operator.** Tabel operator berisi data operator yang menjalankan program sistem antrian dengan menggunakan SMS dan sistem informasi rekam medis. Struktur Tabel Operator ditunjukkan dalam Tabel 4.6.

**Tabel 4.6** Struktur Tabel Operator

| <b>NamaTabel: Operator</b> |                  |               |       |                    |
|----------------------------|------------------|---------------|-------|--------------------|
| Nama Field                 | Deskripsi Field  | Tipe          | Lebar | Keterangan         |
| Id_ Operator               | Id Operator      | <i>Number</i> | (2)   | <i>Primary Key</i> |
| User_ Name                 | Nama Untuk Login | <i>Text</i>   | (15)  | <i>Not Null</i>    |
| Password                   | Password Login   | <i>Text</i>   | (15)  | <i>Not Null</i>    |
| Nama_ Operator             | Nama Operator    | <i>Text</i>   | (35)  | <i>Not Null</i>    |
| Aktif_ Operator            | Status Aktif     | <i>Text</i>   | (3)   | <i>Not Null</i>    |

- **Antrian.** Tabel Antrian berisi data antrian pasien . Semua antrian yang melalui SMS disimpan dalam tabel ini. Struktur Tabel Antrian ditunjukkan dalam Tabel 4.7.

**Tabel 4.7** Struktur Tabel Antrian

| <b>NamaTabel: Antrian</b> |                 |               |                    |                    |
|---------------------------|-----------------|---------------|--------------------|--------------------|
| Nama Field                | Deskripsi Field | Tipe          | Lebar              | Keterangan         |
| No_ Transaksi             | Nomor Transaksi | <i>Number</i> | (5)                | <i>Primary Key</i> |
| Id_ Pasien                | Id Pasien       | <i>Number</i> | (5)                | <i>Foreign Key</i> |
| Id_ Operator              | Id Operator     | <i>Number</i> | (2)                | <i>Foreign Key</i> |
| Id_ Dokter                | Id Dokter       | <i>Number</i> | (2)                | <i>Foreign Key</i> |
| No_ Antrian               | Nomor Antrian   | <i>Number</i> | (2)                | <i>Not Null</i>    |
| Klinik                    | Jenis Klinik    | <i>Text</i>   | (15)               | <i>Not Null</i>    |
| Jam_ Datang               | Waktu Datang    | <i>Time</i>   | <i>Simple Time</i> | <i>Not Null</i>    |
| Status                    | Status Antrian  | <i>Text</i>   | (1)                | <i>Not Null</i>    |
| No_ Handphone             | Nomor Handphone | <i>Text</i>   | (15)               | <i>Allow Null</i>  |
| Tanggal                   | Tanggal SMS     | <i>Date</i>   | <i>Simple Date</i> | <i>Not Null</i>    |
| Isi_ SMS                  | Isi SMS         | <i>Text</i>   | (15)               | <i>Allow Null</i>  |

- **Rekam Medis.** Tabel rekam medis berisi data riwayat sakit pasien. Ketika pasien berobat maka akan terlihat daftar riwayat sakitnya. Setelah pasien berobat maka data-data seperti penyakit, obat disimpan dalam tabel ini. Struktur Tabel Berobat ditunjukkan dalam Tabel 4.8.

Tabel 4.8 Struktur Tabel Rekam Medis

| Nama Tabel: Rekam Medis |                          |        |             |             |
|-------------------------|--------------------------|--------|-------------|-------------|
| Nama Field              | Deskripsi Field          | Tipe   | Lebar       | Keterangan  |
| No_Transaksi            | Nomor Transaksi          | Number | (5)         | Foreign Key |
| Id_Pasien               | Nomor Id Pasien          | Number | (5)         | Foreign Key |
| Id_Dokter               | Nomor Id Dokter          | Number | (2)         | Foreign Key |
| Tanggal_Berobat         | Tanggal Berobat Pasien   | Date   | Simple Date | Not Null    |
| Hasil_Pemeriksaan       | Hasil Pemeriksaan Dokter | Text   | (255)       | Not Null    |
| Obat                    | Obat Yg Diberikan        | Text   | (100)       | Not Null    |

- **Dokter.** Tabel Dokter berisi data dokter spesialis yang membuka praktek bersama. Data dokter spesialis seperti nama, alamat, no telepon dan klinik disimpan pada tabel ini. Sebagai primary key nya adalah Id\_Dokter. Struktur Tabel Dokter ditunjukkan dalam Tabel 4.9.

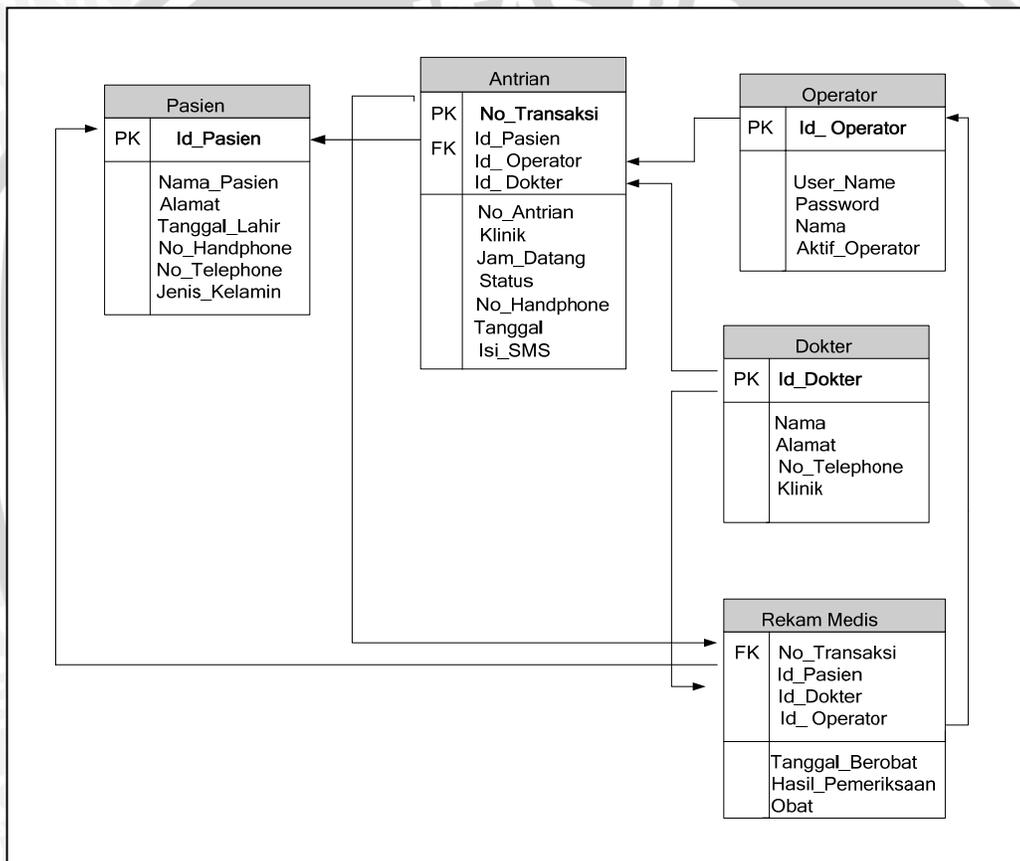
Tabel 4.9 Struktur Tabel Dokter

| Nama Tabel: Dokter |                    |        |       |             |
|--------------------|--------------------|--------|-------|-------------|
| Nama Field         | Deskripsi Field    | Tipe   | Lebar | Keterangan  |
| Id_Dokter          | Nomor Id Dokter    | Number | (2)   | Primary Key |
| Nama               | Nama Dokter        | Text   | (35)  | Not Null    |
| Alamat             | Alamat Rumah       | Text   | (90)  | Not Null    |
| No_Telephone       | No Telephone Rumah | Number | (15)  | Allow Null  |
| Klinik             | Klinik Spesialis   | Text   | (15)  | Not Null    |

### 4.2.3.2 Relasi Tabel

Setelah membuat model dengan *Entity relationship diagram* dan membuat struktur tabel dari masing-masing entitas. Selanjutnya dilakukan relasi dari tabel-tabel tersebut kedalam bentuk fisik.

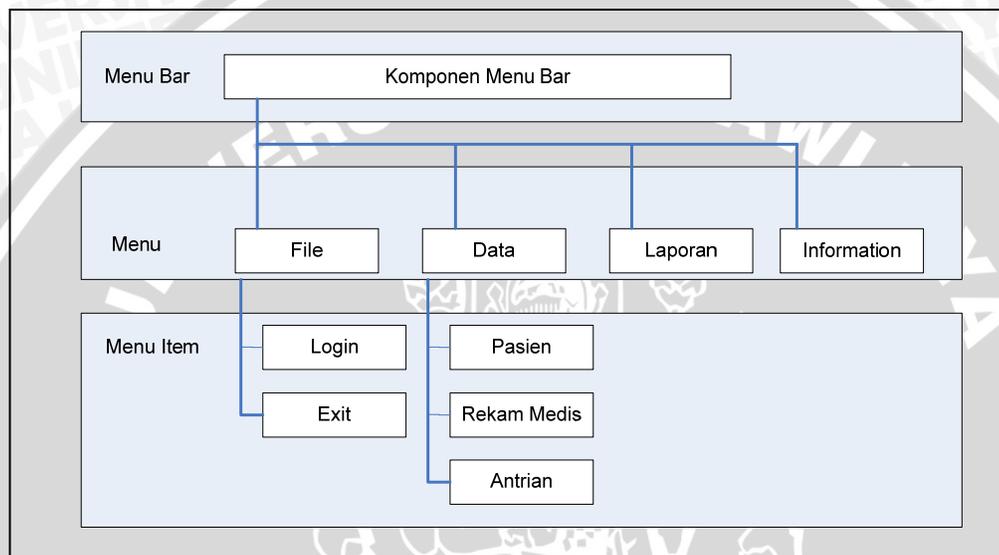
Relasi antar tabel ditandai dengan notasi PK (*Primary Key*) untuk *key* pada tabel dan FK (*Foreign key*) yang digunakan untuk tabel yang berelasi dengan *primary key*. Rancangan relasi antar tabel untuk sistem pengambilan nomor antrian dengan menggunakan SMS dan sistem rekam medis ditunjukkan dalam Gambar 4.11.



Gambar 4.11. Relasi Tabel Fisik SIPENAS  
 Sumber: [Perancangan]

#### 4.2.4 Perancangan User Interface

Antarmuka perangkat lunak SIPENAS menggunakan jenis MDI (*Multiple Document Interface*). Keuntungan penggunaan MDI adalah *form-form* yang akan ditampilkan pada aplikasi dapat diorganisasi dengan baik, sehingga pengguna aplikasi tidak mengalami kesulitan. Selain itu pada *form* utama juga terdapat menu *pull down* yang dapat juga digunakan untuk memilih *form* aplikasi tertentu. Struktur menu *pull down* seperti ditunjukkan dalam Gambar 4.12.



**Gambar 4.12** Struktur Menu *Pull Down* SIPENAS  
Sumber:[ Perancangan]

Selain itu pada antarmuka MDI terdapat toolbar sehingga lebih mempercepat akses ke dalam *form* aplikasi tertentu. Antarmuka MDI memiliki dua bagian yaitu *window* utama yang disebut dengan MDI *Form* dan *window* anak. *Window* utama akan menjadi tempat bagi *form-form* anak yang ditampilkan. Tampilan dari perancangan form utama berupa MDI *form* yang terdiri dari menu *pull down*, toolbar dan *status bar* seperti ditunjukkan dalam Gambar 4.13.

**Gambar 4.13** Rancangan *Form* Utama  
 Sumber:[ Perancangan]

**4.2.4.1 Form Login**

Untuk dapat menjalankan aplikasi maka operator harus login terlebih dahulu. Pada proses ini memerlukan user name dan password. Jika kedua data tersebut telah sesuai maka program-program dalam menu utama dapat dijalankan. Antarmuka *form* login seperti ditunjukkan dalam Gambar 4.14.

**Gambar 4.14** Rancangan *Form* Login  
 Sumber:[ Perancangan]

#### 4.2.4.2 Form Pasien

*Form* ini digunakan untuk memasukkan data pasien. Setiap pasien yang berobat terlebih dahulu harus terdaftar sehingga pasien dapat memanfaatkan layanan pengambilan nomor antrian dengan menggunakan SMS dan memiliki sistem informasi rekam medis. *Form* ini berfungsi untuk melihat data pasien meliputi nama, alamat, nomor telepon, tanggal lahir, jenis kelamin, status, agama, selain itu melalui *form* ini memungkinkan untuk dilakukan perubahan data pasien maupun penghapusan data pasien. Antarmuka *form* pasien ditunjukkan dalam Gambar 4.15.

The image shows a software interface for a patient form. The window title is "Pasien". It contains several text input fields: "Id Pasien", "Nama Pasien", "Tanggal Lahir", "Alamat", "No Telepone", "No Handphone", and "Jenis Kelamin". Below these fields is a "Data Grid" area. At the bottom, there is a row of five buttons: "Add", "Edit", "Delete", "Save", and "Exit".

**Gambar 4.15** Rancangan *Form* Pasien  
Sumber:[ Perancangan]

#### 4.2.4.3 Form Antrian

*Form* ini digunakan untuk menampilkan daftar antrian pasien sesuai dengan masing masing dokter spesialis. Setelah pasien mengirimkan SMS untuk mengambil nomor antrian maka SMS tersebut akan diproses dan disimpan dalam database dan akan

ditampilkan dalam form antrian. Antarmuka *form* antrian ditunjukkan dalam Gambar 4.16.

The image shows a window titled "Antrian" with standard window controls (minimize, maximize, close). The form contains several input fields and data grids:

- A "No Transaksi" field with a "Data Grid" label to its right.
- An "Id Pasien" field with a "Data Grid" label to its right.
- A "Save" button and an "Exit" button.
- Three additional empty "Data Grid" labels stacked vertically.

**Gambar 4.16** Rancangan *Form* Pasien  
Sumber:[ Perancangan]

#### 4.2.4.4 *Form* Rekam Medis

*Form* ini digunakan untuk mencatat riwayat sakit pasien setelah selesai berobat. Data-data pasien hasil pemeriksaan dokter dimasukkan bersama dengan data lainnya meliputi tanggal pasien berobat, dokter spesialis, hasil pemeriksaan, jenis obat. Dengan adanya rekam medis ini maka tiap pasien dapat diketahui daftar riwayat sakitnya. Selain itu dalam *form* ini terdapat fungsi untuk melihat dan mencetak detail rekam medis pasien. Antarmuka *form* rekam medis ditunjukkan dalam Gambar 4.17.

Rekam Medis

Pencatatan Rekam Medis

Detail Rekam Medis

No Transaksi

Nama Pasien

Nama Dokter

Hasil Pemeriksaan

Obat

Data Grid

Save

Exit

**Gambar 4.17** Rancangan *Form* Rekam Medis  
Sumber:[ Perancangan]

#### 4.2.4.5 *Form* Laporan

*Form* ini digunakan untuk mencetak laporan yang akan diberikan kepada pihak manajemen dan dokter spesialis. Laporan terdiri dari harian dan laporan periodik sehingga memudahkan dalam manajemen. Laporan harian adalah laporan yang diberikan kepada masing-masing dokter spesialis, sehingga dokter spesialis dapat mengetahui berapa banyak pasien yang ditangani dalam hari tersebut. . Antarmuka *form* laporan ditunjukkan dalam Gambar 4.18.

|                        |      |              |   |   |
|------------------------|------|--------------|---|---|
| Cetak Laporan          |      | -            | □ | X |
| Laporan Harian         |      |              |   |   |
| Laporan Periodik ..... |      | Sampai ..... |   |   |
| Klinik                 |      |              |   |   |
| Cetak                  | Exit |              |   |   |

**Gambar 4.18** Rancangan *Form* Laporan  
Sumber:[ Perancangan]



## BAB V

### IMPLEMENTASI

Implementasi proses terdiri dari implementasi proses login, proses pasien, proses SMS, proses rekam medis dan proses laporan. Penyajian yang digunakan berupa algoritma yang berbentuk tulisan yaitu menggunakan *pseudocode* yaitu membuat algoritma dengan pembuatan kode yang mirip dengan kode pemrograman yang sesungguhnya

#### 5.1 Proses Login

Proses login dimulai dengan memberikan masukan nilai variabel *user name*, *password*. Sistem akan melakukan beberapa tahap pengecekan masukan login. Melakukan seleksi *database* untuk pengecekan yang sesuai dengan *user name* dan *password*. Jika semua kondisi terpenuhi dan hasilnya valid maka ditampilkan form utama. Algoritma untuk memasukkan data pasien seperti ditunjukkan dalam 5.1

```
1|  Prosedur Login
2|  TYPE user name, password IS STRING
3|  INPUT user name value, password value
4|  IF (user name value empty)THEN
5|      PRINT"Masukkan user name"
6|  ELSE
7|      IF (password value empty) THEN
8|          PRINT"Masukkan password anda"
9|      ELSE
10|         Get data from data store
11|         IF (selection found) THEN
12|             PRINT"Login OK"
13|         ELSE
14|             PRINT"Login salah"
15|         ENDIF
16|     ENDIF
17| ENDIF
18| END PROC
```

**Gambar 5.1** Algoritma Proses Login  
Sumber: [Implementasi]

## 5.2 Proses Pasien

Pada proses pasien terdapat fungsi untuk menejemen data secara keseluruhan meliputi proses input data pasien, edit dan delete. Proses-proses seperti input data, edit dan delete data pasien akan dijelaskan sesuai dengan algoritma programnya.

### 5.2.1 Proses Simpan Data Pasien

Proses input data pasien dimulai dari masukan nama pasien sampai dengan jenis kelamin. Data dimasukkan satu persatu, kemudian setelah selesai program akan menyimpannya ke dalam tabel pasien dengan menggunakan perintah SQL Algoritma untuk memasukkan data pasien seperti ditunjukkan dalam 5.2

```
1| Prosedur simpan data Pasien
2| TYPE nama pasien, alamat pasien, jenis kelamin IS STRING
3| TYPE tanggal lahir IS DATE
4| TYPE no handphone, no telepon IS NUMBER
5| INPUT data nama pasien value, alamat pasien value, tanggal
6| lahir value,
7| no handphone value, no telepon value, jenis kelamin value
8| Simpan data to data store
9| END PROC
```

**Gambar 5.2** Algoritma Proses Simpan Data Pasien  
Sumber: [Implementasi]

### 5.2.2 Proses Edit Data Pasien

Proses lihat dan edit data pasien dimulai dengan seleksi pada tabel pasien untuk menampilkan data pasien yang sudah tersimpan dalam data store. Bila ada perubahan data maka ketika proses simpan akan dilakukan proses update kedalam tabel pasien. Algoritma proses melihat dan mengedit data pasien dapat ditunjukkan pada Gambar 5.3.

```
1| Prosedur edit data Pasien
2| TYPE nama pasien, alamat pasien, jenis kelamin IS STRING
3| TYPE tanggal lahir IS DATE
4| TYPE id pasien, no handphone, no telepon IS NUMBER
5| INPUT id pasien
6| IF (id pasien value empty) THEN
7|     PRINT "Anda belum memasukkan kode pasien"
8| ELSE
9|     Get data from data store
10|    IF (selection not found )THEN
11|        PRINT "data pasien tidak ada"
12|    ELSE
13|        PRINT selection value
14|        IF (data update found) THEN
15|            Update data store
16|        ELSE
17|            PRINT selection value
18|        ENDIF
19|    ENDIF
20| ENDIF
21| END PROC
```

**Gambar 5.3** Algoritma Proses Edit Data Pasien  
Sumber: [Implementasi]

### 5.2.3 Proses Delete Data Pasien

Proses hapus data pasien dimulai dengan seleksi pada tabel pasien untuk menampilkan data pasien yang sudah tersimpan dan disertai dengan pertanyaan untuk meyakinkan data akan benar-benar dihapus. Bila sudah yakin data akan dihapus maka proses penghapusan data pasien dalam tabel pasien akan dilakukan. Algoritma untuk menghapus data pasien dapat ditunjukkan pada Gambar 5.4.

```
1| Prosedur delete data Pasien
2| TYPE nama pasien, alamat pasien, jenis kelamin IS STRING
3| TYPE tanggal lahir IS DATE
4| TYPE id pasien, no handphone, no telepon IS NUMBER
5| INPUT id pasien
6| IF (id pasien value empty) THEN
7|     PRINT "Anda belum memasukkan id pasien"
8| ELSE
9|     Get data from data store
10|    IF (selection not found )THEN
11|        PRINT "data pasien tidak ada"
12|    ELSE
13|        PRINT selection value
14|        PRINT "Data yakin akan dihapus"
14|        IF (data delete yes) THEN
15|            Delete data from data store
16|        ELSE
17|            PRINT selection value
18|        ENDIF
18|    ENDIF
19| ENDIF
20| END PROC
```

**Gambar 5.4** Algoritma Proses Delete Data Pasien  
Sumber: [Implementasi]

### 5.3 Proses Rekam Medis

Proses rekam medis mempunyai dua proses utama yaitu proses untuk memasukkan data hasil pemeriksaan dokter dan proses untuk melihat dan mencetak data medis yang berisi riwayat sakit pasien. Kedua proses tersebut akan dijelaskan sesuai dengan algoritma masing-masing

#### 5.3.1 Proses Simpan Rekam Medis

Proses simpan rekam medis digunakan untuk memasukkan data hasil pemeriksaan dokter kedalam tabel berobat. Proses ini diawali dengan memasukkan data no transaksi sesuai dengan data yang tampil pada grid Masukan berupa no transaksi akan dilakukan

proses seleksi dalam tabel yang merupakan hasil relasi dua tabel yaitu tabel pasien dan antrian. Setelah semua data dimasukkan satu persatu data kemudian disimpan dalam tabel rekam medis. Algoritma untuk menyimpan rekam medis ditunjukkan dalam Gambar 5.5.

```
1| Prosedur simpan data Medis
2| TYPE nama pasien, nama dokter, hasil pemeriksaan, obat
3| IS STRING
4| TYPE nomor transaksi IS NUMBER
5| INPUT no transaksi value
6| IF (no transaksi value empty) THEN
7|     PRINT"anda belum memasukkan no transaksi"
8| ELSE
9|     Get data from data store
10|     IF (selection not found )THEN
11|         PRINT "Data tersebut tidak ada"
12|     ELSE
13|         Save to data store
14|     ENDIF
15| ENDIF
```

**Gambar 5.5** Algoritma Proses Simpan Rekam Medis  
Sumber: [Implementasi]

### 5.3.2 Proses Cetak Rekam Medis

Proses Cetak rekam medis digunakan untuk menampilkan data riwayat sakit pasien. Proses ini diawali dengan memasukkan data id pasien sesuai dengan data pasien yang akan dicetak. Masukan berupa id pasien akan dilakukan proses seleksi dalam tabel yang merupakan hasil relasi dua tabel yaitu tabel pasien dan berobat. Tampilan dari data riwayat sakit akan ditampilkan sebelum dicetak. Data riwayat sakit akan diberikan kepada dokter sebagai bagian untuk analisa penyakit pasien. Algoritma untuk mencetak rekam medis ditunjukkan dalam Gambar 5.6.

```
1|  Prosedur cetak data Medis
2|  TYPE id pasien IS NUMBER
3|  INPUT id pasien value
4|  IF (id pasien value empty) THEN
5|      PRINT"anda belum memasukkan id pasien"
6|  ELSE
7|      Get data from data store
8|      IF (selection not found )THEN
9|          PRINT "Data tersebut tidak ada"
10|     ELSE
11|         PRINT selection value
12|     ENDIF
13| ENDIF
14| END PROC
```

**Gambar 5.6** Algoritma Proses Cetak Rekam Medis  
Sumber: [Implementasi]

## 5.4 Proses SMS

Proses SMS memiliki tiga proses utama yaitu koneksi handphone dengan komputer, proses membaca inbox dan proses validasi SMS untuk selanjutnya dikirim pesan balasan kepada pasien. Ketiga proses tersebut akan dijelaskan Algoritmanya masing-masing

### 5.4.1 Proses Koneksi Handphone

Setelah komponen Active X untuk mengolah SMS telah terpasang, koneksi, fungsi terima dan kirim SMS dapat dilakukan dengan menggunakan bahasa pemrograman visual basic. Algoritma koneksi antara handphone dan komputer seperti ditunjukkan dalam Gambar 5.7.

```
1| Prosedur koneksi
2| TYPE portcom IS STRING
3| TYPE inboxSMS IS MFBUS15.FBSmsBox
4| INPUT MFBUS15Controll.Connect & portcom
5| IF (MFBUS15Controll.Connect) THEN
6|   PRINT "Koneksi OK"
7| ELSE
8|   PRINT " Koneksi gagal"
9| ENDIF
10| END PROC
```

**Gambar 5.7** Algoritma Proses Koneksi Handphone  
Sumber: [Implementasi]

#### 5.4.2 Proses Membaca Inbox

Proses membaca inbox digunakan untuk membaca setiap SMS yang dikirimkan oleh pasien. SMS yang ada di handphone SMS gateway akan di ambil kemudian dibaca untuk selanjutnya dilakukan validasi. Setelah selesai dibaca SMS langsung dihapus untuk menghindari inbox handphone penuh. Algoritma untuk membaca inbox seperti ditunjukkan dalam Gambar 5.8.

```
1| Prosedur membaca inbox
2| TYPE i,jumlah,vo,no,pengirim IS NUMBER
3| TYPE inboxSMS IS MFBUS15.FBSmsBox
4| Jumlah = Count Inbox
5| FOR i= 1 to jumlah
6|   No Pengirim =. Sender
7|   Isi SMS     =. Text
8|   Terima SMS =. Date
9|   Delete SMS
10| Next
10| END PROC
```

**Gambar 5.8** Algoritma Proses Membaca Inbox  
Sumber: [Implementasi]

### 5.4.3 Proses Validasi SMS

Proses validasi SMS digunakan untuk memvalidasi setiap SMS yang telah dibaca oleh proses membaca inbox. Ada dua parameter utama yaitu nomor pengirim dan isi SMS yang akan dilakukan validasi dengan melakukan fungsi pencarian dalam data store. Algoritma untuk memvalidasi SMS seperti ditunjukkan dalam Gambar 5.9.

```
1| Prosedur validasi SMS
2| TYPE i,jumlah,vo,no,pengirim IS NUMBER
3| TYPE inboxSMS IS MFBUS15.FBSmsBox
4| TYPE cari1,cari2,cari3,salah1,salah2,salah3 IS STRING
5| Jumlah = Count Inbox
6| FOR i= 1 to jumlah
7| INPUT SMS value
8| Get data from data store pasien
9| IF (selection not found) THEN
10| KIRIM "anda tidak terdaftar"
11| ELSE
12| Get data from data store antrian
14| IF (selection not found) THEN
15| Get data from data store dokter
16| IF (selection not found) THEN
17| KIRIM "Format SMS Salah"
18| ELSE
19| WAKTU
20| Save to data store
23| KIRIM "Nomor antrian anda : datang jam : "
24| ENDIF
25| ELSE
26| KIRIM"Anda sudah masuk dalam antrian"
27| ENDIF
28| ENDIF
29| NEXT
30| END PROC
```

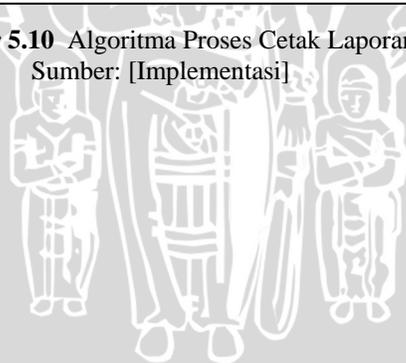
**Gambar 5.9** Algoritma Proses Validasi SMS  
Sumber: [Implementasi]

### 5.5 Proses Mencetak Laporan

Proses cetak laporan digunakan untuk mencetak laporan yang akan diberikan kepada masing-masing dokter. Dari laporan ini dokter dapat mengetahui berapa banyak jumlah pasien yang ditangani olehnya dalam satu hari. Selain itu dokter dapat juga meminta laporan dalam rentang waktu tertentu . Tampilan dari laporan akan ditampilkan sebelum dicetak . Algoritma untuk mencetak laporan ditunjukkan dalam Gambar 5.10.

```
1| Prosedur cetak Laporan
2| TYPE tanggal IS DATE
3| TYPE klinik IS STRING
4| INPUT tanggal value, klinik value
5| IF option laporan harian value= TRUE THEN
6|   Get data from data store value date now and klinik
7|   PRINT selection value
8| ELSE
9|   Get data from data store value date awal, akhir and klinik
10|  PRINT selection value
11| ENDIF
12| END PROC
```

**Gambar 5.10** Algoritma Proses Cetak Laporan  
Sumber: [Implementasi]



## BAB VI

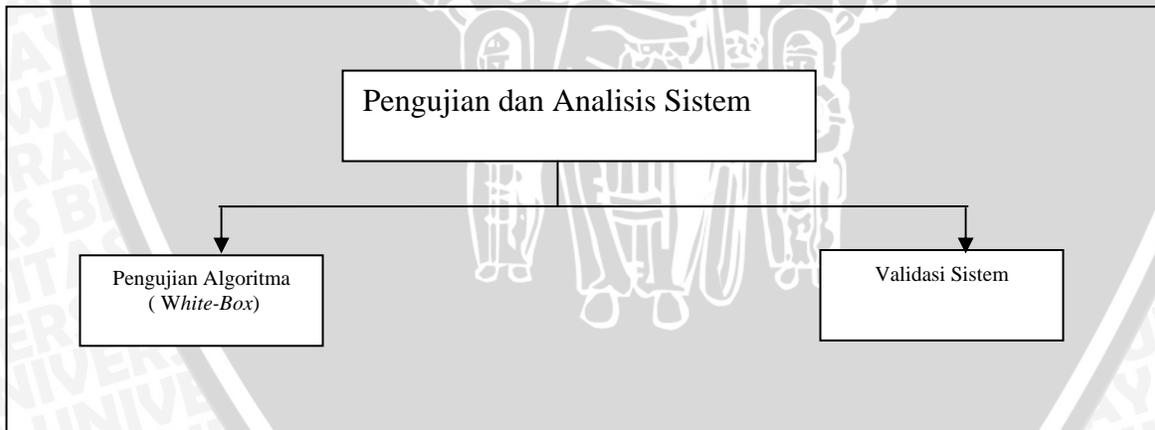
### PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai pengujian dan analisis terhadap implementasi Sistem pengambilan nomor antrian dengan menggunakan SMS. Pengujian algoritma dengan menggunakan teknik *white-box testing*, dan validasi sistem.

Pengujian *white-box* bertujuan untuk melakukan pengujian struktur jalur logika untuk menjamin tidak ada kesalahan dalam perangkat lunak yang dibuat. Pengujian ini meliputi teknik pengujian jalur dasar dan pengujian struktur kontrol. Validasi sistem dengan melakukan pengujian langsung bertujuan untuk mengetahui validitas dari perangkat lunak yang dibuat.

Dari pengujian yang telah dilakukan akan dilakukan analisis tentang perangkat lunak yang telah dibuat. Analisis yang dibuat mencakup pengujian dengan *white-box* dan kinerja perangkat lunak ketika diimplementasikan pada kondisi yang sama dengan kondisi sesungguhnya.

Diagram pohon pengujian dan analisis perangkat lunak ditunjukkan dalam Gambar 6.1



**Gambar 6.1** Diagram Pohon Pengujian Dan Analisis Perangkat Lunak  
Sumber:[Pengujian]

## 5.1 Pengujian Algoritma dengan *White-Box*

Teknik pengujian yang dilakukan adalah teknik pengujian kotak putih (*white box testing*) yaitu melakukan pengujian jalur dasar (*basis path testing*). Pada pengujian jalur dasar akan dilakukan pembuatan *programme design language* (PDL), pembuatan grafik alir, menentukan kompleksitas siklomatis, menentukan jalur independen dan melakukan kasus uji.

### 5.1.1 Pengujian Proses Login

- **Pengujian Jalur Dasar (*basis path testing*)**
  - a. Pembuatan PDL untuk desain *test case* dengan simpul – simpul yang diidentifikasi.

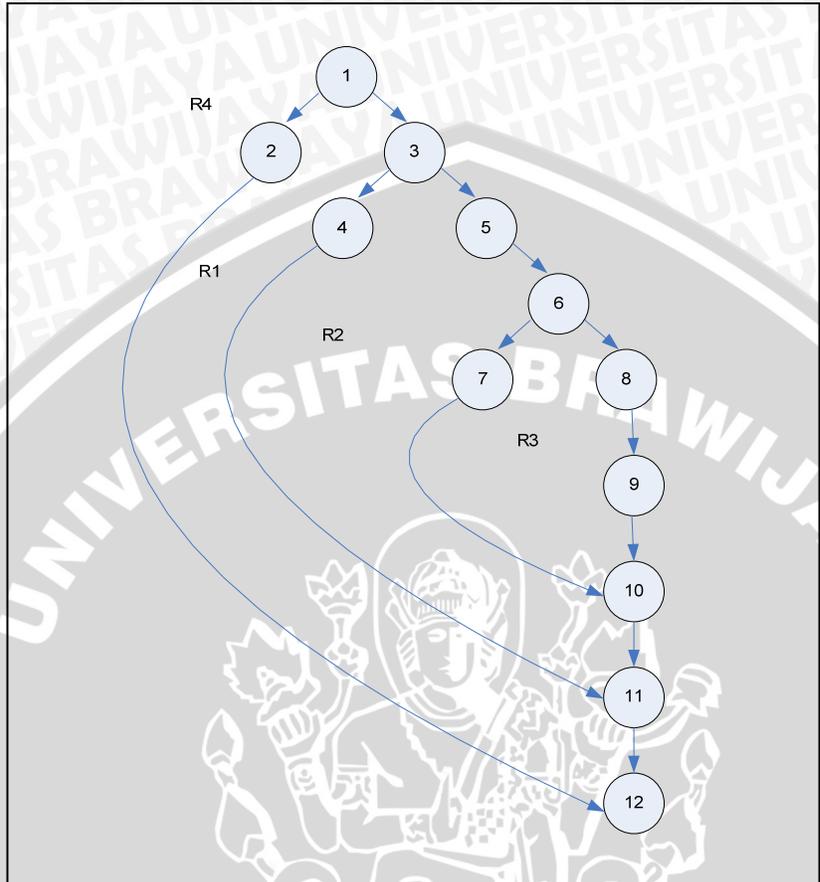
```

1| Prosedur Login
2| TYPE user name, password IS STRING
3| INPUT user name value, password value } → 1
4| IF (user name value empty) THEN
5|     PRINT"Masukkan user name" → 2
6| ELSE
7|     IF (password value empty) THEN } → 3
8|     PRINT"Masukkan password anda" → 4
9|     ELSE → 5
10| Get data from data store → 6
11|     IF (selection found) THEN
12|     PRINT"Login OK" → 7
13|     ELSE → 8
14|     PRINT"Login salah" → 9
15|     ENDIF → 10
16|     ENDIF → 11
17| ENDIF → 12
18| END PROC

```

**Gambar 6.2** PDL Login  
Sumber : [Pengujian]

b. Pembuatan grafik alir yang sesuai.



Gambar 6.3 Grafik Aliran Dari Proses Login  
Sumber : [Pengujian]

c. Penentuan kompleksitas siklomatis dari grafik alir resultan.

$$V(G) = 4 \text{ region}$$

$$V(G) = 14 \text{ edge} - 12 \text{ node} + 2 = 4$$

$$V(G) = 3 \text{ predicate node} + 1 = 4$$

d. Penentuan jalur independen.

- Jalur 1 : 1 – 2 – 12
- Jalur 2 : 1 – 3 – 4 – 11 – 12
- Jalur 3 : 1 – 3 – 5 – 6 – 7 – 10 – 11 – 12
- Jalur 4 : 1 – 3 – 5 – 6 – 8 – 9 – 10 – 11 – 12

- e. Melakukan kasus uji dan Analisis.

**Test case jalur 1:**

Nilai(*username*) = null(*empty*)

**Hasil yang diharapkan:** operator tidak bisa mengakses dan tampilkan pesan “Masukkan Username”

**Hasil pengujian:** operator tidak bisa mengakses dan ditampilkan pesan “Masukkan Username”

**Test case jalur 2:**

Nilai(*username*) = ada(*not null*)

Nilai(*password*) = null(*empty*)

**Hasil yang diharapkan:** operator tidak bisa mengakses dan tampilkan pesan “Masukkan Password”

**Hasil pengujian:** operator tidak bisa mengakses dan ditampilkan pesan “Masukkan Password”

**Test case jalur 3:**

Nilai(*username*) = ada(*not null*)

Nilai(*password*) = ada(*not null*)

Hasil seleksi ditemukan

**Hasil yang diharapkan:** operator bisa mengakses sistem dan ditampilkan pesan “Login Ok”

**Hasil pengujian:** operator bisa mengakses sistem dan ditampilkan pesan “Login Ok”

**Test case jalur 4:**

Nilai(*username*) = ada(*not null*)

Nilai(*password*) = ada(*not null*)

Hasil seleksi tidak ditemukan

**Hasil yang diharapkan:** operator tidak bisa mengakses sistem dan ditampilkan pesan “Login salah”

**Hasil yang diharapkan:** operator tidak bisa mengakses dan tampilkan pesan “Login salah”, sebab *username* atau *password* tidak sesuai.

**Hasil pengujian:** operator tidak bisa mengakses dan ditampilkan pesan “Login salah”

f. Kesimpulan

Dari pengujian *white-box* yang telah dilakukan dapat disimpulkan bahwa proses pengecekan login tidak terdapat kesalahan terjamin validitasnya.

### 5.1.2 Pengujian Proses Edit Pasien

▪ **Pengujian Jalur Dasar (*basis path testing*)**

- a. Pembuatan PDL untuk desain *test case* dengan simpul – simpul yang diidentifikasi.

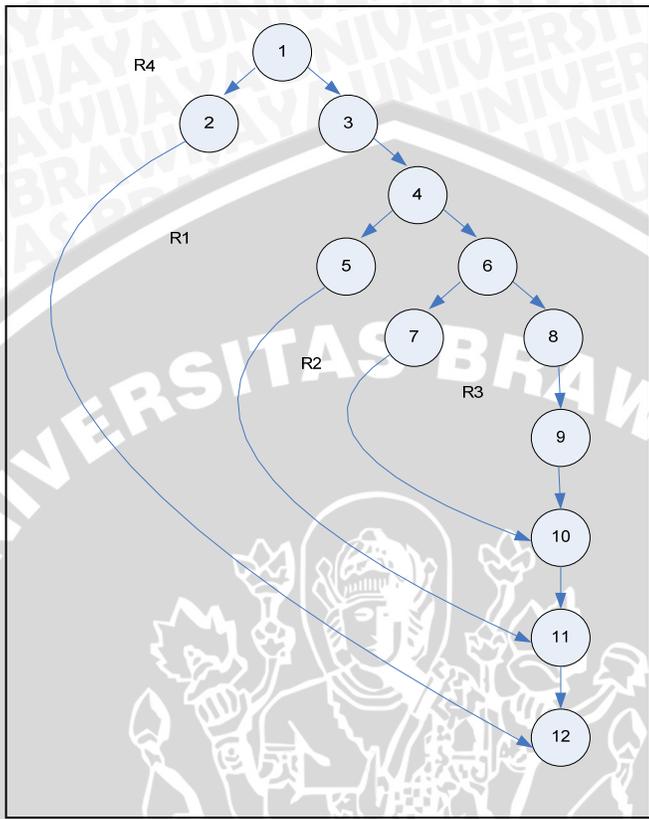
```

1| Prosedur edit Pasien
2| TYPE nama pasien, alamat pasien, jenis kelamin IS STRING
3| TYPE tanggal lahir IS DATE
4| TYPE id pasien, no handphone, no telepon IS NUMBER
5| INPUT id pasien
6| IF (id pasien value empty) THEN } → 1
7|     PRINT "Anda belum memasukkan id pasien" → 2
8| ELSE → 3
9|     Get data from data store
10|     IF (selection not found )THEN } → 4
11|         PRINT "data pasien tidak ada" → 5
12|     ELSE → 6
13|         PRINT selection value
14|         IF (data update found) THEN } → 7
15|         Update data store
16|     ELSE → 8
17|         PRINT selection value → 9
18|     ENDIF → 10
19| ENDIF → 11
20| ENDIF → 12
21| END PROC

```

**Gambar 6.4** PDL Lihat Dan Edit Data Pasien  
Sumber : [Pengujian]

b. Pembuatan grafik alir yang sesuai.



**Gambar 6.5** Grafik Aliran Dari Prosedur Lihat Dan Edit Data Pasien  
Sumber :[ Pengujian]

c. Penentuan kompleksitas siklomatis dari grafik alir resultan.

$$V(G) = 4 \text{ region}$$

$$V(G) = 14 \text{ edge} - 12 \text{ node} + 2 = 4$$

$$V(G) = 3 \text{ predicate node} + 1 = 4$$

d. Penentuan jalur independen.

- Jalur 1 : 1 – 2 – 12
- Jalur 2 : 1 – 3 – 4 – 5 – 10 – 11 – 12
- Jalur 3 : 1 – 3 – 4 – 6 – 7 – 10 – 11 – 12
- Jalur 4 : 1 – 3 – 4 – 6 – 8 – 9 – 10 – 11 – 12

- e. Melakukan kasus uji dan Analisis.

**Test case jalur 1:**

Nilai(*id pasien*) = null(*empty*)

**Hasil yang diharapkan:** operator tidak mendapatkan tampilan nama pasien, alamat, tanggal lahir, nomor handphone, nomor telepon, jenis kelamin dan tampilan pesan “Anda belum memasukkan id pasien”

**Hasil pengujian:** operator tidak mendapatkan tampilan nama pasien, alamat, tanggal lahir, nomor handphone, nomor telepon, jenis kelamin dan tampilan pesan “Anda belum memasukkan id pasien”

**Test case jalur 2:**

Nilai(*id pasien*) = ada(*not null*)

Hasil seleksi tidak ditemukan

**Hasil yang diharapkan:** operator tidak mendapatkan tampilan nama pasien, alamat, tanggal lahir, nomor handphone, nomor telepon, jenis kelamin dan ditampilkan pesan “Data pasien tidak ada”

**Hasil pengujian:** operator tidak mendapatkan tampilan nama pasien, alamat, tanggal lahir, nomor handphone, nomor telepon, jenis kelamin dan ditampilkan pesan “Data pasien tidak ada”

**Test case jalur 3:**

Nilai(*id pasien*) = ada(*not null*)

Hasil seleksi ditemukan dan ada data yang diubah.

**Hasil yang diharapkan:** operator mendapatkan tampilan nama pasien, alamat, tanggal lahir, nomor handphone, nomor telepon, jenis kelamin dan dapat mengubah data kemudian menyimpannya.

**Hasil pengujian** operator mendapatkan tampilan nama pasien, alamat, tanggal lahir, nomor handphone, nomor telepon, jenis kelamin dan dapat mengubah data kemudian menyimpannya.

**Test case jalur 4:**

Nilai(*id pasien*) = ada(*not null*)

Hasil seleksi ditemukan tetapi tidak ada data yg diubah oleh operator

**Hasil yang diharapkan:** operator hanya mendapatkan tampilan nama pasien, alamat, tanggal lahir, nomor handphone, nomor telepone, jenis kelamin

**Hasil pengujian:** operator hanya mendapatkan tampilan nama pasien, alamat, tanggal lahir, nomor handphone, nomor telepone, jenis kelamin.

f. Kesimpulan

Dari pengujian *white-box* yang telah dilakukan dapat disimpulkan bahwa proses lihat dan edit data pasien tidak terdapat kesalahan terjamin validitasnya.

### 5.1.3 Pengujian Proses Cetak Rekam Medis

- **Pengujian Jalur Dasar (*basis path testing*)**

- a. Pembuatan PDL untuk desain *test case* dengan simpul – simpul yang diidentifikasi.

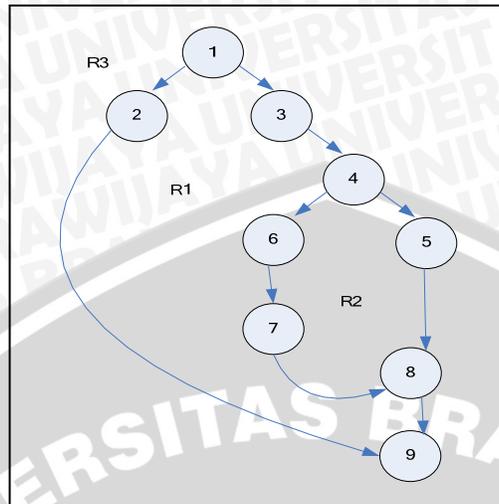
```

1| Prosedur cetak Rekam Medis
2| TYPE id pasien IS NUMBER
3| INPUT id pasien value           } → 1
4| IF (id pasien value empty) THEN }
5|     PRINT"anda belum memasukkan id pasien" → 2
6| ELSE → 3
7|     Get data from data store     } → 4
8|     IF (selection not found )THEN }
9|         PRINT "Data tersebut tidak ada" → 5
10|    ELSE → 6
11|        PRINT selection value → 7
12|    ENDIF → 8
13| ENDIF → 9
14| END PROC

```

**Gambar 6.6** PDL Lihat Rekam Medis  
Sumber : [Pengujian]

b. Pembuatan grafik alir yang sesuai.



**Gambar 6.7** Grafik Aliran Dari Proses Lihat Rekam Medis  
Sumber : [Pengujian]

c. Penentuan kompleksitas siklomatis dari grafik alir resultan.

$$V(G) = 3 \text{ region}$$

$$V(G) = 10 \text{ edge} - 9 \text{ node} + 2 = 3$$

$$V(G) = 2 \text{ predicate node} + 1 = 3$$

d. Penentuan jalur independen.

Jalur 1 : 1 - 2 - 9

Jalur 2 : 1 - 3 - 4 - 5 - 8 - 9

Jalur 3 : 1 - 3 - 4 - 6 - 7 - 8 - 9

e. Melakukan kasus uji dan Analisis.

**Test case jalur 1:**

Nilai(*id pasien*) = null(*empty*)

**Hasil yang diharapkan:** operator tidak mendapatkan tampilan id pasien, nama pasien, jenis kelamin, tanggal berobat, hasil pemeriksaan, obat dan tampilan pesan “Anda belum memasukkan id pasien”

**Hasil pengujian:** operator tidak mendapatkan tampilan id pasien, nama pasien, jenis kelamin, tanggal berobat, hasil pemeriksaan, obat dan tampilan pesan “Anda belum memasukkan id pasien”

**Test case jalur 2:**

Nilai(*id pasien*) = ada(*not null*)

Hasil seleksi tidak ditemukan

**Hasil yang diharapkan:** operator tidak mendapatkan tampilan id pasien, nama pasien, jenis kelamin, tanggal berobat, hasil pemeriksaan, obat dan ditampilkan pesan “Data tersebut tidak ada”

**Hasil pengujian:** operator tidak mendapatkan tampilan id pasien, nama pasien, jenis kelamin, tanggal berobat, hasil pemeriksaan, obat dan ditampilkan pesan “Data tersebut tidak ada”

**Test case jalur 3:**

Nilai(*id pasien*) = ada(*not null*)

Hasil seleksi ditemukan

**Hasil yang diharapkan:** operator mendapatkan tampilan id pasien, nama pasien, jenis kelamin, tanggal berobat, hasil pemeriksaan, obat pasien.

**Hasil pengujian** operator mendapatkan tampilan id pasien, nama pasien, jenis kelamin, tanggal berobat, hasil pemeriksaan, obat dan siap untuk dicetak.

## f. Kesimpulan

Dari pengujian *white-box* yang telah dilakukan dapat disimpulkan bahwa proses lihat rekam medis pasien tidak terdapat kesalahan terjamin validitasnya.

**5.1.4 Pengujian Proses Cetak Laporan****▪ Pengujian Jalur Dasar (*basis path testing*)**

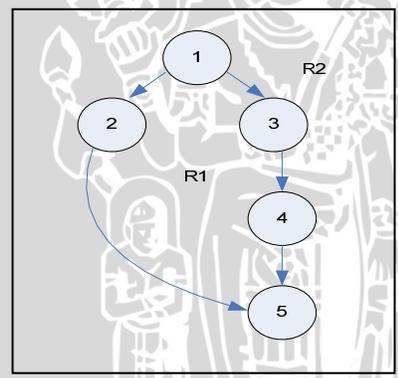
- a. Pembuatan PDL untuk desain *test case* dengan simpul – simpul yang diidentifikasi.

```

1|  Prosedur cetak Laporan
2|  TYPE tanggal IS DATE
3|  TYPE klinik IS STRING
4|  INPUT tanggal value, klinik value
5|  IF option laporan harian value= TRUE THEN
6|    Get data from data store value date now and klinik
7|    PRINT selection value
8|  ELSE
9|    Get data from data store value date awal, akhir and klinik
10|   PRINT selection value
11| ENDIF
12| END PROC
    
```

**Gambar 6.8** PDL Cetak Laporan  
 Sumber : [Pengujian]

b. Pembuatan grafik alir yang sesuai.



**Gambar 6.9** Grafik Aliran Cetak Laporan  
 Sumber : [Pengujian]

c. Penentuan kompleksitas siklomatis dari grafik alir resultan.

$$V(G) = 2 \text{ region}$$

$$V(G) = 5 \text{ edge} - 5 \text{ node} + 2 = 2$$

$$V(G) = 1 \text{ predicate node} + 1 = 2$$

d. Penentuan jalur independen.

- Jalur 1 : 1 – 2 – 5
- Jalur 2 : 1 – 3 – 4 – 5

- e. Melakukan kasus uji dan Analisis.

**Test case jalur 1:**

Nilai(*tanggal*) = ada(*not null*)

Nilai(*klirik*) = ada(*not null*)

**Hasil yang diharapkan:** operator dapat mencetak laporan sesuai dengan tanggal dan jenis klinik.

**Hasil pengujian:** operator dapat mencetak laporan sesuai dengan tanggal dan jenis klinik.

**Test case jalur 2:**

Nilai(*tanggal*) = ada(*not null*)

Nilai(*klirik*) = ada(*not null*)

**Hasil yang diharapkan:** operator dapat mencetak laporan sesuai dengan rentang tanggal dan jenis klinik.

**Hasil pengujian:** operator dapat mencetak laporan sesuai dengan rentang tanggal dan jenis klinik.

**Kesimpulan**

Dari pengujian *white-box* yang telah dilakukan dapat disimpulkan bahwa proses cetak laporan tidak terdapat kesalahan terjamin validitasnya.

## 5.2 Validasi Sistem

Pengujian pada sistem ini dilakukan untuk mengetahui apakah sistem dapat berjalan dengan baik dan telah sesuai dengan perancangan. Pengujian yang dilakukan mengambil sebagian modul dari aplikasi yang dibuat.

### 5.2.1 Pengujian Login [PRJ-010]

Pengujian yang dilakukan dengan memberikan masukan pada *login* untuk mengetahui respon sistem saat diberikan *login* yang benar dan salah.

**a. Tujuan**

- Pengujian ini dilakukan untuk mengetahui respon sistem ketika diberikan login

**b. Prosedur dan hasil pengujian**

## 1. Kasus Uji 1:

Memasukan username dan password yang benar.

## • Prosedur Uji:

1. Melakukan proses login dengan menggunakan username = user01 dan password = userawal01

## • Hasil yang diharapkan

1. Akan ditampilkan pesan “Login OK”
2. Form utama ditampilkan.

## • Hasil Pengujian

1. Ditampilkan pesan “Login OK”



**Gambar 6.10** Pesan “Login OK”  
Sumber : [Pengujian]

2. Form utama ditampilkan.



**Gambar 6.11** Tampilan Form Utama  
Sumber : [Pengujian]

## 2. Kasus Uji 2:

Memasukan username dan password yang salah.

- Prosedur Uji:

1. Melakukan proses login dengan menggunakan username = coba dan password = userawal01

- Hasil yang diharapkan

1. Akan ditampilkan pesan “Login salah”
2. Form utama tidak ditampilkan.

- Hasil Pengujian

1. Ditampilkan pesan “Login salah”



**Gambar 6.12** Pesan “Login salah”  
Sumber : [Pengujian]

### c. Analisis

Hasil pengujian login menunjukkan hasil valid. Hasil yang diharapkan sesuai dengan hasil yang didapatkan dalam pengujian.

## 5.2.2 Pengujian Input Data Pasien [PRJ-011]

Pengujian input data pasien digunakan untuk memasukkan data pasien kedalam database.

### a. Tujuan

- Pengujian ini dilakukan untuk mengetahui apakah program dapat menyimpan data pasien dalam database.

**b. Prosedur dan hasil pengujian**

1. Kasus Uji 1:

Memasukan data pasien.

- Prosedur Uji:

1. Mengaktifkan form pasien
2. Memasukkan data pasien dengan memberikan masukan

Nama pasien = Marini  
 Alamat pasien = Jl.Bantaran Blok G No 27  
 Tanggal Lahir = 05-07-1978  
 No Handphone = 08155512524  
 No Telephone = 0341415468  
 Jenis Kelamin = Wanita

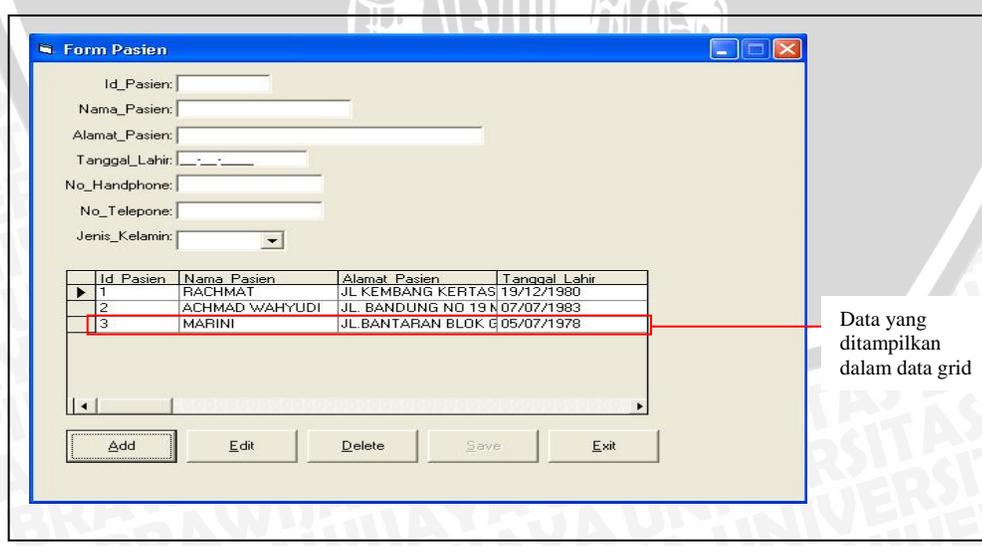
3. Memberikan perintah *query* untuk melihat data pasien yang disimpan dalam database.

- Hasil yang diharapkan

1. Data pasien yang telah dimasukkan langsung ditampilkan dalam data grid.
2. Data pasien disimpan dalam database.

- Hasil Pengujian

1. Data pasien yang dimasukkan ditampilkan dalam *data grid*.



Data yang ditampilkan dalam data grid

**Gambar 6.13** Form Pasien Setelah Menerima Masukan Data Pasien  
 Sumber : [Pengujian]

2. Tampilan data pasien dalam database setelah dilakukan *query*.

| Query1 : Select Query |             |                          |               |               |
|-----------------------|-------------|--------------------------|---------------|---------------|
|                       | Nama_Pasien | Alamat_Pasien            | Tanggal_Lahir | No_Handphone  |
| ▶                     | MARINI      | JL.BANTARAN BLOK G NO 27 | 05/07/1978    | +628155512524 |
| *                     |             |                          |               |               |

**Gambar 6.14** Hasil *Query* Dalam Tabel Pasien  
Sumber : [Pengujian]

**c. Analisis**

Hasil pengujian input data pasien menunjukkan hasil valid. Hasil yang diharapkan sesuai dengan hasil yang didapatkan dalam pengujian.

**5.2.3 Pengujian Antrian Melalui SMS [PRJ-001], [PRJ-002], [PRJ-003], [PRJ-004]**

Pengujian antrian melalui SMS digunakan untuk mengetahui respon program apakah dapat mengirimkan SMS balasan.

**a. Tujuan**

- Pengujian ini dilakukan untuk mengetahui respon program apakah dapat mengirmkan SMS balasan. Jika hasil validasi SMS valid maka akan dikirimkan nomor antrian dan jam datang jika hasil validasi tidak valid akan dikirimkan pesan kesalahan.

**b. Prosedur dan hasil pengujian**

1. Kasus Uji 1:

Mengirim SMS valid.

- Prosedur Uji:

1. Pasien yang terdaftar mengirim SMS dengan format yang telah ditentukan yakni Klinik kemudian spasi jenis klinik yang akan dituju. Jenis klinik yang ada yakni klinik mata, klinik THT, klinik anak dan klinik umum sesuai dengan yang ada pada klinik dokter praktek bersama. Contoh pengetikan SMS nya adalah Klinik Mata sedangkan

untuk huruf besar dan kecil tidak menjadi masalah. Dalam pengujian ini pasien mengirimkan SMS klinik mata.

2. Memberikan perintah *query* untuk melihat data SMS yang disimpan dalam tabel antrian.
  - Hasil yang diharapkan
    1. Akan diberikan balasan nomor antrian dan jam datang
    2. SMS pasien disimpan dalam tabel antrian.
  - Hasil Pengujian
    1. Pada handphone pasien diterima SMS dengan isi nomor antrian dan jam datang.



**Gambar 6.15** Tampilan SMS Balasan Untuk Data SMS Valid  
Sumber : [Pengujian]

2. Tampilan data antrian dalam tabel database setelah dilakukan *query*

|   | No_Antrian | Id_Pasien | Klinik      | Jam_Datang | Status | No_Handphone   |
|---|------------|-----------|-------------|------------|--------|----------------|
| ▶ | 1          | 1         | KLINIK MATA | 18:00      | 0      | +6281805000584 |

**Gambar 6.16** Hasil *Query* Dalam Tabel Antrian  
Sumber : [Pengujian]

2. Kasus Uji 2:
  - Pasien yang tidak terdaftar.
    - Prosedur Uji:
      1. Pasien yang tidak terdaftar mengirim SMS dengan format yang telah ditentukan. Format yang telah ditentukan yakni Klinik kemudian spasi

jenis klinik yang akan dituju. Jenis klinik yang ada yakni klinik mata, klinik THT, klinik anak dan klinik umum sesuai dengan yang ada pada klinik dokter praktek bersama. Contoh pengetikan SMS nya adalah Klinik Mata sedangkan untuk huruf besar dan kecil tidak menjadi masalah. Dalam pengujian ini pasien mengirimkan SMS klinik mata.

- Hasil yang diharapkan
  1. Dikirimkan SMS balasan dengan isi SMS "Anda tidak terdaftar"
- Hasil Pengujian
  1. Pada handphone pasien diterima SMS dengan isi "Anda tidak terdaftar"



**Gambar 6.17** Tampilan SMS Balasan Untuk Pasien Yang Tidak Terdaftar  
Sumber : [Pengujian]

### 3. Kasus Uji 3:

Mengirim SMS tidak valid.

- Prosedur Uji:
  1. Pasien yang terdaftar mengirim SMS dengan format yang tidak sesuai dengan yang ditentukan. Format yang telah ditentukan yakni Klinik kemudian spasi jenis klinik yang akan dituju. Jenis klinik yang ada yakni klinik mata, klinik THT, klinik anak dan klinik umum sesuai dengan yang ada pada klinik dokter praktek bersama. Contoh pengetikan SMS nya adalah Klinik Mata sedangkan untuk huruf besar dan kecil tidak menjadi masalah. Dalam pengujian ini pasien mengirimkan SMS klinik penyakit dalam.

- Hasil yang diharapkan
  1. Dikirimkan SMS balasan dengan isi SMS “Format SMS Yang Anda Ketik Salah”
- Hasil Pengujian
  1. Pada handphone pasien diterima SMS dengan isi “Format SMS Yang Anda Ketik Salah”



**Gambar 6.18** Tampilan SMS Balasan untuk Format SMS Salah  
Sumber : [Pengujian]

#### 4. Kasus Uji 4:

Pasien sudah masuk daftar antrian.

- Prosedur Uji:
  1. Pasien yang terdaftar telah mengambil nomor antrian dan sudah masuk dalam daftar antrian mengirim SMS dengan format yang ditentukan. Format yang telah ditentukan yakni Klinik kemudian spasi jenis klinik yang akan dituju. Jenis klinik yang ada yakni klinik mata, klinik THT, klinik anak dan klinik umum sesuai dengan yang ada pada klinik dokter praktek bersama. Contoh pengetikan SMS nya adalah Klinik Mata sedangkan untuk huruf besar dan kecil tidak menjadi masalah. Dalam pengujian ini pasien mengirimkan SMS klinik mata.
- Hasil yang diharapkan
  1. Dikirimkan SMS balasan dengan isi SMS “Anda sudah masuk dalam antrian no: datang jam :”
- Hasil Pengujian
  1. Pada handphone pasien diterima SMS dengan isi “Anda sudah masuk dalam antrian no: datang jam :”



**Gambar 6.19** Tampilan SMS Balasan Untuk Pasien Sudah Masuk Antrian  
Sumber : [Pengujian]

### c. Analisis

Hasil pengujian antrian melalui SMS menunjukkan hasil valid. Hasil yang diharapkan sesuai dengan hasil yang didapatkan dalam pengujian.

#### 5.2.4 Pengujian Cetak Rekam Medis [PRJ-007]

Pengujian cetak rekam medis digunakan untuk mengetahui apakah program dapat mencetak detail rekam medis pasien sesuai dengan yang diinginkan.

##### a. Tujuan

- Pengujian ini dilakukan untuk mengetahui apakah program dapat mencetak detail rekam medis pasien meliputi id pasien, nama pasien, jenis kelamin, tanggal berobat, hasil pemeriksaan dan jenis obat.

##### b. Prosedur dan hasil pengujian

###### 1. Kasus Uji 1:

Menampilkan detail rekam medis.

- Prosedur Uji:
  1. Mengaktifkan *form* rekam medis.
  2. Memberikan masukan berupa id pasien = 1 untuk menampilkan detail rekam medis pasien.
- Hasil yang diharapkan
  1. Ditampilkan detail rekam medis pasien.
- Hasil Pengujian
  1. Data report menampilkan detail rekam medis pasien.

**DataReport1**  
Zoom 100%

**Riwayat Sakit Pasien** **Klinik Dokter Spesialis Praktek**  
Jl.Panglima Sudirman No 68 Gresik

---

**Id** 1  
**Nama** RACHMAT  
**Jenis Kelamin:** PRIA

---

| Tanggal Berobat: | Hasil Pemeriksaan:  | Obat:                                  |
|------------------|---|--|
| 6/12/2007        | Mata baik-baik saja. Agak lelah,terlalu banyak tidur malam. | gak usah pake obat,tidur yang cukup ya |

**Gambar 6.20** Tampilan Detail Rekam Medis Pasien  
Sumber : [Pengujian]

## 2. Kasus Uji 2:

Mencetak detail rekam medis ke printer.

- **Prosedur Uji:**

1. Mengaktifkan *form* rekam medis.
2. Memberikan masukan berupa id pasien = 1 untuk menampilkan detail rekam medis pasien.
3. Mencetak detail rekam medis yang ditampilkan oleh data report dengan melakukan klik pada tombol print.

- **Hasil yang diharapkan**

1. Detail rekam medis dicetak sesuai dengan yang ditampilkan oleh data report.

- **Hasil Pengujian**

1. Detail rekam medis yang dicetak sama dengan detail rekam medis yang ditampilkan oleh data report.

**c. Analisis**

Hasil pengujian cetak rekam medis menunjukkan hasil valid. Hasil yang diharapkan sesuai dengan hasil yang didapatkan dalam pengujian.

**5.2.5 Pengujian Cetak Laporan [PRJ-008]**

Pengujian cetak laporan digunakan untuk mengetahui apakah program dapat mencetak laporan sesuai dengan yang diinginkan.

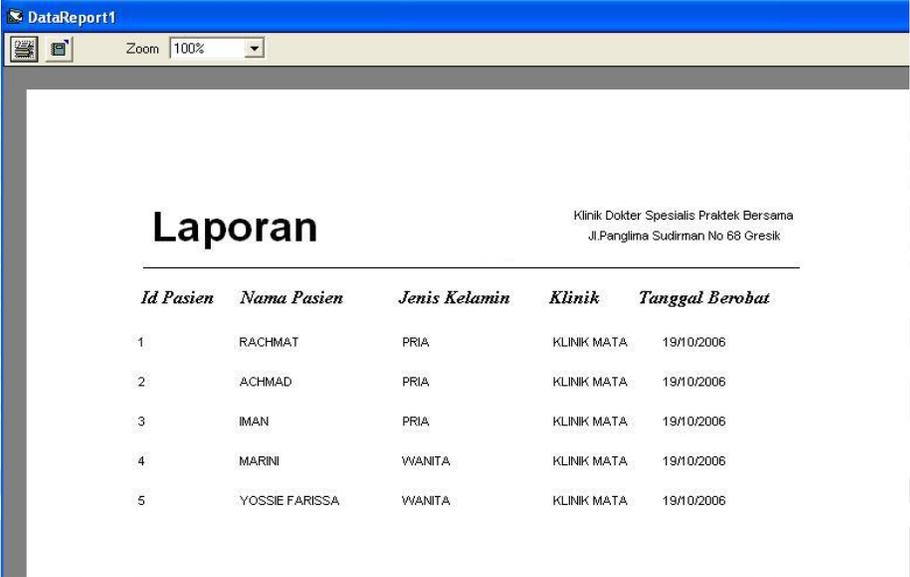
**a. Tujuan**

- Pengujian ini dilakukan untuk mengetahui apakah program dapat mencetak laporan sesuai dengan masukan tanggal dan jenis klinik

**b. Prosedur dan hasil pengujian****1. Kasus Uji 1:**

Menampilkan laporan.

- Prosedur Uji:
  1. Mengaktifkan *form* laporan.
  2. Memberikan masukan berupa tanggal = 19/10/2006 dan klinik = Klinik mata
- Hasil yang diharapkan
  1. Ditampilkan laporan sesuai tanggal dan jenis klinik.
- Hasil Pengujian
  1. Data report menampilkan laporan sesuai masukan tanggal dan jenis klinik.



| <i>Id Pasien</i> | <i>Nama Pasien</i> | <i>Jenis Kelamin</i> | <i>Klinik</i> | <i>Tanggal Berobat</i> |
|------------------|--------------------|----------------------|---------------|------------------------|
| 1                | RACHMAT            | PRIA                 | KLINIK MATA   | 19/10/2006             |
| 2                | ACHMAD             | PRIA                 | KLINIK MATA   | 19/10/2006             |
| 3                | IMAN               | PRIA                 | KLINIK MATA   | 19/10/2006             |
| 4                | MARINI             | WANTA                | KLINIK MATA   | 19/10/2006             |
| 5                | YOSSIE FARISSA     | WANTA                | KLINIK MATA   | 19/10/2006             |

**Gambar 6.21** Tampilan Laporan  
Sumber : [Pengujian]

2. Kasus Uji 2:

Mencetak laporan ke printer.

- Prosedur Uji:
  1. Mengaktifkan *form* laporan.
  2. Memberikan masukan berupa tanggal = 19/10/2006 dan klinik = Klinik mata
  3. Mencetak laporan yang ditampilkan oleh data report dengan melakukan klik pada tombol print.
- Hasil yang diharapkan
  1. Laporan dicetak sesuai dengan yang ditampilkan oleh data report.
- Hasil Pengujian
  1. Laporan yang dicetak sama dengan laporan yang ditampilkan oleh data report.

**c. Analisis**

Hasil pengujian cetak laporan menunjukkan hasil valid. Hasil yang diharapkan sesuai dengan hasil yang didapatkan dalam pengujian.

## BAB VII PENUTUP

### 7.1 Kesimpulan

Dari hasil perancangan dan pengujian sistem pengambilan nomor antrian dengan menggunakan SMS dan aplikasi rekam medis, dapat disimpulkan bahwa:

1. Dari pengujian algoritma menggunakan *white-box testing*, sistem yang dibuat tidak menemukan adanya kesalahan struktur jalur logika dan terjamin validitasnya.
2. Pada pengujian sistem secara nyata, proses pengambilan nomor antrian dengan menggunakan SMS dapat mengirimkan balasan SMS secara otomatis berupa informasi nomor antrian dan jam datang.

### 7.2 Saran

Saran yang dapat diberikan untuk pengembangan aplikasi ini antara lain:

1. Pada ruang tunggu pasien dapat dipasang display berupa monitor atau rangkaian dari beberapa led yang menunjukkan nomor antrian.
2. Dengan menggunakan jaringan komputer, Sistem rekam medis pasien dapat diakses langsung oleh dokter dalam ruangan klinik dengan menggunakan komputer.

**DAFTAR PUSTAKA**

- [SOF-06] Anonymous, 2006, "Compatibility", <http://www.Softwarecave.nl/> , Diakses terakhir 19 Mei 2006
- [PIN-06] Anonymous, 2006, "Nokia 3315 Pin out", <http://www.Pinout.ru/> , Diakses terakhir 19 Mei 2006
- [ADI-06] Anonymous,..2006,.."SQL.For.Access"..[http://www.adit.co.uk/html/SQL\\_for\\_Access.html](http://www.adit.co.uk/html/SQL_for_Access.html) , Diakses terakhir 19 Mei 2006
- [BRA-01] Bradley, Julia., 2001, "Advanced Programming Using Visual Basic", Mcgraw Hill
- [FAT-02] Fathansyah, Ir., 2002, "Basis Data". Bandung: Informatika
- [HAD-04] Hadi, Rahadian., 2004, "Membuat Laporan Dengan Crystal Reports 8.5 Dan Visual Basic 6.0", Elex Media
- [IBR-98] Ibrahim, KF., 1998, "Teknik Digital", Andi Yogyakarta
- [IMR-04] Imron Rozidi, Romzi., 2004, "Membuat Sendiri SMS Gateway" , Andi Yogyakarta
- [KHA-03] Khang, Bustam. Ir., 2003, "Trik Pemrograman Aplikasi Berbasis SMS", Elex Media
- [MAN-04] Mangkulo, Hengky Alexander., 2004, "Pemrograman Visual Basic 6.0 Dan Microsoft Access", Elex Media
- [MUN-03] Munir, Rinaldi., 2005, "Matematika Diskrit", Bandung: Informatika
- [PEA-05] Peacock, Wayne., 2005, "A Introduction To Nokia F-Bus", <http://www.embedtronics.com/nokia/fbus.html> , Diakses terakhir 19 Mei 2006
- [POH-97] Pohan, Husni Iskandar., 1997, "Pengantar Perancangan Sistem" Jakarta: Erlangga
- [PRE-02] Pressman, Roger.S. Phd., 2002, "Rekayasa Perangkat Lunak", Buku Satu, Penerbit Andi
- [SUR-03] Suryo Kusumo, Ario. Drs., 2003, "Pemrograman Database Dengan Visual Basic 6.0", Elex Media

- [WHI-03] White Horn, Mark., 2003, "Seluk Beluk Database Relational" , Erlangga
- [WIN-06] Winarko, Edi., 2006, "Perancangan Database Dengan Power Designer 6.32", Prestasi Pustaka

