

SISTEM INFORMASI FARMASI
DI RSU PKU MUHAMMADIYAH YOGYAKARTA

7/11/2001
10 Nop 01
FTEK-INDUSRI-VI
ARsip
11/01/2001



Oleh :

WINARSIH

No. Mhs : 95 351 100

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2001

**SISTEM INFORMASI FARMASI DI RSUL PKU
MUHAMMADIYAH YOGYAKARTA**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika



Oleh:

Nama : WINARSIH
No. Mahasiswa : 95 351 100
NIRM : 95005101311312099

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

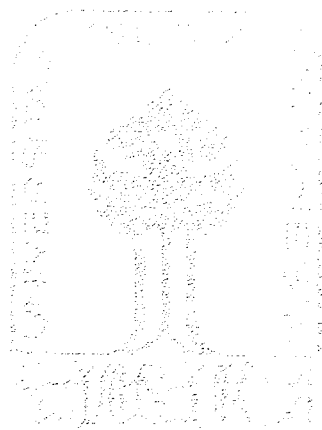
2001



LEMBAR PENGESAHAN PEMBIMBING

SISTEM INFORMASI FARMASI DI RSU-PKU MUHAMMADIYAH YOGYAKARTA

TUGAS AKHIR



Oleh :

Nama : WINARSIH
No. Mahasiswa : 95 351 100
NIRM : 95005101311312099

Yogyakarta, Agustus 2001

Pembimbing

(Dr. Sri Hartati, MSc)

LEMBAR PENGESAHAN PENGUJI

SISTEM INFORMASI FARMASI DI RSU-PKU
MUHAMMADIYAH YOGYAKARTA

TUGAS AKHIR

Oleh:

Nama : Winarsih
No. Maasiswa : 95 351 100
NIRM : 95005101311312099

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

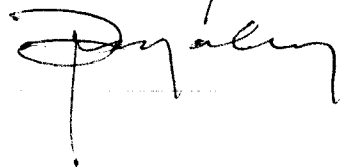
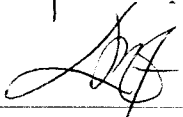
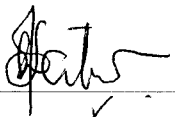
Yogyakarta, Agustus 2001

Tim Penguji

DR. Sri Hartati, MSc
ketua

Ami Fauziah, ST
Anggota I

Teduh Dirgahayu, ST
Anggota II



Mengetahui,
Fakultas Teknologi Industri
Universitas Islam Indonesia

Bachrun Sutrisno, M.Sc

MOTTO

Hanya penderitaan hidup yang dapat mengajarkan kepada manusia bagaimana menghargai kebaikan dan keindahan hidup

Barang siapa yang menyembunyikan dengki dengan jalan berbohong dan siapa yang memasyurkan kabar buruk, Ia itu bodoh dan jahat adanya

Cinta yang ditanam pada kewajiban yang luhur akan terus tumbuh walaupun kejelitaan telah berganti uban dan ketampanan menjadi renta

Jangan katakan semua yang diketahui tetapi ketahuilah yang semua dikatakan

HALAMAN PERSEMBAHAN

Karya kecil ini kupersembahkan untuk...

- œ Bapak dan ibu, terimakasih atas doanya yang tulus, kasih sayang dan bantuan materi yang telah diberikan selama ini*
- œ Suamiku Mas Januariusdi, ST dan buah hatiku Alif Ananda Wijapena Prayogi yang telah memberikan cinta dan perhatiannya*
- œ Kakakku Mas Dono dan adikku Sri, terimakasih atas segala bantuan dan pengorbanan waktunya*

UCAPAN TERIMAKASIH

Special thank's to Didi Cs atas semua bantuan dan waktunya

Rindang, thank's udah nganterin aku demo

Linda, thank's atas kritikan-kritikannya dan akhirnya kita lulus bareng

Maria, kamu teman yang baik, kapan nyusul wisuda

Pipih, thanks infonya

Lia, Sarie, Yoni, Wahyu, Mba' Ita, Mba' Maya thank's telah menjadi

teman yang baik

Semua teman-teman informatika '95

KATA PENGANTAR

Bismillahirrohmanirrohim

Assalamu'alaikum Wr. Wb.

Puja dan puji syukur kehadirat Allah SWT yang telah melimpahkan Rahmat dan Hidayah-Nya sehingga penyusun dapat menyelesaikan laporan Tugas Akhir yang berjudul :

SISTEM INFORMASI FARMASI DI RSUD MUHAMMADIYAH
YOGYAKARTA.

Laporan Tugas Akhir ini dimaksudkan guna memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.

Pada kesempatan ini penyusun mengucapkan terimakasih kepada :

1. Rektor Universitas Islam Indonesia, Bapak Prof. DR. H. Zaini Dahlan, MA
2. Dekan Fakultas Teknologi Industri, Bapak Ir. H. Bachrun Sutrisno
3. Ketua Jurusan Teknik Informatika, Ibu Sri Kusumadewi, MMT
4. Dosen pembimbing, Ibu DR. Sri Hartati, Msc
5. Seluruh staff bagian farmasi RSUD Muhammadiyah Yogyakarta
6. Semua pihak yang telah membantu secara langsung maupun tidak langsung

Atas segala bantuan dan saran yang telah diberikan kepada penyusun.

Mengingat keterbatasan kemampuan penyusun dalam menyusun laporan Tugas Akhir, maka penyusunan laporan Tugas Akhir ini tidak terlepas dari kekurangan dan kesalahan, untuk itu penyusun mohon maaf yang sebesar-besarnya.

Besar harapan penyusun, laporan Tugas Akhir ini dapat memberikan manfaat bagi pihak-pihak yang bersangkutan.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, Agustus 2001

Penyusun

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
HALAMAN MOTO	iii
HALAMAN PERSEMBAHAN	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Metodologi Penelitian	3
1.7 Sistematika Penulisan	5
BAB II LANDASAN TEORI	
2.1 Tinjauan Pustaka	8
2.2 Landasan Teori	9
2.2.1 Definisi Sistem	9
2.2.2 Definisi Informasi	10
2.2.3 Definisi Manajemen	10
2.2.4 Sistem Informasi Manajemen	10
2.2.5 Konsep – Konsep Pokok SIM	11
2.2.6 Kriteria Sistem yang Baik	12

2.3	<i>Contex Diagram</i>	13
2.4	<i>Data Flow Diagram Levelled</i>	14
2.5	Definisi Basis Data	18
2.6	Normalisasi	21
2.6.1	Bentuk Normal Kesatu (First Normal Form)	22
2.6.2	Bentuk Normal Kedua (Second Normal Form)	22
2.6.3	Bentuk Normal Ketiga (Third Normal Form)	23
2.6.4	Bentuk Normal Keempat dan Bentuk Normal Kelima ...	24
2.7	Kardinalitas/ Derajat Relasi	26
BAB III	ANALISIS KEBUTUHAN PERANGKAT LUNAK	
3.1	Metode Analisis.....	29
3.1.1	Metode Pengumpulan Data	29
3.1.1.1	Metode Wawancara	30
3.1.1.2	Metode Observasi	31
3.1.1.3	Metode Literatur	32
3.2	Hasil Analisis Kebutuhan	32
3.2.1	Identifikasi Masalah	32
3.2.2	Identifikasi Penyebab Masalah	32
3.2.3	Kebutuhan Data Masukan	33
3.2.4	Kebutuhan Proses	33
3.2.5	Kebutuhan Data Keluaran	33
3.2.6	Kebutuhan Antarmuka	34
3.2.7	Kebutuhan Perangkat Lunak	34
3.2.8	Kebutuhan Perangkat Keras	35
3.3	Keamanan Data	35
BAB IV	PERANCANGAN PERANGKAT LUNAK	
4.1	Metode Perancangan	37
4.2	Hasil Perancangan	37

4.2.1 Perancangan Sistem Secara Global dengan Menggunakan Diagram Konteks	38
4.2.2 Perancangan Model Proses dengan Data <i>Flow Diagram</i> (DFD)	38
4.2.2.1 DFD Level 2 Proses Pemasukkan Obat	41
4.2.2.2 DFD Level 2 Proses Transaksi Obat	42
4.3 Normalisasi	43
4.3.1 Normalisasi Data Pembelian Obat	43
4.3.2 Normalisasi Data Penjualan Obat	46
4.4 Relasi Tabel	49
4.5 Perancangan Basis Data	50
4.6 Perancangan Menu	55
4.7 Perancangan Output	57

BAB V IMPLEMENTASI PERANGKAT LUNAK

5.1 Batasan Implementasi	59
5.2 Alasan Pemilihan Aplikasi Pengembang	59
5.3 Keamanan Sistem	60
5.4 Pasilitas Pengembang	60
5.5 Implementasi Sistem Informasi Farmasi	63
5.5.1 Implementasi <i>Form</i> Data Obat	64
5.5.2 Implementasi <i>Form</i> Data Pemasok	70
5.5.3 Implementasi <i>Form</i> Data Dokter	72
5.5.4 implementasi <i>Form</i> Data Pasien	74
5.5.5 implementasi <i>Form</i> Cari Obat	76
5.5.6 Implementasi <i>Form</i> Cari Pemasok	79
5.5.7 implementasi <i>Form</i> Cari Dokter	81
5.5.8 Implementasi <i>Form</i> Pembelian Obat	82
5.5.9 Implementasi <i>Form</i> Nota Penjualan	84

5.5.10 Implementasi <i>Form</i> Return Obat	85
5.5.11 Implementasi <i>Form</i> User	87
5.5.12 Implementasi <i>Form</i> Akses Aplikasi Farmasi RSUD Muhammadiyah	89
5.5.13 Implementasi <i>Form</i> Restore Data	92
5.5.14 Implementasi <i>Form</i> Backup Data	95
5.5.15 Implementasi <i>Form</i> Penghapusan Data	98
5.5.16 Implementasi Pembuatan Bentuk Laporan	100
5.5.17 Implementasi Menu Utama	100
 BAB VI ANALISIS KERJA PERANGKAT LUNAK	
6.1 Pendahuluan	103
6.2 Analisis Kerja Terhadap Akses Aplikasi	104
6.3 Analisis Kerja Terhadap Berkas Obat	104
6.4 Analisis Kerja Terhadap Berkas Pemasok	104
6.5 Analisis Kerja Terhadap Berkas Dokter	105
6.6 Analisis Kerja Terhadap Berkas Pasien	105
6.7 Analisis Kerja Terhadap Berkas Pembelian	106
6.8 Analisis Kerja Terhadap Berkas Penjualan	106
6.9 Analisis Kerja Terhadap Berkas Return	107
6.10 Pesan atau Informasi	107
6.11 Analisis Sistem Terhadap Masalah	110
6.12 Evaluasi Sistem	112
 BAB VII PENUTUP	
7.1 Kesimpulan	113
7.2 Saran	114

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1	Diagram konteks sistem informasi farmasi	14
Gambar 2.2	DFD sistem informasi pembelian obat	15
Gambar 2.3	Contoh proses	16
Gambar 2.4	Aliran dan proses	17
Gambar 2.5	Penyimpanan	17
Gambar 2.6	Terminator	18
Gambar 2.7	Contoh relasi satu ke satu	27
Gambar 2.8	Contoh relasi satu ke banyak	27
Gambar 2.9	Contoh relasi banyak ke satu	28
Gambar 2.10	Contoh relasi banyak ke banyak	28
Gambar 4.1	Diagram Konteks Sistem Informasi Farmasi	38
Gambar 4.2	DFD level 1 sistem informasi	40
Gambar 4.3	DFD Level 2 proses pemasukan	41
Gambar 4.4	DFD Level 2 proses transaksi	42
Gambar 4.5	Atribut-atribut hasil rangkuman data pembelian	44
Gambar 4.6	Bentuk normal ke-1 pada data pembelian obat	45
Gambar 4.7	Bentuk normal ke-2 pada data pembelian obat	45
Gambar 4.8	Bentuk normal ke-3 pada data pembelian	46
Gambar 4.9	Tabel universal pada data penjualan	47
Gambar 4.10	Bentuk normal ke-1 data penjualan	47
Gambar 4.11	Bentuk normal ke-2 data penjualan	48
Gambar 4.12	Bentuk normal ke-2 data penjualan	48
Gambar 4.13	Relasi tabel sistem informasi farmas	49
Gambar 4.14	Tampilan menu utama	57

Gambar 5.1	<i>Toolbar form control</i>	62
Gambar 5.2	Form obat <i>page frame</i> pemasukkan data obat	65
Gambar 5.3	Form obat <i>page frame</i> daftar data obat	68
Gambar 5.4	Form data pemasok <i>page frame</i> pemasukan data pemasok	71
Gambar 5.5	Form data pemasok <i>page frame</i> daftar pemasok	72
Gambar 5.6	Form data dokter <i>page frame</i> pendataan dokter	73
Gambar 5.7	Form data dokter <i>page frame</i> daftar dokter	74
Gambar 5.8	Form data pasien <i>page frame</i> pendataan pasien	75
Gambar 5.9	Form data pasien <i>page frame</i> daftar pasien	76
Gambar 5.10	Form cari obat	77
Gambar 5.11	Form cari pemasok	79
Gambar 5.12	Form cari dokter	81
Gambar 5.13	Form pembelian obat	83
Gambar 5.14	Form nota penjualan	84
Gambar 5.15	Modifikasi <i>text</i> pada <i>form</i> nota penjualan	85
Gambar 5.16	Form return obat	86
Gambar 5.17	Form user	88
Gambar 5.18	Form akses aplikasi farmasi RSUD Muhammadiyah	90
Gambar 5.19	Form restore data	92
Gambar 5.20	Form backup data	95
Gambar 5.21	Form perawatan	98
Gambar 5.22	Kotak dialog menu designer	101
Gambar 6.1	Konfirmasi kesalahan pengisian kerangkaan kode obat	108
Gambar 6.2	Konfirmasi kesalahan pengisian data	108
Gambar 6.3	Peringatan kesalahan pemasukkan <i>password</i>	109
Gambar 6.4	Informasi pembelian obat	109
Gambar 6.5	Konfirmasi penjualan obat	109
Gambar 6.6	Tampilan kotak pesan	111

DAFTAR GAMBAR

Gambar 2.1	Diagram konteks sistem informasi farmasi	14
Gambar 2.2	DFD sistem informasi pembelian obat	15
Gambar 2.3	Contoh proses	16
Gambar 2.4	Aliran dan proses	17
Gambar 2.5	Penyimpanan	17
Gambar 2.6	Terminator	18
Gambar 2.7	Contoh relasi satu ke satu	27
Gambar 2.8	Contoh relasi satu ke banyak	27
Gambar 2.9	Contoh relasi banyak ke satu	28
Gambar 2.10	Contoh relasi banyak ke banyak	28
Gambar 4.1	Diagram Konteks Sistem Informasi Farmasi	38
Gambar 4.2	DFD level 1 sistem informasi	40
Gambar 4.3	DFD Level 2 proses pemasukan	41
Gambar 4.4	DFD Level 2 proses transaksi	42
Gambar 4.5	Atribut-atribut hasil rangkuman data pembelian	44
Gambar 4.6	Bentuk normal ke-1 pada data pembelian obat	45
Gambar 4.7	Bentuk normal ke-2 pada data pembelian obat	45
Gambar 4.8	Bentuk normal ke-3 pada data pembelian	46
Gambar 4.9	Tabel universal pada data penjualan	47
Gambar 4.10	Bentuk normal ke-1 data penjualan	47
Gambar 4.11	Bentuk normal ke-2 data penjualan	48
Gambar 4.12	Bentuk normal ke-2 data penjualan	48
Gambar 4.13	Relasi tabel sistem informasi farmas	49
Gambar 4.14	Tampilan menu utama	57

ABSTRAKSI

Tugas Akhir ini berjudul “Sistem Informasi Farmasi di RSUD PKU Muhammadiyah Yogyakarta”, pada Bagian Farmasi melayani distribusi obat sehingga dibutuhkan kecepatan, keakuratan data dan informasi yang terorganisir dengan baik.

Sistem Informasi Farmasi ini dianalisis dan dirancang dengan menggunakan pendekatan terstruktur (Structure Approach) dengan tool (alat) Data Flow Diagram (DFD), normalisasi dan struktur database.

Berdasarkan analisis dan perancangan terstruktur, maka sistem ini diimplementasikan dengan menggunakan software Visual Foxpro Versi VI .

Selain kecepatan, keakuratan data dan informasi yang terorganisir dengan baik, Sistem Informasi Farmasi ini juga mempunyai obyektivitas untuk menyiapkan laporan-laporan, seperti laporan persediaan obat, laporan pembelian obat, laporan penjualan, laporan return.

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Informasi dalam era globalisasi seperti sekarang ini sudah bukan rahasia, bahwa informasi telah menjadi sumber daya strategis bagi kemajuan suatu perusahaan atau instansi, kecepatan dan keakuratan mengolah informasi menjadi hal yang sangat penting dalam pengambilan keputusan dan menjadi alat bersaing untuk memperoleh hasil yang diharapkan. Informasi memperlihatkan gejala yang semakin meningkat, sehingga perlu penanganan yang tepat dan sesuai dengan prinsip efisiensi waktu, tepat guna dan dapat dipercaya. Tersedianya data yang semakin banyak mendorong manusia untuk mencari sistem baru agar setiap data yang diperoleh dapat diolah menjadi informasi yang baik.

Teknologi komputer dalam penerapannya dapat mencakup dalam berbagai bidang pelayanan, misalnya dalam bidang pelayanan kesehatan masyarakat seperti di rumah sakit. Dalam sebuah rumah sakit bidang pelayanannya sangat banyak, seperti pelayanan gizi, pelayanan rawat inap, pelayanan rawat jalan, pelayanan medis dan pengobatan, laboratorium dan masih banyak lagi. Dalam hal ini rumah sakit PKU Muhammadiyah Yogyakarta merupakan rumah sakit yang memberikan pelayanan kesehatan bagi seluruh lapisan masyarakat dan dalam pelayanannya rumah sakit PKU Muhammadiyah sudah menggunakan sistem informasi berbasis komputer dalam

beberapa bidang pelayanan seperti dalam bidang pelayanan obat – obatan di bagian farmasi. Namun demikian sistem informasi pengolahan data obat di bagian farmasi yang digunakan hanya sistem informasi manajemen saja sedangkan dari pihak pasien dan keluarga tidak bisa menggunakan fasilitas tersebut. Dengan menggunakan sistem informasi berbasis multiuser diharapkan pelayanan terhadap masyarakat akan lebih mudah.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas maka penulis merumuskan masalah yang akan dibahas dalam penelitian adalah menerapkan sistem informasi farmasi di rumah sakit PKU Muhammadiyah Yogyakarta untuk mempermudah dalam memberikan informasi untuk bagian-bagian yang berhubungan dengan bagian farmasi dan untuk pasien.

1.3 Batasan Masalah

Sesuai dengan judul penelitian ini, maka batasan masalah yang akan diteliti hanya mengenai pengolahan data obat berbasis multiuser yang meliputi:

1. Data persediaan obat, data obat yang terjual, harga obat, nama obat, jenis obat dan tanggal kadaluarsa yang datanya dapat diakses oleh bagian penjualan obat (apoiek), bagian apotek farmasi, direktur dan oleh pasien itu sendiri.
2. Dalam pembelian obat ke pemasok dilakukan dengan tunai, tidak melakukan proses utang-piutang.

1.4 Tujuan Penelitian.

Tujuan penelitian di rumah sakit umum PKU Muhammadiyah Yogyakarta adalah .

1. Dapat belajar menganalisa masalah-masalah yang ada di bagian Farmasi RSU PKU Muhammadiyah dan dapat menerapkan ilmu yang didapat di bangku kuliah.
2. Melakukan desain dan implementasi sistem informasi farmasi di rumah sakit PKU Muhammadiyah Yogyakarta.

1.5 Manfaat Penelitian

Dari hasil penelitian nantinya diharapkan dapat memberikan manfaat sebagai berikut :

1. Memberikan kemudahan dalam pelayanan informasi kepada pasien atau pegawai farmasi serta bagian-bagian yang berhubungan dengan bagian farmasi.
2. Bagi instansi dapat dipergunakan sebagai bahan pertimbangan untuk pengolahan data obat di bagian farmasi rumah sakit PKU Muhammadiyah Yogyakarta.

1.6 Metodologi Penelitian

Penelitian yang dilakukan merupakan bentuk sebuah studi kasus pada rumah sakit PKU Muhammadiyah Yogyakarta. Penelitian ini difokuskan pada bagian farmasi. Adapun tahapan dari metodologi penelitian adalah sebagai berikut :

1. Penelitian Lapangan

Penelitian merupakan usaha pengumpulan data dan informasi (data primer dan sekunder) secara intensif dan disertai analisis pada semua data yang dikumpulkan dengan cara :

- a. Wawancara, yaitu dengan mengadakan tanya jawab dengan responden dalam hal ini adalah para pimpinan dan staf rumah sakit.
- b. Observasi, mengamati secara langsung kegiatan perusahaan untuk mendapatkan gambaran nyata atas kasus yang diteliti.
- c. Studi pustaka, merupakan penelitian yang dilakukan untuk memperoleh informasi mengenai landasan teori, sistematika penulisan dan kerangka berpikir ilmiah dari literatur dan buku acuan untuk memecahkan permasalahan.

2. Implementasi Perangkat Lunak

Dalam mengimplementasikan masalah ke dalam suatu perangkat lunak, mempunyai tahapan-tahapan, yaitu :

- Menganalisa kebutuhan perangkat lunak
- Merancang atau mendisain perangkat lunak
- Implementasi
- Pengujian sistem

1.7 Sistematika Penulisan

Dalam penulisan tugas akhir ini terdiri dari tujuh bab, yaitu :

BAB I PENDAHULUAN

Dalam bab I dijelaskan mengenai latar belakang masalah yang ada, rumusan masalah, manfaat penelitian, tujuan penelitian, batasan masalah, metode pengumpulan data dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini berisi tentang uraian teori-teori yang diambil dari literatur-literatur yang ada dan berhubungan dengan permasalahan yang dihadapi seperti : definisi sistem, definisi informasi, definisi manajemen, sistem informasi manajemen, konsep – konsep pokok SIM, kriteria sistem yang baik, *Data Flow Diagram Levelled*, definisi basis data, normalisasi, bentuk-bentuk normalisasi, *Entity Relationship*, serta kardinalitas/ derajat relasi.

BAB III ANALISIS KEBUTUHAN PERANGKAT LUNAK

Menguraikan tentang analisis kebutuhan perangkat lunak yang meliputi, metode analisis yang digunakan yaitu analisis terstruktur dimana transformasi proses *input-output* dinyatakan dengan diagram arus data (*Data Flow Diagram*), metode pengumpulan data, identifikasi titik keputusan, dan hasil analisis kebutuhan perangkat lunak yang terdiri dari kebutuhan masukan, kebutuhan keluaran, kebutuhan proses,

kebutuhan perangkat lunak, kebutuhan perangkat keras, antarmuka yang akan dikembangkan serta keamanan data.

BAB IV PERANCANGAN PERANGKAT LUNAK

Bab ini memuat uraian tentang metode perancangan terstruktur (*Structured Design Method*) menggunakan *Data Flow Oriented Design*, *Data Flow Oriented Design*, hasil perancangan yang meliputi perancangan sistem secara global dengan menggunakan diagram konteks, perancangan model proses dengan *Data Flow Diagram* (DFD), normalisasi, relasi tabel, perancangan basis data, perancangan masukan, perancangan keluaran, serta antarmuka yang dikembangkan.

BAB V IMPLEMENTASI PERANGKAT LUNAK

Bab ini menjelaskan batasan implementasi yang berisi keamanan sistem, serta dijelaskan juga mengenai alasan pemilihan aplikasi, serta implementasinya itu sendiri.

BAB VI PENGUJIAN SISTEM

Berisi analisis kinerja perangkat lunak yang menjelaskan tentang kinerja dari perangkat lunak yang telah dibuat serta menguji kebenaran sistem dengan, baik itu dokumentasi maupun literatur-literatur.

BAB VII KESIMPULAN

Mengemukakan kesimpulan dari hasil penelitian yang telah dilakukan serta saran yang membangun untuk perusahaan.

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Ilmu sistem berkaitan dengan kompleksitas (kerumitan) dari suatu dunia nyata. Kompleksitas dapat dipandang dari dua aspek yaitu : aspek sistem itu sendiri dan aspek manusia.

Sifat ilmu sistem adalah memperkenalkan konsep sistem untuk membentuk komponen – komponen struktural yang berupa kerangka sistem sebagai hasil dari pemikiran tertentu dan juga wawasan yang memadai untuk penerapan kerangka sistem.

Pengertian sistem tergantung pada latar belakang cara pandang orang yang mencoba diidentifikasinya. Menurut rekayasa, sistem dipandang sebagai proses masukan (*input*) yang ditransformasikan menjadi keluaran (*output*) tertentu. Menurut awam, sistem dipandang sebagai cara atau metode untuk mencapai suatu tujuan.

Sistem sebagai suatu agregasi atau kumpulan obyek – obyek yang terangkai dalam interaksi dan saling ketergantungan yang teratur . Sistem sebagai suatu set elemen-elemen yang berada dalam keadaan yang saling berhubungan. Dengan demikian sistem dapat berupa kesatuan yang terdiri atas jaringan kerja kausal dari bagian – bagian yang saling bergantung.

Dari kumpulan definisi tentang sistem dapat dilihat adanya kesamaan pengertian tentang sistem. Secara umum sistem didefinisikan sebagai sekumpulan elemen-elemen yang saling berinteraksi untuk mencapai suatu tujuan tertentu di dalam lingkungan yang kompleks.

Dalam sistem informasi diperlukan data sebagai bahan mentah dari informasi yang dirumuskan sebagai sekelompok lambang-lambang acak yang menunjukkan jumlah atau tindakan-tindakan [DAV93]. Informasi sebagai data yang telah diolah menjadi suatu bentuk penting bagi yang menerimanya dan bermanfaat dalam pengambilan suatu keputusan.

2.2 Landasan Teori

2.2.1 Definisi Sistem

Sistem merupakan seperangkat elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem terdiri dari beberapa subsistem, subsistem – subsistem dalam suatu sistem tidak dapat berdiri sendiri – sendiri, tetapi saling berhubungan membentuk suatu kesatuan sehingga tujuan sistem dapat tercapai. Dalam hal ini sistem adalah sekumpulan elemen-elemen yang saling berinteraksi untuk mencapai suatu tujuan tertentu didalam lingkungan yang kompleks. Elemen yang dimaksud disini adalah bagian – bagian dari sistem seperti input, proses dan output dimana setiap bagian mempunyai beberapa nilai atau harga yang bersama-sama menggambarkan keadaan sistem pada saat tertentu.

2.2.2 Definisi Informasi

Informasi adalah sebuah istilah yang tidak tepat dalam pemakaiannya secara umum. Informasi di dapat mengenai data mentah, data tersusun, kapasitas sebuah saluran komunikasi, dan sebagainya. Tetapi ada beberapa gagasan yang mendasari pemakaian istilah “informasi” dalam sistem informasi : informasi memperkaya pengajian, mempunyai nilai kejutan atau menerangkan sesuatu yang penerimanya tidak tahu atau tidak tersangka.

Definisi umum untuk informasi dalam pemakaian sistem informasi adalah sebagai berikut [JOG89] :

“ Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang “.

2.2.3 Definisi Manajemen

Definisi manajemen yaitu suatu proses tertentu yang terdiri atas pengorganisasian, pergerakan dan pengawasan yang dilakukan untuk menentukan dan mencapai tujuan yang telah ditetapkan dengan menggunakan manusia dan sumber – sumber lainnya.

2.2.4 Sistem Informasi Manajemen

Sistem informasi manajemen adalah pendekatan – pendekatan yang direncanakan dan disusun untuk memberikan bantuan pegawai yang memudahkan proses manajerial kepada pejabat pimpinan.

Definisi sebuah sistem informasi manajemen secara umum adalah sebuah sistem manusia atau mesin yang terpadu (*integrated*) untuk menyajikan informasi guna mendukung fungsi operasi manajemen dan pengambilan keputusan dalam sebuah organisasi [DAV93]. Sistem ini menggunakan perangkat keras dan perangkat lunak komputer, prosedur pedoman, model manajemen dan keputusan dan sebuah “*database*”.

Komputer telah menambahkan teknologi baru pada sistem informasi. Akibatnya sebuah sistem informasi berdasarkan komputer akan berbeda dengan sistem – sistem yang diolah secara manual atau elektro mekanis. Pada dasarnya kita dapat membahas sistem informasi manajemen tanpa komputer tetapi adalah kemampuan komputer yang membuat SIM terwujud.

2.2.5 Konsep – Konsep Pokok SIM

Sebuah sistem informasi manajemen bukanlah sekedar suatu perkembangan teknologi juga berhubungan dengan organisasi dan manusia pengolahnya, oleh sebab itu pemahaman utuh terhadap sistem informasi berdasarkan komputer harus juga termasuk memahami konsep – konsep yang berhubungan dengan informasi, pemakaian informasi dan nilai informasi. Konsep – konsep pokok yang tercakup dalam SIM adalah [DAV93] :

1. Informasi

Informasi menambahkan sesuatu pada penyajian yaitu berhubungan dengan waktu dan mutu.

2. Manusia sebagai pengolah informasi

Kemampuan manusia sebagai pengolah informasi menentukan keterbatasan dalam sistem informasi dan mengesankan dasar – dasar rancangan mereka.

3. Konsep sistem

Karena SIM adalah sebuah sistem maka konsep sistem perlu untuk memahami dan merancang pada pengembangan sistem informasi.

4. Konsep pengambilan keputusan

Rancangan SIM bukan hanya harus mencerminkan rancangan optimasi, tetapi juga teori perilaku pengambilan keputusan dalam organisasi.

5. Nilai informasi

Informasi akan mengubah keputusan. Perubahan dalam nilai hasil akan menentukan nilai informasi.

2.2.6 Kriteria Sistem yang Baik

Kriteria sistem yang baik adalah [JOG89] :

a. Kegunaan

Sistem harus menghasilkan informasi yang tepat pada waktunya dan relevan untuk proses pengambilan keputusan manajemen dan personal operasi di dalam organisasi.

b. Ekonomis

Semua bagian dari sistem termasuk laporan-laporan, pengawasan-pengawasan, dan lain-lain harus menyumbangkan suatu nilai tambah sekurang-kurangnya sebesar biayanya.

c. Keandalan

Keluaran (*output*) sistem harus mempunyai tingkat ketelitian yang tinggi dan sistem itu harus mampu beroperasi secara efektif dan efisien.

d. Kapasitas

Suatu sistem harus mempunyai kapasitas yang memadai untuk menangani periode-periode operasi puncak seperti pada saat operasi normal.

e. Kesederhanaan

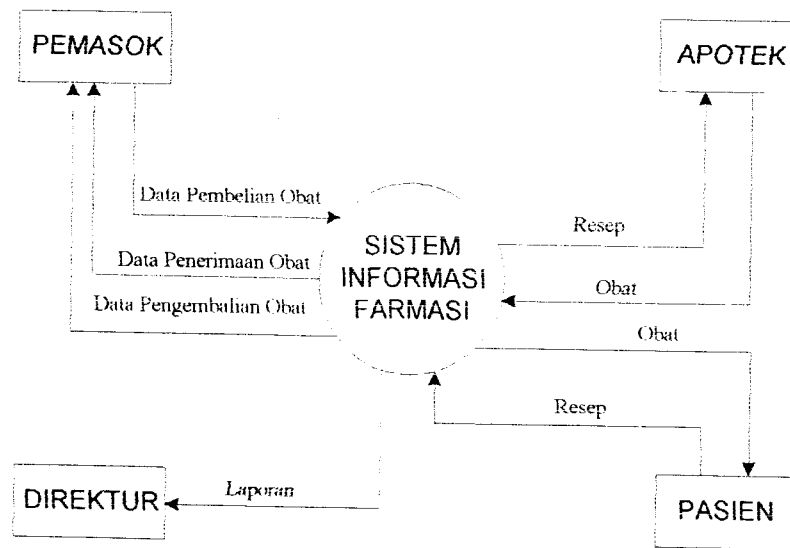
Sistem harus cukup sederhana sehingga struktur dan operasinya dapat dengan mudah dimengerti dan prosedurnya dapat diikuti.

f. Fleksibilitas

Sistem harus cukup fleksibel untuk menampung perubahan-perubahan.

2.3 Context Diagram

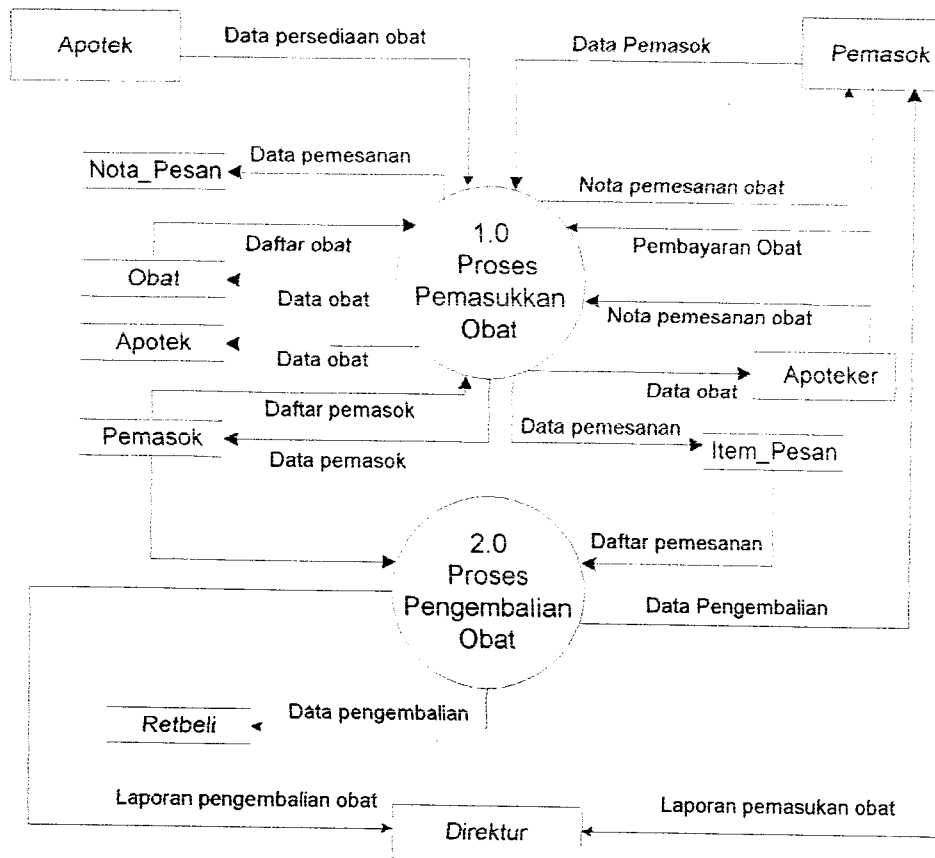
Context diagram adalah kasus khusus DFD (bagian dari DFD yang berfungsi memetakan model lingkungan), yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. Gambar 2.1 dibawah ini adalah contoh context diagram.



Gambar 2.1 Diagram konteks sistem informasi farmasi

2.4 Data Flow Diagram Levelled

Model ini menggambarkan sistem sebagai jaringan kerja antar fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data yang disebut DFD. Sebagai perangkat analisis, model ini hanya mampu memodelkan sistem dari satu sudut pandang yaitu sudut pandang fungsi. Gambar 2.2 di bawah ini adalah contoh DFD sistem informasi pembelian obat.



Gambar 2.2 DFD Sistem informasi pembelian obat

Ada empat komponen dalam model ini yaitu [POH97] :

1. Proses

Komponen pertama dalam model ini dinamakan proses, dapat disebut juga gelembung (*bubble*), fungsi, dan transformasi. Proses menunjukkan transformasi dari masukan menjadi keluaran, dalam hal ini sejumlah masukan dapat menjadi hanya satu keluaran ataupun sebaliknya. Proses direpresentasikan dalam bentuk lingkaran (bisa juga oval atau bujur sangkar dengan sudut melengkung). Proses umumnya didefinisikan dengan kata tunggal atau kalimat sederhana, sehingga

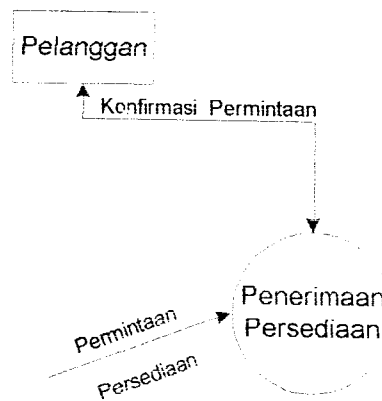
definisi yang dipakai lebih mengidentifikasi subyek proses daripada obyek itu sendiri.



Gambar 2.3 Contoh proses

2. Aliran

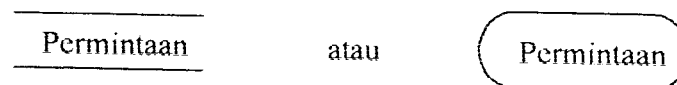
Komponen ini direpresentasikan dengan menggunakan panah yang menuju ke/dari proses. Digunakan untuk menggambarkan paket data atau informasi dari satu bagian ke bagian lain dari sistem dimana penyimpanan mewakili lokasi penyimpanan data. Nama berfungsi untuk mendefinisikan arti dari aliran tersebut, dan ditulis untuk mengidentifikasi aliran tersebut. Ujung panah menunjukkan kemana data bergerak ke/dari proses, penyimpanan atau terminator atau keduanya. Aliran yang digambarkan sebagai panah dengan dua ujung menggambarkan terjadinya dialog. Aliran dapat juga menyebar atau menyatu, misalnya sejumlah atribut dapat membentuk satu aliran, atau satu aliran menyebar menjadi sejumlah atribut. Atribut dalam hal ini dapat merupakan bagian atau duplikasi dari aliran.



Gambar 2.4 Aliran dan proses

3. Penyimpanan

Komponen ini digunakan untuk memodelkan kumpulan data atau paket data. Notasi yang digunakan adalah garis sejajar, segiempat dengan sudut melengkung, atau persegi panjang. Panah yang bergerak dari penyimpanan berarti penggunaan paket data tunggal (*single packet*), penggunaan data paket kelompok (*multiple packet*), penggunaan pembagian paket (*portions of a packet*) dan penggunaan perbagian lebih dari satu paket. Jika aliran tidak didefinisikan, maka keseluruhan paket informasi dari penyimpanan digunakan, demikian juga dengan aliran yang mempunyai nama sama dengan penyimpanan.

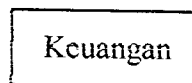


Gambar 2.5 Penyimpanan

4. Terminator

Komponen berikutnya dalam model ini direpresentasikan menggunakan persegi panjang yang mewakili entiti luar dimana sistem berkomunikasi. Biasanya notasi ini melambangkan orang atau kelompok orang misalnya organisasi luar sistem, grup, departemen, perusahaan pemerintah, dan berada di luar kontrol sistem yang dimodelkan.

Terminator merupakan bagian luar sistem dan aliran data (panah) yang dihubungkan dengan terminator (ke/dari proses, ke/dari penyimpanan) dalam sistem memodelkan hubungan antara sistem dengan dunia luar.



Gambar 2.6 Terminator

2.5 Definisi Basis Data

Basis data terdiri dari 2 kata, yaitu basis dan data. Basis dapat diartikan sebagai markas atau gudang, sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu obyek. Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti [FAT99]:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data/arsip. Tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip. Perbedaannya hanya terletak pada media penyimpanan yang digunakan. Basis data menggunakan media penyimpanan elektronik. Secara lengkap, pemanfaatan basis data dilakukan untuk memenuhi sejumlah tujuan (objektif) sebagai berikut [FAT99] :

1. Kecepatan dan kemudahan (*speed*)

Pemanfaatan basis data memungkinkan kita untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan lebih cepat dan mudah.

2. Efisiensi ruang penyimpanan (*space*)

Keterkaitan yang erat antar kelompok data dalam sebuah basis data, maka redundansi (pengulangan) data pasti akan selalu ada. Dengan basis data efisiensi/optimalisasi penggunaan ruang penyimpanan dapat dilakukan karena kita dapat melakukan penekanan jumlah redundansi data, baik dengan menerapkan

sejumlah pengkodean atau dengan membuat relasi-relasi (dalam bentuk file) antar kelompok data yang saling berhubungan.

3. Keakuratan (*Accuracy*)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan tipe data, domain data keunikan data dan sebagainya, yang secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan pemasukan/penyimpanan data.

4. Ketersediaan (*Availability*)

Tidak semua data dalam basis data selalu kita gunakan/butuhkan. Karena itu kita dapat memilah adanya data utama/master/referensi, data transaksi data histori dan data kadaluarsa, dapat kita atur untuk dilepaskan dari sistem basis data yang sedang aktif.

5. Kelengkapan (*Completeness*)

Dalam sebuah basis data, disamping data kita juga harus menyimpan struktur (baik yang mendefinisikan objek-objek dalam basis data maupun definisi detail dari tiap objek, seperti struktur file/tabel atau indeks). Untuk mengakomodasi kebutuhan kelengkapan data yang semakin berkembang, maka kita hanya dapat menambah record-record data, tetapi juga dapat melakukan perubahan struktur dalam basis data, baik dalam bentuk penambahan objek baru (tabel) atau dengan penambahan field-field baru pada suatu tabel.

6. Keamanan (*Security*)

Untuk sistem yang besar dan serius aspek keamanan harus diterapkan dengan ketat. Dengan begitu kita dapat menentukan siapa-siapa pemakai yang boleh menggunakan basis data beserta objek-objek didalamnya dan menentukan jenis-jenis operasi apa saja yang boleh dilakukan.

7. Kebersamaan pemakaian (*Sharability*)

Basis data yang dikelola oleh sistem (aplikasi) yang mendukung lingkungan multiuser, akan dapat memenuhi kebutuhan ini, tetapi tetap dengan menjaga/menghindari terhadap munculnya persoalan baru seperti inkonsistensi data (karena data yang sama diubah oleh banyak pemakai pada saat yang bersamaan) atau kondisi *deadlock* (karena ada banyak pemakai yang saling menunggu untuk menggunakan data).

2.6 Normalisasi

Tujuan desain adalah mengkonstruksi relasi tanpa redundansi. Untuk melakukan ini diperlukan pendefinisian kondisi yang memenuhi relasi tanpa redundansi. Kondisi ini didefinisikan dalam terminologi relasi normal (*normal relations*). Relasi seharusnya berada dalam bentuk normal tertinggi dan bergerak dari bentuk normal satu dan seterusnya untuk setiap kali membatasi hanya satu jenis redundansi.

2.6.1 Bentuk Normal Kesatu (*First Normal Form*)

Relasi berada dalam bentuk normal jika semua nilai atribut sederhana. Bentuk normal kesatu mempunyai ciri setiap bentuk dalam file datar atau rata, data dibentuk dalam satu record demi satu record dan nilai dari field-field berupa *atomic value*. Tidak ada set atribut yang berulang atau atribut yang bernilai ganda. Tiap field hanya satu pengertian. Dari bentuk tidak normal, dibentuk menjadi bentuk normal kesatu dengan memisahkan data pada field-field yang tepat dan bernilai atomik.

Tabel 2.1 Relasi tidak normal order

# ORDER	Tanggal Order	Isi Order	
t20	5 Juli 1987	# Item	Jumlah Item
		pc6	24
		bw3	83
		ty6	37

Tabel 2.2 Relasi normal order

#_Order	Tanggal_Order	#_Item	Jumlah_Item
T20	5 Juli 1987	pc6	24
T20	5 Juli 1987	bw3	83
T20	5 Juli 1987	ty6	37

2.6.2 Bentuk normal kedua (*Second Normal Form*)

Bentuk normal tahap kedua (2NF) terpenuhi jika pada sebuah tabel, semua atribut yang tidak termasuk dalam *key primer* memiliki ketergantungan fungsional pada *key primer* secara utuh. Sebuah tabel dikatakan tidak memenuhi 2NF, jika ketergantungannya hanya bersifat parsial (hanya tergantung pada sebagian dari *key primer*).

Tabel 2.3 DATA PROYEK

#_Personil	#_Proyek	Anggaran_Proyek	Total waktu_personil_proyek
P1	Proj1	20	20
p3	Proj1	20	16
p2	Proj2	17	35
p2	Proj3	84	42
p3	Proj2	17	17
p2	Proj1	20	83
p4	Proj3	84	41
	Proj4	90	

Bentuk normal kedua sebagai berikut :

Tabel 2.4 PROYEK

#_Proyek	Anggaran Proyek
proj1	20
proj2	17
proj3	84
proj4	90

Tabel 2.5 PEKERJA relasi table normal ke-2

#_Personil	#_Proyek	Total waktu_personil_proyek
p1	proj1	20
p3	proj1	16
p2	proj2	35
p2	proj3	42
p3	proj2	17
p2	proj1	83
p4	proj3	41

2.6.3 Bentuk normal ketiga (*Third Normal Form*)

Relasi dalam bentuk normal kedua masih mungkin mengandung redundansi .
 Relasi dalam bentuk normal kedua tidak menyimpan fakta tentang bagian dari kunci relasi karena itu masih mungkin mengandung redundansi jika menyimpan fakta

tentang atribut bukan utama. Secara umum relasi dalam bentuk normal ketiga harus memenuhi syarat sebagai berikut :

1. Berada dalam bentuk normal kedua
2. Tidak berisi ketergantungan fungsional antara atribut bukan utama.

Tabel 2.6 PROYEK

# Proyek	Manajer	Tanggal lahir
p1	i	Desember 1968
p3	i	Desember 1967
p2	a	Januari 1974
p4	x	Juni 1977

Bentuk normal ketiga sebagai berikut :

Tabel 2.7 PROYEK

# Proyek	Manajer
p1	i
p3	i
p2	a
p4	x

Tabel 2.8 MANAJER relasi table normal ke-3

Manajer	Tanggal lahir
i	Desember 1968
a	Januari 1974
X	Juni 1977

2.6.4 Bentuk Normal Ke-4 dan Bentuk Normal Ke-5

Penerapan aturan normalisasi sampai tahap ketiga sesungguhnya sudah sangat memadai untuk menghasilkan tabel-tabel yang berkualitas baik. Namun demikian dari sejumlah literatur dijumpai adanya pembahasan tentang bentuk normal tahap keempat dan bentuk normal tahap kelima. Bentuk normal tahap keempat berkaitan

dengan sifat ketergantungan banyak-nilai (*Multivalue Dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Sedangkan bentuk normal tahap kelima (merupakan nama lain dari *Project-join normal form*/PJNF) berkenaan dengan ketergantungan relasi antar tabel (*Join dependency*).

Tabel 2.9 AREA PENJUALAN

Perwakilan penjualan	Pelanggan	Produk
Jajang	Computa	Printer
Jajang	Computa	CPU
Lilis	Kaledia	Printer
Lilis	Kaledia	Mouse
Lilis	Kaledia	Monitor
Jajang	Wisno	Printer
Jajang	Wisno	CPU
Lilis	Computa	Printer
Lilis	Computa	Mouse
Lilis	Computa	Monitor

Bentuk normal keempat adalah sebagai berikut :

Tabel 2.10 PELANGGAN

Perwakilan penjualan	Pelanggan
Jajang	Computa
Lilis	Kaledia
Jajang	Wisno
Lilis	Computa

Tabel 2.11 PRODUK relasi normal ke-4

Perwakilan Penjualan	Produk
Jajang	Printer
Jajang	CPU
Lilis	Printer
Lilis	Mouse
Lilis	Monitor

Tabel 2.12 AREA PENJUALAN bentuk normal ke-5

Perwakilan penjualan	Pelanggan	Produk
Jajang	Computa	Printer
Jajang	Computa	CPU
Lilis	Kaledia	Printer
Lilis	Kaledia	Mouse
Lilis	Kaledia	Monitor
Jajang	Wisno	Printer
Jajang	Wisno	CPU
Lilis	Computa	Printer
Lilis	Computa	Mouse
Lilis	Computa	Monitor

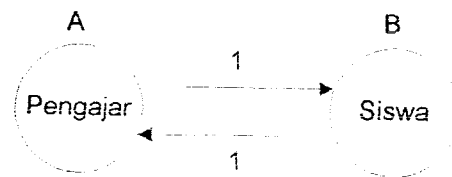
Dalam bentuk normal kelima ditambahkan konstrain/pembatas yaitu tidak semua perwakilan_penjualan dari produk dapat menjual produk ke pelanggan.

2.7 Kardinalitas/Derajat Relasi

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain . Dari sejumlah kemungkinan banyaknya hubungan antar entitas tersebut, kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari himpunan entitas yang satu ke himpunan entitas yang lain dan begitu juga sebaliknya. Kardinalitas relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dapat berupa :

1. Kardinalitas relasi satu ke Satu (*One to one*)

Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B berhubungan paling banyak dengan satu entitas pada himpunan entitas A.

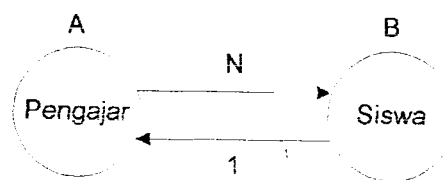


Gambar 2.7 Contoh relasi satu ke satu

Pada contoh relasi pada gambar 2.7 diasumsikan pada pelajaran privat dimana satu guru mengajar satu siswa dan satu siswa hanya diajar oleh satu guru

2. Kardinalitas relasi satu ke banyak (*One to many*)

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



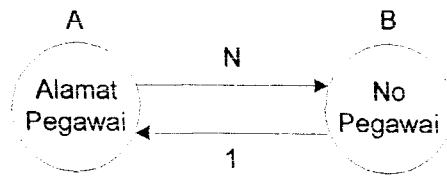
Gambar 2.8 Contoh relasi satu ke banyak.

Pada contoh relasi satu ke banyak seperti pada gambar 2.8 diasumsikan pada sistem pengajaran di Sekolah Dasar dimana satu guru mengajar banyak siswa dan siswa hanya diajar oleh satu guru.

3. Kardinalitas banyak ke satu (*Many to one*)

Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan satu entitas pada himpunan entitas B, tetapi tidak sebaliknya, dimana setiap

entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B.

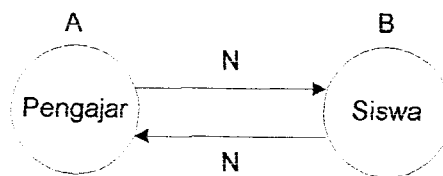


Gambar 2.9 Contoh relasi banyak ke Satu

Pada Gambar 2.9 dimisalkan pada satu alamat pegawai terdapat puluhan pegawai, maka hubungan antara atribut alamat pegawai dengan nomor pegawai adalah satu alamat menunjukkan banyak pegawai.

4. Kardinalitas relasi banyak ke banyak (*Many to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian juga sebaliknya, di mana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.



Gambar 2.10 Contoh relasi banyak ke banyak.

Pada Gambar 2.10 dimisalkan pada sistem pengajaran di Perguruan Tinggi dimana satu guru mengajar banyak siswa dan siswa diajar oleh banyak guru.

BAB III

ANALISIS KEBUTUHAN PERANGKAT LUNAK

3.1 Metode Analisis

Metode analisis yang digunakan adalah analisis terstruktur dimana transformasi informasi *input output* dinyatakan dengan diagram arus data (*Data Flow Diagram*) yang diperoleh dengan mengamati kebutuhan informasinya. Pembuatan diagram arus data dilihat berdasarkan aliran informasi dari sistem yang sudah ada. Dalam tahap ini penggunaan notasi-notasi untuk menggambarkan arus data dari sistem sangat membantu dalam proses komunikasi dengan pemakai sistem, untuk memahami sistem secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau dimana data tersebut disimpan untuk proses pengembangan selanjutnya.

Metode pengumpulan data menggunakan teknik wawancara, observasi, dan literatur. Metode analisis digunakan untuk mengidentifikasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

3.1.1 Metode Pengumpulan Data

Dalam penelitian ini metode pengumpulan data yang digunakan berupa wawancara, observasi, dan literatur.

3.1.1.1 Metode Wawancara

Wawancara (*interview*) adalah suatu bentuk komunikasi verbal, yaitu semacam percakapan yang bertujuan untuk memperoleh informasi. Wawancara telah diakui sebagai teknik pengumpulan data / fakta yang penting dan banyak dilakukan dalam pengembangan sistem informasi. Wawancara pada umumnya dapat dibedakan menjadi dua macam, yaitu:

- Wawancara berstruktur

Dalam wawancara ini semua pertanyaan telah dirumuskan sebelumnya dengan cermat, biasanya secara tertulis.

- Wawancara tak terstruktur (bebas)

Dalam wawancara serupa ini tidak dipersiapkan daftar pertanyaan sebelumnya. Pewawancara hanya menghadapi suatu masalah secara umum. Pertanyaan yang timbul secara spontanitas pewawancara.

Wawancara dilakukan pada bagian farmasi RSUD Muhammadiyah Yogyakarta, tujuan dilakukan wawancara, untuk mengetahui sistem kerja bagian farmasi, sehingga didapatkan data-data yang nantinya digunakan untuk pengembangan sistem informasi di bagian farmasi RSUD Muhammadiyah Yogyakarta. Hasil metode wawancara yang didapat adalah kumpulan data mengenai data-data yang diperlukan, untuk siapa informasi tersebut dibutuhkan dan hasil dari informasi tersebut yang berupa laporan.

3.1.1.2 Metode Observasi

Metode observasi atau pengamatan merupakan salah satu metode pengumpulan data / fakta yang cukup efektif. Observasi merupakan pengamatan langsung suatu kegiatan yang bertujuan untuk menguji validitas dari data hasil wawancara yang telah didapatkan. Metode observasi yang telah dijalankan dalam penelitian ini adalah metode yang langsung ke bagian farmasi sehingga dapat dilihat secara langsung tentang pengolahan data obat yang selama ini telah berjalan. Hasil observasi adalah sebagai berikut :

1. Fasilitas yang digunakan untuk sistem pengolahan data di bagian farmasi menggunakan software aplikasi Clipper yang hanya digunakan untuk pemasukan data obat.
2. Software yang digunakan tidak dapat untuk mengolah data menjadi bentuk informasi yang dapat digunakan oleh pegawai farmasi untuk pembuatan laporan dan oleh pasien untuk mengetahui harga obat.
3. Kesalahan dalam pemasukan data obat yang berakibat kode-kode obat menjadi rangkap, hal ini disebabkan oleh petugas yang kurang teliti dalam memasukkan data obat dan tidak didukung oleh software yang dapat mengatasi kerangkapan data.

3.1.1.3 Metode Literatur

Metode literatur merupakan metode pengumpulan data dari hasil pembacaan literatur yang berhubungan dengan sistem informasi manajemen dan dokumen-dokumen di bagian farmasi.

3.2 Hasil Analisis Kebutuhan

Hasil analisis yang diperoleh dari sistem pengolahan data obat di bagian Farmasi ada beberapa proses masukan data dan proses keluatan data.

3.2.1 Identifikasi masalah

Sistem pengolahan data di bagian farmasi RSUD Muhammadiyah masih menggunakan software aplikasi under DOS sehingga tampilannya kurang menarik baik dilihat dari bentuk form maupun warnanya dan tidak dilengkapi dengan fasilitas pesan kesalahan apabila user salah memasukkan kode obat disamping itu juga sering terjadi kerangkapan data obat. Hal ini menimbulkan informasi yang dihasilkan tidak akurat. Kerangkapan data obat ini disebabkan karena software yang ada tidak dilengkapi dengan fasilitas yang menampilkan pesan kesalahan untuk mengatasi apabila ada dua kode data yang sama.

3.2.2 Identifikasi Penyebab Masalah

Dari identifikasi masalah yang telah diteliti, maka dapat diketahui penyebab masalahnya. Identifikasi penyebab masalah dari sistem yang telah ada adalah software yang digunakan tidak dapat mengatasi masalah kerangkapan data obat.

Kerangkapan data obat yang ada disebabkan karena kesalahan petugas dalam memasukkan data obat.

3.2.3 Kebutuhan Data Masukan

Untuk proses masukan data, terdiri dari beberapa proses, antara lain:

- a. Proses pemasukkan obat (pembelian obat) dari pemasok ke apotek.
- b. Proses pengembalian obat (*return* pembelian).
- c. Proses penjualan obat ke pasien.

3.2.4 Kebutuhan Proses

Dalam sistem informasi bagian farmasi data obat dibutuhkan oleh pegawai farmasi itu sendiri maupun oleh pasien, jadi perlu diterapkan sistem informasi berbasis multiuser untuk mendukung kebutuhan akan informasi tersebut.

3.2.5 Kebutuhan Data Keluaran

Keluaran atau output adalah produk dari sistem informasi bagian farmasi adalah sebagai berikut :

1. Laporan data obat
2. Laporan daftar pemasok
3. Laporan daftar dokter
4. Laporan daftar pasien
5. Laporan persediaan obat
6. Laporan pembelian obat

7. Laporan penjualan obat
8. Laporan pengembalian obat
9. Laporan rincian resep
10. Laporan kadaluarsa

3.2.6 Kebutuhan Antarmuka

Kebutuhan antarmuka yang diinginkan oleh pemakai didasarkan pada hasil wawancara dengan pihak-pihak yang terkait. Antarmuka yang diharapkan yaitu *user friendly* (ramah pengguna) dengan tampilan yang lebih menarik serta dapat menjamin keamanan data yang ada. Antarmuka yang diharapkan adalah sebagai berikut :

1. Tampilan yang lebih menarik dibandingkan dengan sistem yang ada, diharapkan menggunakan software berbasis visual.
2. Dapat menjamin keamanan data yang ada dengan cara pengamanan kata sandi (*password*).

3.2.7 Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk penerapan Sistem Informasi Farmasi di RSUD Muhammadiyah berbasis multiuser dengan topologi jaringan bus linear adalah sebagai berikut :

- a. Novell Netware versi 3.11, 3.12, 4.1 atau versi yang lebih baru.
- b. Sistem operasi windows 95 atau versi di atasnya.
- c. Microsoft Visual Foxpro for windows versi 6.0

3.2.8 Kebutuhan Perangkat Keras

Perangkat keras yang diperlukan terdiri dari unit masukan, unit pengolah dan unit keluaran. Perangkat keras yang dibutuhkan untuk penerapan Sistem Informasi Farmasi di RSUD Muhammadiyah dalam multiuser (jaringan) yaitu :

a. Workstation

- Komputer PC dengan prosessor Pentium atau dengan prosessor yang lebih baik.
- Memori internal (RAM) minimal sebesar 16 MB.
- Monitor VGA atau yang lebih baik.
- Card LAN Ethernet NIC
- Mouse
- Printer Epson LX 800 atau di atasnya.

b. Server

- Komputer PC dengan prosessor Pentium atau komputer dengan prosessor yang lebih baik.
- Memori internal (RAM) minimal 32 MB
- Monitor VGA atau yang lebih baik.
- Card LAN Ethernet NIC

3.3 Keamanan Data

Keamanan data meliputi seluruh proses yang diperlukan untuk memastikan keamanan data didalam suatu sistem. Keamanan data merupakan salah satu unsur yang harus dipertimbangkan dalam proses desain suatu sistem, karena suatu sistem

tanpa keamanan data yang baik akan merugikan, sebab data akan bebas diakses oleh pihak-pihak yang tidak bertanggung jawab.

Keamanan data dapat diterapkan dengan pembuatan tabel pengguna (*user*) yang disertai dengan sandi (*password*) sehingga hanya pengguna yang namanya tercantum dalam tabel tersebut dan mengetahui sandi saja yang dapat mengakses data tersebut.

BAB IV

PERANCANGAN PERANGKAT LUNAK

4.1 Metode Perancangan

Metode perancangan yang digunakan untuk pengembangan sistem informasi farmasi adalah metode perancangan terstruktur (*Structured Design Method*) menggunakan *Data Flow Oriented Design*. *Data Flow Oriented Design* merupakan konsep perancangan termudah dengan penekanan pada sistem modular, *top down design* dan pemrograman terstruktur dengan menggunakan *Data Flow Diagram* (DFD).

Selain menggunakan *Data Flow Diagram*, pada perancangan sistem ini, menggunakan teknik-teknik normalisasi dan relasi tabel serta perancangan database. Mengenai metode dan teknik-teknik dalam pembuatan normalisasi, relasi tabel, dan perancangan *database*-nya telah dijelaskan di bab 2.

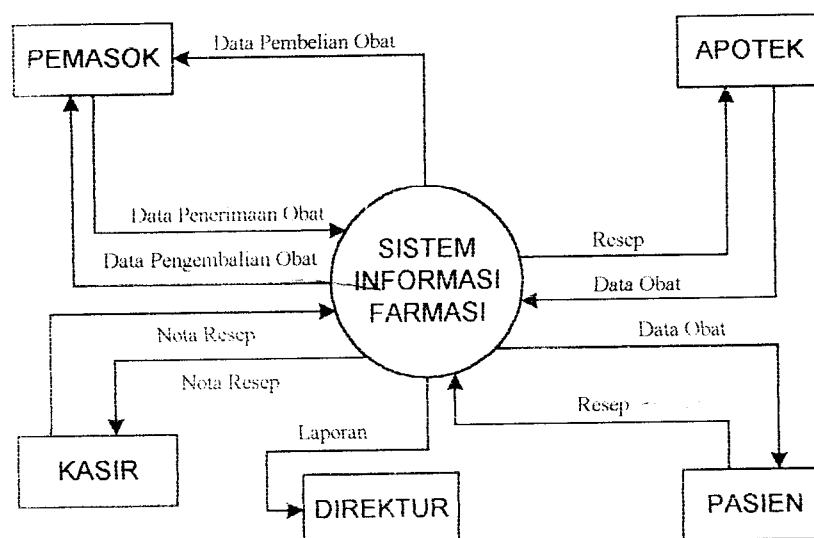
4.2 Hasil Perancangan

Hasil dari metode perancangan sistem yang digunakan dapat dibedakan menjadi beberapa bagian, yaitu:

4.2.1 Perancangan Sistem Secara Global dengan Menggunakan Diagram

Konteks

Diagram konteks ini proses aliran data dijabarkan secara global. Aliran data pada sistem informasi farmasi ini bersumber dari pemasok dan pasien dan selanjutnya data diolah dalam proses pengolahan data untuk menghasilkan informasi yang berupa laporan kepada direktur. Diagram konteksnya dapat dilihat pada gambar 4.1.



Gambar 4.1. Diagram Konteks Sistem Informasi Farmasi

4.2.2 Perancangan Model Proses dengan Data Flow Diagram (DFD)

Dalam *data flow diagram* level 1 sistem informasi farmasi yang ditunjukkan pada gambar 4.2 dibagi menjadi tiga proses yaitu :

1. Proses Pemasukkan Obat, berfungsi untuk melakukan proses pencatatan dan *update* data-data obat dari pemasok serta data-data persediaan obat dari apotek.

Inputnya : apotek dan pemasok.

Outputnya : data-data dari apotek dan data obat dari pemasok.

2. Proses Pengembalian Obat, berfungsi untuk melakukan proses pencatatan dan *update* data-data pengembalian obat ke pemasok

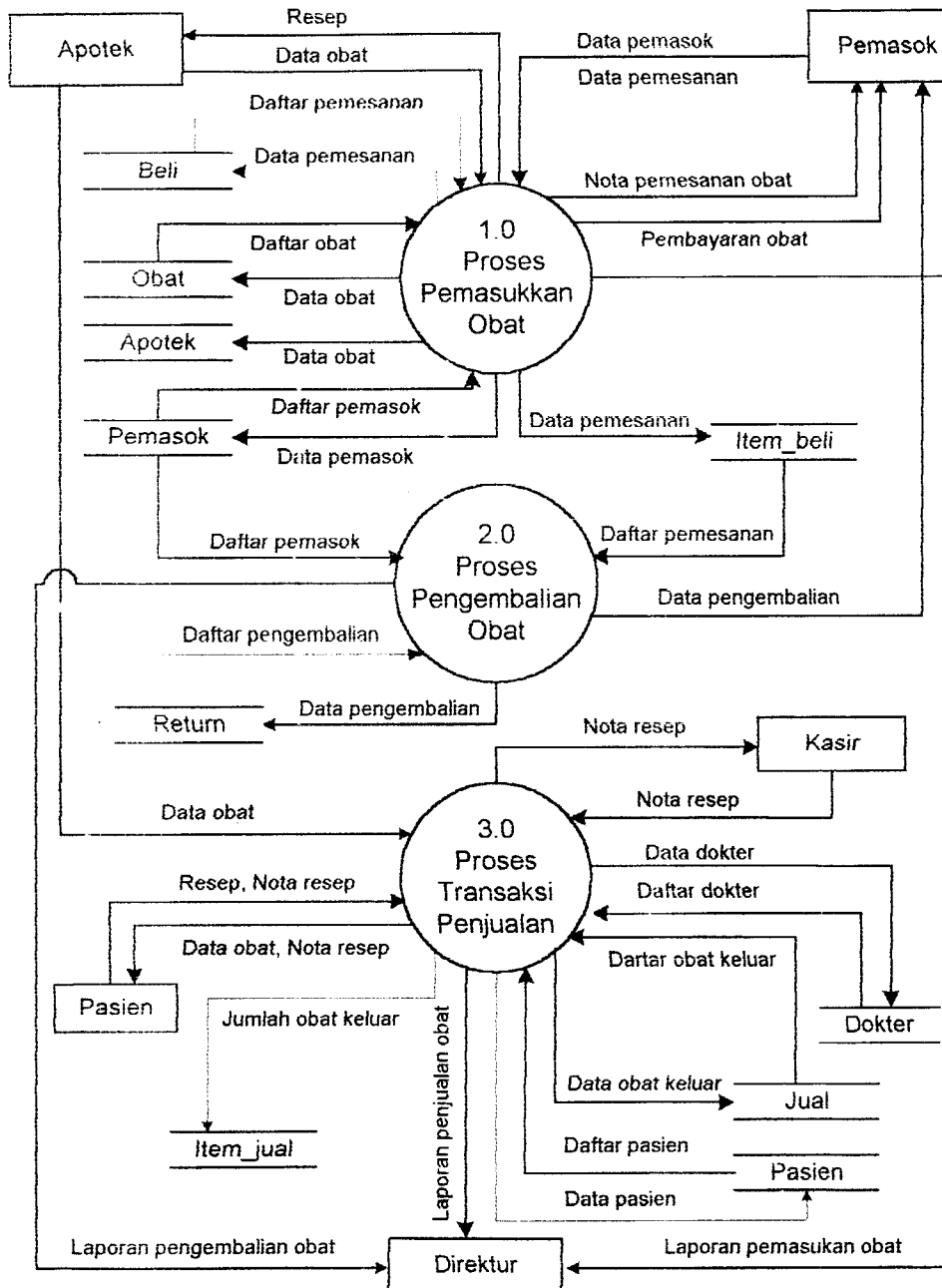
Inputnya : data-data obat dan pemasok.

Outputnya : data-data pengembalian obat.

3. Proses Penjualan obat, berfungsi untuk melakukan pencatatan dan *update* data-data penjualan obat ke pasien.

Inputnya : data-data obat dan resep.

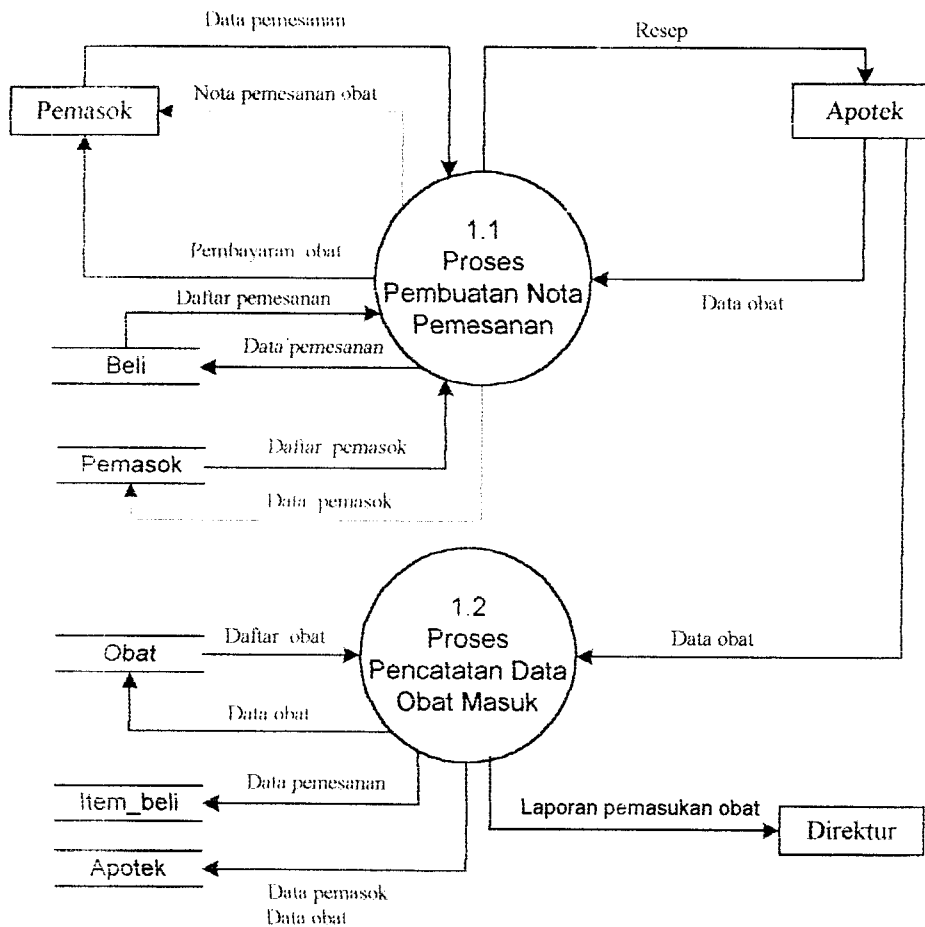
Outputnya : data-data obat, nota resep.



Gambar 4.2. DFD level 1 Sistem Informasi

4.2.2.1 DFD Level 2 Proses Pemasukkan Obat

Pada proses pemasukkan obat terbagi menjadi 2 proses yaitu : proses pembuatan nota pemesanan dan pencatatan data obat masuk, kedua proses ini dapat dilihat pada gambar 4.3.



Gambar 4.3. DFD Level 2 Proses Pemasukan

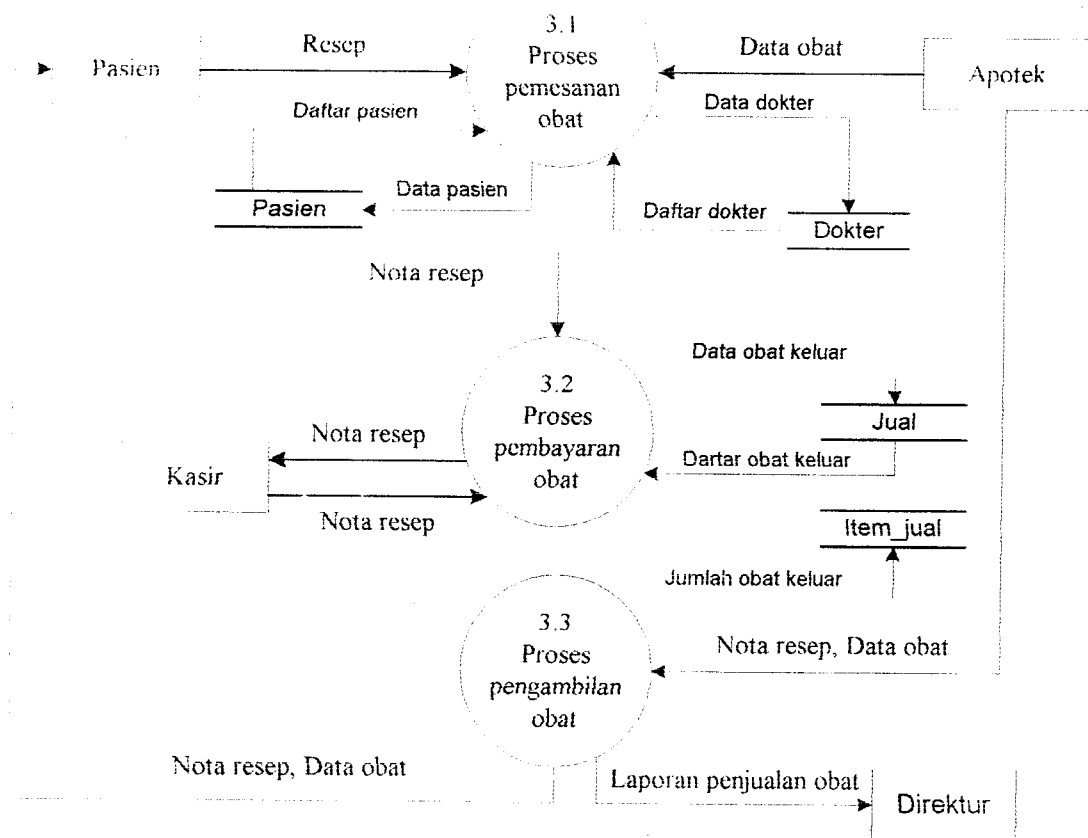
Keterangan :

1. Proses Pembuatan nota pemesanan (Proses 1.1) mendapat input dari apoteker dan bagian apotek berupa nota pemesanan obat dan data obat.

2. Proses pencatatan data obat masuk (Proses 1.2) mendapat input dari pemasok berupa data pemasok dan data obat.

4.2.2.1 DFD Level 2 Proses Transaksi Obat

Untuk proses transaksi obat dibagi dalam 3 proses yang terdiri dari proses pemesanan obat, proses pembayaran obat, dan proses pengambilan obat. Hasil dari rancangan ketiga proses ini dapat dilihat pada gambar 4.4.



Gambar 4.4. DFD Level 2 Proses Transaksi

Keterangan :

1. Proses pemesanan obat (Proses 3.1) mendapat input dari pasien berupa resep.
2. Proses pembayaran obat (Proses 3.2) mendapat input dari pasien berupa nota resep dan diolah sehingga menjadi nota resep lunas.
3. Proses pengambilan obat (Proses 3.3) mendapat input dari apotek berupa nota resep dan data obat.

4.3 Normalisasi

Normalisasi adalah suatu proses pengelompokkan data ke dalam sejumlah tabel yang mewakili sejumlah *entity* dan hubungan antar mereka. Bentuk-bentuk normalisasi data pada sistem informasi farmasi adalah sebagai berikut :

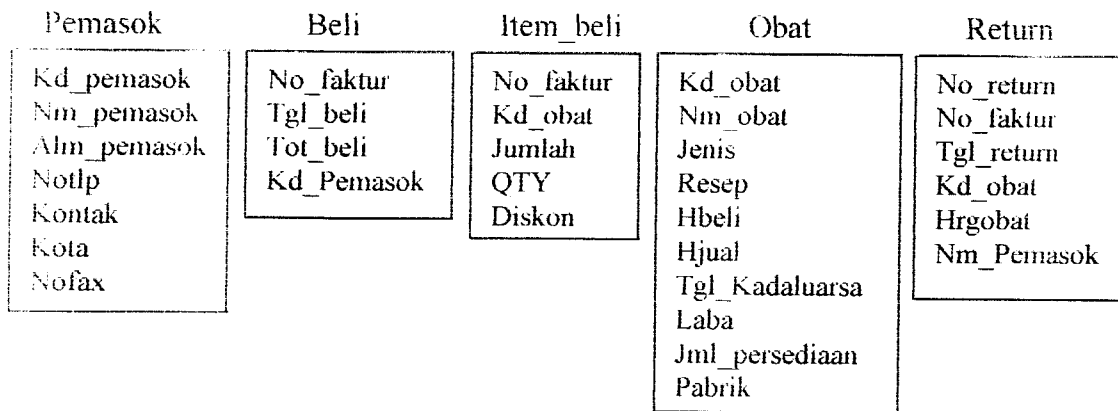
4.3.1 Normalisasi Data Pembelian Obat

Pada bentuk normal ke-1 semua data merupakan *atomic value*, artinya data dibentuk dalam *flat file* (file datar / rata) dalam satu *record* demi satu *record*, dan tidak ada set atribut yang berulang atau atribut yang bernilai ganda. Sebelum membuat bentuk normal ke-1 dari data pembelian obat, akan dirangkum atribut-atribut yang terjadi pada pembelian obat. Dari hasil hasil rangkuman atribut-atribut yang terlibat dalam pembelian obat nantinya akan dibentuk normal ke-1. Atribut-atribut yang didapatkan dapat dilihat pada gambar 4.5.

Kode Pemasok
Nama Pemasok
Alamat Pemasok
Kontak Person
Kota
No Telepone
No Fax
No Faktur
Kode Obat
Nama Obat
Jenis Obat
Jumlah Obat
Harga Pokok
Harga Jual
Tgl Terima
Tgl Tempo
Tgl Kadaluasa
No Return

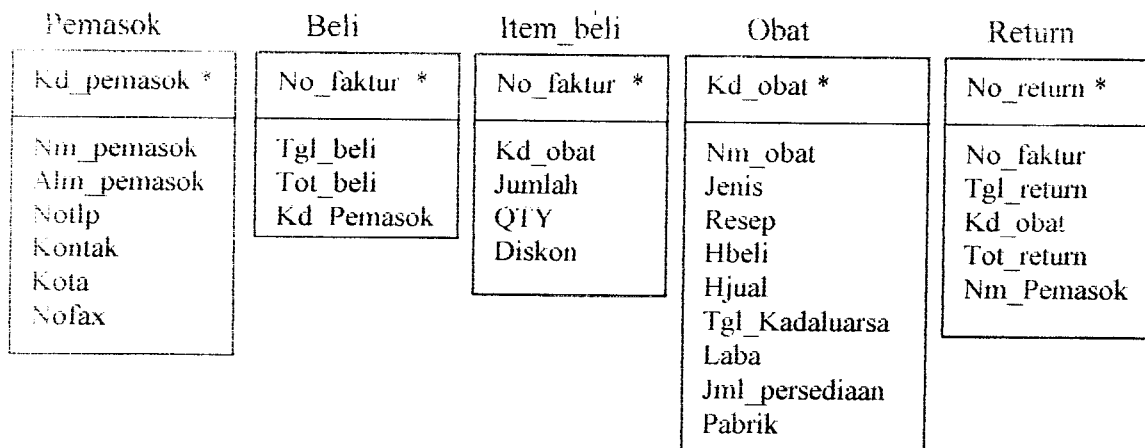
Gambar 4.5 Atribut-atribut hasil rangkuman data pembelian

Dari hasil rangkuman atribut-atribut di atas dapat dikomposisikan menjadi 5 tabel dengan atribut yang digunakan. Untuk membentuk normal ke-1 semua data merupakan *atomic value*, artinya data dibentuk dalam *flat file* (file datar / rata) dalam satu *record* demi satu *record*, dan tidak ada set atribut yang berulang atau atribut yang bernilai ganda. Bentuk normal ke-1 dapat dilihat pada gambar 4.6.



Gambar 4.6 Bentuk normal ke-1 pada data pembelian obat

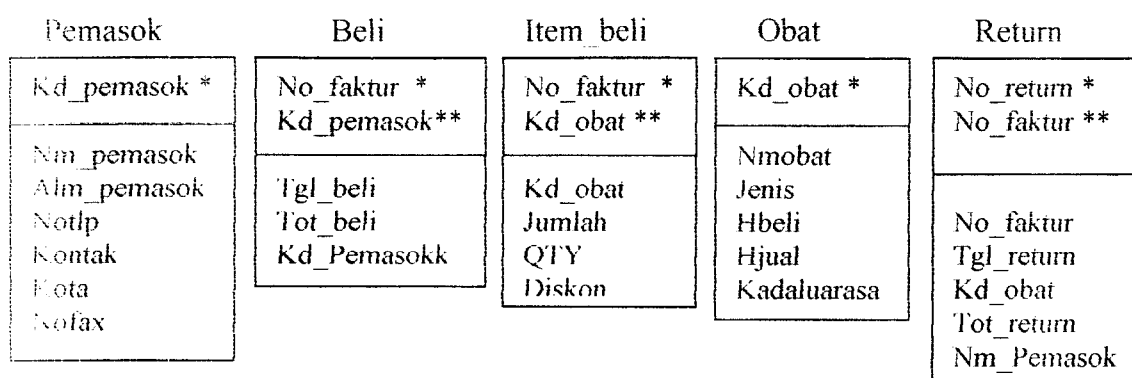
Pada bentuk normal ke-2 harus memenuhi bentuk normal ke-1 dan semua atribut yang bukan kunci memiliki ketergantungan fungsional pada kunci secara utuh. Bentuk dari normal ke-2 seperti pada gambar 4.7.



Gambar 4.7 Bentuk normal ke-2 pada data pembelian obat

Keterangan : * Kunci primer (*primary key*)

Pada normalisasi bentuk ke-3, harus memenuhi bentuk normal ke-2 dan tidak ditemukan adanya ketergantungan fungsional antara item-item data bukan kunci. Bentuk normal ke-3 menekankan relasi antara tabel dengan memberikan kunci tamu (*foreign key*) pada tabel yang berelasi untuk melengkapi satu hubungan yang menunjukkan ke induknya. Bentuk normal ke-3 seperti terlihat pada gambar 4.8.



Gambar 4.8 Bentuk normal ke-3 pada data pembelian

Keterangan : * Kunci primer (*primary key*)

** Kunci tamu (*foreign key*)

4.3.2 Normalisasi Data Penjualan Obat

Berikut ini sebuah tabel universal yang merupakan hasil rekonstruksi dari data penjualan. Dari hasil rangkuman atribut-atribut yang terlibat dalam penjualan obat nantinya akan dibentuk normal ke-1. Atribut-atribut yang didapatkan dapat dilihat pada gambar 4.9.

Kode Dokter
Nama Dokter
Kode Pasien
Nama Pasien
No Faktur
Kode Obat
Tanggal Jual
Harga
Jumlah

Gambar 4.9 Tabel universal pada data penjualan

Dari tabel universal di atas dapat dilakukan dekomposisi seperti pada data pembelian obat. Dekomposisi dari tabel universal di atas terbagi menjadi 4 buah tabel dengan atribut dengan atribut yang menyertainya. Seperti pada gambar 4.10.

Pasien	Jual	Item_jual	Dokter
Kode_pasien	No_faktur	No_faktur	Kd_dokter
Nm_pasien	Kd_dokter	Tgl_jual	Nm_dokter
Sex	Tgl_jual	Kd_obat	Spesialis
Alamat	Kd_pasien	QTY	Alamat
Notlp	Kd_dokter	Jlh_jual	No_tlp
Kota			

Gambar 4.10 Bentuk normal ke-1 data penjualan

Pada bentuk normal ke-2 harus memenuhi bentuk normal ke-1 dan semua atribut yang bukan kunci memiliki ketergantungan fungsional pada kunci secara utuh. Bentuk dari normal ke-2 seperti pada gambar 4.11.

Pasien	Jual	Item_jual	Dokter
Kode pasien *	No_faktur *	No_faktur *	Kd_dokter *
Nm_pasien Sex Alamat No_tlp Kota	Kd_dokter Tgl_jual Kd_pasien Kd_dokter	Tgl_jual Kd_obat QTY Jlh_jual	Nm_dokter Spesialis Alamat No_tlp

Gambar 4.11 Bentuk normal ke-2 data penjualan

Keterangan : * Kunci primer (*primary key*)

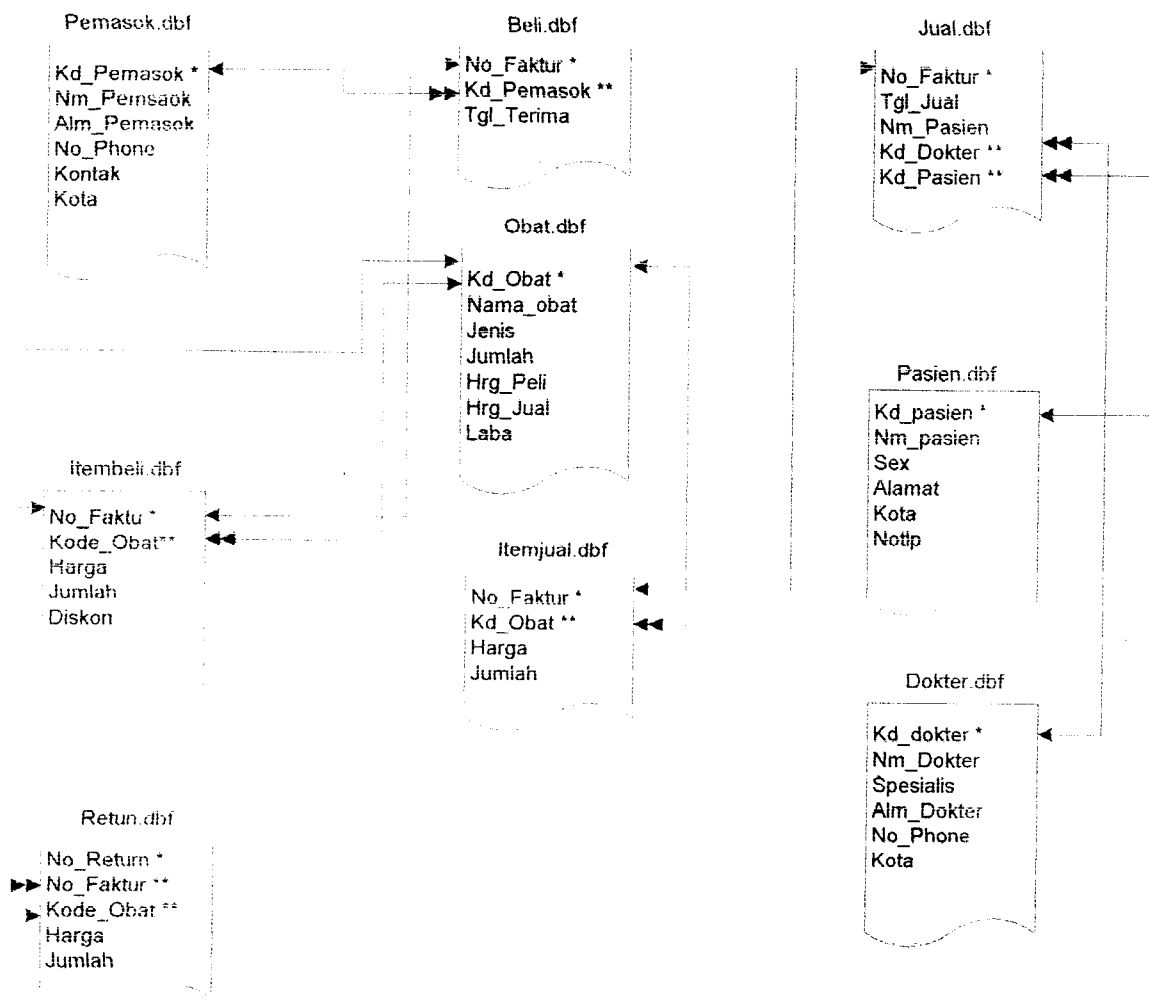
Pada normalisasi bentuk ke-3, harus memenuhi bentuk normal ke-2 dan tidak ditemukan adanya ketergantungan fungsional antara item-item data bukan kunci. Bentuk normal ke-3 menekankan relasi antara tabel dengan memberikan kunci tamu (*foreign key*) pada tabel yang berelasi untuk melengkapi satu hubungan yang menunjukkan ke induknya. Bentuk normal ke-3 seperti terlihat pada gambar 4.12.

Pasien	Jual	Item_jual	Dokter
Kode pasien *	No_faktur * Kd_dokter**	No_faktur * Kd_obat**	Kd_dokter * Nm_dokter**
Nm_pasien Sex Alamat No_tlp Kota	Tgl_jual Kd_pasien Kd_dokter	Tgl_jual Kd_obat QTY Jlh_jual	Spesialis Alamat No_tlp

Gambar 4.12 Bentuk normal ke-3 data penjualan

4.4 Relasi Tabel

Bentuk relasi antara tabel basis data di gambarkan seperti pada gambar 4.13.



Gambar 4.13 Relasi tabel sistem informasi farmasi

4.5 Perancangan Basis Data

Pengembangan Sistem Informasi Farmasi di RSUD Muhammadiyah Yogyakarta menggunakan 9 buah berkas yaitu sebagai berikut :

1. Berkas Pasien

Berkas pasien adalah tabel data untuk mencatat data semua pasien yang melakukan transaksi obat di bagian farmasi.

Nama Tabel 4.1. Pasien.dbf

No	Nama Field	Type	Lebar
1	KODE_PASIEN	Karakter	6
1	NM PASIEN	Karakter	25
2	SFX	Karakter	8
3	ALAMAT	Karakter	25
4	KOTA	Karakter	15
5	NO TLP	Karakter	12
Panjang record			91

2. Berkas Pemasok

Berkas pemasok adalah tabel untuk menyimpan data tiap-tiap pemasok yang mengirimkan obat ke bagian farmasi

Nama Tabel 4.2. Pemasok.dbf

No	Nama Field	Type	Lebar
1	KD_PEMASOK	Karakter	6
2	NM_PEMASOK	Karakter	25
3	ALM_PEMASOK	Karakter	30
5	NO_TELP	Numerik	12
6	KONTAK	Karakter	25
7	KOTA	Karakter	15
Panjang record			112

3. Berkas Obat

Berkas obat adalah tabel untuk mencatat data obat yang masuk, tanggal kadaluarsa obat dan mencatat jumlah persediaan.

Nama Tabel 4.3. Obat.dbf

No	Nama Field	Type	Lebar
1	KD_OBAT	Karakter	6
2	NM_OBAT	Karakter	15
3	JENIS	Karakter	8
4	RESEP	Karakter	10
5	TGL_KADALUARSA	Tanggal	8
6	HBELI	Numerik	9
7	HJUAL	Numerik	9
8	LABA	Numerik	9
9	JML_PERSEDIAAN	Numerik	5
10	PABRIK	Karakter	15
Panjang record			94

4. Berkas Beli

Berkas beli merupakan tabel untuk menyimpan data pembelian yang telah dilakukan dalam jangka waktu tertentu untuk menambah persediaan obat yang ada.

Nama Tabel 4.4. Beli.dbf

No	Nama Field	Type	Lebar
1	KD_BELI	Karakter	6
2	TGL_BELI	Tanggal	8
3	TOT_BELI	Numerik	9
4	KD PEMASOK	Karakter	6
Panjang record			29

5. Berkas Item Beli

Berkas item beli adalah tabel untuk menyimpan data obat-obat yang dibeli dalam satu faktur pembelian.

Nama Tabel 4.5. Itembeli.dbf

No	Nama Field	Type	Lebar
1	NO_FAKTUR	Karakter	6
2	KD_OBAT	Karakter	6
3	JUMLAH	Numerik	9
4	QTY	Numerik	4
5	DISKON	Numerik	5
Panjang record			30

6. Berkas Dokter

Berkas dokter adalah tabel untuk menyimpan data-data dokter.

Nama Tabel 4.6. Dokter.dbf

No	Nama Field	Type	Lebar
1	KD_DOKTER	Karakter	4
2	NM_DOKTER	Karakter	25
3	SPELIALIS	Karakter	15
4	ALAMAT	Karakter	30
5	NO_TELP	Numerik	8
Panjang record			82

7. Berkas Jual

Berkas jual adalah tabel untuk menyimpan data penjualan yang telah dilakukan pada jangka waktu tertentu.

Nama Tabel 4.7. Jual.dbf

No	Nama Field	Type	Lebar
1	NO_FAKTUR	Karakter	6
2	TGL_JUAL	Tanggal	8
3	TOTAL_JUAL	Numerik	9
4	KD_PASIEN	Karakter	6
5	KD_DOKTER	Karakter	6
Panjang record			35

8. Berkas Item Jual

Berkas item jual adalah tabel untuk menyimpan data penjualan yang dilakukan dalam satu faktur.

Nama Tabel 4.8. Itemjual.dbf

No	Nama Field	Type	Lebar
1	NO_FAKTUR	Karakter	6
2	TGL_JUAL	Tanggal	8
3	KD_OBAT	Karakter	6
4	QTY	Numerik	6
5	JLH_JUAL	Numerik	9
Panjang record			35

9. Berkas Return

Berkas return adalah tabel untuk menyimpan data pengembalian obat ke pemasok karena rusak/cacat.

Nama Tabel 4.9. Return.dbf

No	Nama Field	Type	Lebar
1	NO_RETURN	Karakter	6
2	NO_FAKTUR	Karakter	6
3	TGL_RETURN	Tanggal	8
4	TOT_RETURN	Numerik	9
5	KD_OBAT	Karakter	6
6	NM_PEMASOK	Karakter	25
Panjang record			60

4.6 Perancangan Menu

Menu dirancang dengan penyajian tombol-tombol yang dapat dipergunakan untuk melakukan berbagai kegiatan untuk masukan data, proses data dan keluaran data.

Untuk memilih menu ada dua cara :

1. Dengan menekan huruf tertentu.
2. Dengan menekan tombol menu.

Menu utama dalam program ini terdiri dari :

1. File

- Setup Obat

- Setup Pasien

- Setup Dokter

- Setup Pemasok

- Keluar

2. Transaksi

- Penjualan Obat

- Pembelian Obat

- Return Obat

3. Cari Data

- Cari Data Obat

- Cari Data Dokter

Cari Data Pemasok

4. Cetak

Cetak Obat

Cetak Pasien

Cetak Pemasok

Cetak Dokter

Cetak Laporan Pembelian

Cetak Laporan Penjualan

Cetak Laporan Return

5. Utilitas

Setup Password

Backup Data

Restore Data

Perawatan

6. Sekilas

Tentang

Bentuk dari tampilan menu utama dapat dilihat pada gambar 4.14.

Farmasi – RSU PKU Muhammdiyah Yogyakarta					
File	Transaksi	Cari	Cetak	Utilitas	Tentang

Gambar 4.14. Tampilan Menu Utama

4.7 Perancangan Output

Perancang output meliputi 7 laporan yang terdiri dari Cetak Obat, Cetak Pasien, Cetak Pemasok, Cetak Dokter, Cetak Laporan Pembelian, Cetak Laporan Penjualan, Cetak Laporan Return format laporannya adalah sebagai berikut :

Laporan Persediaan obat

RSU PKU Muhammdiyah Yogyakarta
 Jl. KH. Ahmad Dahlan No. 20 Yogyakarta 55122
 Tlp. (0274) 512653, 513871

Kode Obat	Nama Obat	Jenis Obat	Tgl. Kadaluarsa	Harga/satuan	Jumlah Pers

Laporan Pembelian Obat

RSU PKU Muhammadiyah Yogyakarta
 Jl. KH. Ahmad Dahlan No. 20 Yogyakarta 55122
 Tlp. (0274) 512653, 513871

Tgl. Terima	Nm Pemasok	Kode Obat	Nama Obat	Tgl. Kadaluarsa	Jml. Obat	Harga Obat
-------------	------------	-----------	-----------	-----------------	-----------	------------

Laporan Penjualan

RSU PKU Muhammadiyah Yogyakarta
 Jl. KH. Ahmad Dahlan No. 20 Yogyakarta 55122
 Tlp. (0274) 512653, 513871

Kode Obat	Nama Obat	Tgl Kadaluarsa	Jml. Obat	Hrg Obat	Tgl Keluar
-----------	-----------	----------------	-----------	----------	------------

Laporan Return

RSU PKU Muhammadiyah Yogyakarta
 Jl. KH. Ahmad Dahlan No. 20 Yogyakarta 55122
 Tlp. (0274) 512653, 513871

Kd Obat	Nanam Obat	QTY	Harga	Jumlah
---------	------------	-----	-------	--------

BAB V

IMPLEMENTASI PERANGKAT LUNAK

5.1 Batasan Implementasi

Implementasi secara umum merupakan tahapan dimana sistem siap dibangun dengan bahasa pemrograman tertentu, sehingga diketahui apakah sistem yang dibuat benar-benar dapat menghasilkan tujuan yang diinginkan.

Implementasi sistem yang dibuat di tugas akhir ini adalah sederhana, dalam hal ini lebih ditekankan pada alur proses kegiatan yang terjadi di farmasi yaitu alur proses pemasukkan atau pembelian obat serta penjualan obat ke pasien yang disertai dengan dokumentasi. Bahasa pemrograman yang dipilih untuk menjalankan program tersebut yaitu visual FoxPro versi 6.0.

5.2 Alasan Pemilihan Aplikasi Pengembangan

Perangkat lunak yang dipakai yaitu FoxPro versi 6.0. Pemilihan perangkat lunak yang digunakan didasarkan atas pertimbangan dan alasan antara lain :

1. Visual FoxPro merupakan perangkat lunak untuk pengembangan sistem yang berbasis pada sistem operasi windows dan mudah digunakan karena bersifat *visual programming language* yang berarti dapat dilakukan desain program secara visual.
2. Memiliki kemampuan yang cukup baik untuk mengembangkan sistem meminimalkan pengkodean.

3. Bahasa pemrograman ini berbasis pada *database programming*, sehingga sudah cukup akrab bagi programmer.

5.3 Keamanan Sistem

Sistem informasi *multi user* memerlukan adanya sistem keamanan (*security*) yang baik. Sistem keamanan yang digunakan antara lain :

1. Hak Akses

Berfungsi untuk memberikan hak akses pada user, sesuai dengan unit yang ditempati. User yang tidak berhak mengakses suatu data tertentu tidak diberikan hak untuk mengaksesnya.

2. User Login

Sistem login menyaring user sesuai dengan nama user dan *password* yang dimiliki. User yang tidak memiliki *password* sendiri aksesnya akan ditolak (tidak dapat masuk atau menggunakan sistem informasi ini).

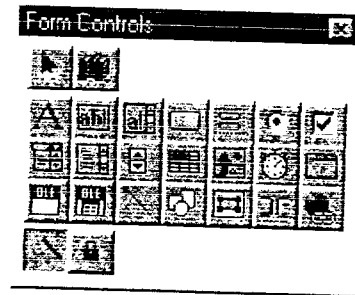
5.4 Fasilitas Pengembangan

Untuk menyusun aplikasi Sistem Informasi Farmasi (SIF) dibutuhkan objek-objek utama yang terdiri dari :

1. *Folder*, atau disebut juga dengan direktori, merupakan sebuah lokasi penyimpanan dalam media simpan yang digunakan untuk menampung *file-file* objek yang ada pada aplikasi sistem informasi farmasi. Sebelum menyusun objek-objek lain seperti *database*, tabel, *form*, *report* dan yang lainnya, maka harus membentuk atau menyusun *folder* ini.








2. *Database*, digunakan untuk mencatat dan menampung data-data yang ada pada aplikasi sistem informasi farmasi yang akan disusun nanti. Dalam aplikasi sistem informasi farmasi hanya membutuhkan satu *database* dengan nama apotek, yang di dalamnya terdapat beberapa tabel.
3. *Table*, merupakan *file* dengan ekstensi .DBF, yang digunakan untuk menyimpan data. Tabel-tabel yang ada pada sistem informasi farmasi telah dijelaskan pada bab 4.
4. *Form*, merupakan salah satu objek yang digunakan untuk mengelola data tabel. Dengan *form* ini dapat menampilkan, mengedit, menambah data *record* yang ada dalam sebuah tabel.
5. *Report*, merupakan format objek yang digunakan untuk mengetahui informasi-informasi yang dibutuhkan.
6. *Menu*, digunakan untuk mengintegrasikan objek-objek yang sudah dibentuk sebelumnya, sehingga dapat dijalankan secara terintegrasi.
7. *Project*, merupakan langkah awal untuk membentuk sebuah aplikasi. Jadi aplikasi belum dapat dibentuk sebelum menyusun *project* ini.










Pada sub bab fasilitas pengembangan ini akan dijelaskan, macam-macam dan bentuk kontrol *form* yang membangun *form-form* aplikasi sistem informasi farmasi, seperti pada gambar 5.1.



Gambar 5.1 *Toolbar form control*

Fungsi dan nama dari objek-objek yang terdapat pada *form control*.

-  *Select Object* digunakan untuk mengubah dan memindahkan kontrol
-  *Label* digunakan untuk mendefinisikan teks yang tidak dapat diubah oleh *user*, misalnya judul *form*, grafik
-  *Text Box*, dipergunakan untuk memasukkan satu baris teks yang dapat diubah dan dimasukkan oleh *user*.
-  *Command Button* digunakan untuk mengaktifkan perintah yang didefinisikan misalnya tombol keluar untuk menutup *form*.
-  *Command Group* digunakan untuk menghubungkan tombol perintah yang satu dengan yang lain.
-  *Option Group* digunakan untuk menampilkan beberapa pilihan, tetapi hanya dapat memilih salah satu diantara beberapa pilihan.
-  *Combo Box* digunakan untuk menciptakan kotak *combo drop down* atau kotak daftar *drop down*, sehingga *user* dapat memilih satu *item* atau memasukkan nilai.

-  *Edit Box* digunakan untuk memasukkan beberapa baris teks, dapat diubah dan dimasukkan oleh *user*.
-  *Spinner* berfungsi untuk memasukan nilai angka dengan sebuah *range* yang sudah ditentukan.
-  *Grid* digunakan untuk menampilkan *grid* yang ditampilkan dalam *spreadsheet*.
-  *Image* (kotak gambar) dipakai untuk menampilkan sebuah gambar atau grafik dalam sebuah *form* agar kelihatan menarik.
-  *Timer* mendefinisikan kontrol waktu, dimana kita dapat memproses khusus yang berkaitan dengan waktu dan intervalnya.
-  *Page Frame* digunakan untuk membuat halaman bertingkat seperti tapak pada setiap pemasukkan data pada aplikasi ini.
-  *Line/garis* digunakan untuk membuat halaman bertingkat seperti tapak pada setiap pemasukkan data pada aplikasi ini.
-  *Shape* digunakan untuk membuat gambar kota baik di warna atau dibuat transparan.
-  *Container* mengambil *space* antara kontrol *toolbar*.

5.5 Implementasi Sistem Informasi Farmasi

Aplikasi sistem informasi farmasi mempunyai sebuah menu utama yang di dalamnya terdapat beberapa menu *bar*. Pada setiap menu *bar* yang ada pada menu

utama tersebut mempunyai beberapa pilihan submenu yang jika dipilih akan melakukan suatu proses. Implementasi dalam sistem informasi farmasi yang dibahas dalam bab ini terdiri dari bagaimana pembuatan *form* pendataan, pembuatan *form* laporan dan pembuatan menu aplikasi.

Sistem informasi farmasi ini terdapat 7 buah *form* utama pendataan, 4 buah *form* bantu, 4 buah *form* utility. *Form* utama terdiri dari *form* obat (*fobat.scx*), *form* pemasok (*fdistrib.scx*), *form* dokter (*fdokter.scx*), *form* pasien (*fpasien.scx*), *form* pembelian obat (*fbeli.scx*), *form* penjualan obat (*fjual.scx*), *form* return obat (*freturn.scx*), dan 4 buah *form* bantu yang terdiri dari *form* cari dokter (*cari_dokter.scx*), *form* cari obat (*cari_obat.scx*), *form* cari pasien (*cari_pasien.scx*), *form* cari pemasok (*cari_pemasok.scx*). Sedangkan untuk laporan terdiri dari 7 buah, yaitu laporan persediaan obat, laporan penjualan, laporan pembelian, laporan pengembalian, daftar dokter, daftar pasien, daftar pemasok.

5.4.1 Implementasi Form Obat

Form obat (*fobat.scx*) berfungsi untuk mencatat atau menyimpan semua data obat yang masuk maupun yang keluar. Berkas pendataan obat tersimpan dalam *database* *obat.dbf*. Tampilan *form* pendataan obat dapat dilihat pada gambar 5.2.

Pemasukan Data Obat		Daftar Data Obat	
Kode Obat	KO.001	Harga Beli	250
Nama Obat	Panadol	Harga Jual	260
Jenis	Tablet	Laba Obat	10
Resep	Non Resep	Stok	0
Pabrik	Kimia Farma	Kadaluarsa	12-12-2002


Gambar 5.2 Form obat *page frame* pemasukkan data obat


Langkah-langkah pembuatan *form* obat *page frame* pemasukan data obat yaitu :

- a. Mengaktifkan visual FoxPro.
- b. Membuat *form* kosong, dengan cara sebagai berikut :
 - 1) Klik menu *File*, dan pilih *New*, kemudian dalam kotak pilihan pilih *form* dan klik *New File*.
 - 2) Memberi judul *form*, aktifkan *properties*, kemudian klik pada *caption* ketik dengan nama **Obat**.
- c. Membuat *Page Frame*


Pemasukan Data Obat	Daftar Data Obat
---------------------	------------------

 - 1) Klik objek *page frame* pada *form control*, kemudian drag ke *form*. Untuk membuat judul pada *page frame* mengaktifkan *properties*, kemudian memilih **Page1** dan pilih *caption* diisi dengan **Pemasukan Data Obat**.

- 2) Untuk membuat judul **Daftar Obat**, memilih **Page2** dan pilih *caption* diisi dengan **Daftar Obat**.
- d. Membuat tulisan (*label*) **Kode Obat** pada *page frame*.
- 1) Klik objek *label* pada *form control*, kemudian drag ke *page frame*, sehingga akan muncul **label1**. Untuk merubah label tersebut, aktifkan *properties* kemudian pilih *caption* ketik dengan **Kode Obat**.
 - 2) Jika ingin menebalkan tulisan aktifkan *properties* dan pilih *FontBold* kemudian mengganti isinya dengan **T-True**.
 - 3) Untuk membuat label-label yang lainnya dapat mengulangi langkah-langkah di atas.
- e. Membuat *TextBox*  pada *page frame*.
- 1) Klik objek *TextBox* pada *form control*, kemudian drag ke *page frame*, sehingga akan muncul **text1**. Untuk merubah text tersebut, aktifkan *properties* kemudian pilih *name* ketik dengan **Kd_obat**.
 - 2) Jika ingin menebalkan tulisan aktifkan *properties* dan pilih *FontBold* kemudian mengganti isinya dengan **T-True**.
 - 3) Untuk membuat text-text yang lainnya dapat mengulangi langkah-langkah di atas.

f. Membuat *ComboBox* 

- 1) Klik *combobox* pada *form control*, kemudian drag ke *frame page (page1)*.
- 2) Aktifkan *builder* dengan mengklik kanan pada *mouse*, pilih menu 1 *List Item*, klik pada *Fill the List With* kemudian pilih data *Entered By Hand*.
- 3) Mengetik pada kolom nama-nama yang akan dimasukkan ke dalam *combobox*.

g. Membuat *CommandButton* 

- 1) Klik objek *command button* pada *form control*, kemudian drag ke *page frame*.
- 2) Aktifkan *properties* kemudian pilih *name* diganti dengan **none**, dan *picture* diisi dengan gambar yang diinginkan.
- 3) Untuk membuat *command button* yang lainnya dapat mengulangi langkah 1 dan 2.

Pada *form* pendataan obat terdapat juga program bantu yang berfungsi untuk melihat daftar obat yang telah dimasukkan, seperti terlihat pada gambar 5.3.

Kd Obat	Nama Obat	Jenis	Resep	Pabrik	Harga Beli
KE-001	Panadol	Tablet	Non Resep	Kimia Farma	250

Gambar 5.3 Form obat *page frame* daftar data obat

Langkah-langkah dalam pembuatan *page frame* daftar data obat adalah sebagai berikut :

- a. Mengaktifkan *Page Frame* daftar data obat.
- b. Membuat *Grid*


Kd Obat	Nama Obat
000003	Paramex
000004	Napasin
000005	Balsem
000006	Bodrex

- 1) Mengaktifkan *page2*, klik *grid* pada *form control*, kemudian *drag* ke *frame page*.
- 2) Mengaktifkan *builder* dengan mengklik kanan pada *mouse*, pilih menu 1 *grid item*, kemudian memilih *field* yang akan dimasukkan.
- 3) Mengatur *DeleteMark* dan *RecordMark* pada *properties* menjadi **F-False**.

Prosedur-prosedur yang terdapat pada *form* obat adalah :


Prosedur untuk melihat awal data.

```


    go top
    wait window"Awal Record" nowait
    thisform.refresh()
  
```


Prosedur untuk melihat awal data.

```


    go bottom
    wait window"Akhir Record" nowait
    thisform.refresh()
  
```


Prosedur untuk melihat data sebelum.

```


    if NOT BOF()
      skip -1
    endif
    if BOF()
      wait window"Awal Record" nowait
    endif
    thisform.refresh()
  
```


Prosedur untuk melihat data berikut.

```


    if NOT EOF()
      skip
    endif
    if EOF()
      wait window"Akhir Record" nowait
      skip-1
    endif
    thisform.refresh()
  
```

Prosedur untuk memasukkan data baru.

```


    Locate For Empty(Kode_obat) &&
    If !Found() &&
      Append Blank &&
      Replace Stok_obat With 0
    Endif &&
    Thisform.Refresh() &&
  
```

Prosedur untuk menghapus data.



```
pesan=messagebox("Apakah data akan dihapus ?",4+32,"Perhatian")
if pesan=6
    delete
    thisform.refresh()
endif
go top
thisform.show()
```

Prosedur untuk mengkoreksi data yang dimasukkan.



```
thisform.frame1.page1.txtkode.setfocus()
```

Prosedur untuk menyimpan data.



```
tabelupdate()
thisform.refresh()
```

Prosedur untuk membatalkan pengisian data.



```
tabelrevert()
thisform.refresh()
go top
```

Prosedur untuk keluar dari *form* pendataan obat.



```
thisform.release()
thisform.refresh()
```

5.4.2 Implementasi Form Data Pemasok

Form data pemasok (**Fdistrib,scx**) berfungsi untuk mencatat atau menyimpan semua data pemasok. Berkas pemasok tersimpan dalam *database* *distrib.dbf*. Tampilan *form* data pemasok *page frame* pendataan pemasok dapat dilihat pada gambar 5.4.

The screenshot shows a software window titled "Pemasok". Inside, there is a form titled "Pemasukan Data Pemasok" and a section titled "Daftar Pemasok". The form contains the following data:

No. Pemasok	PS001
Nama Pemasok	PT. KALBE FARMA
Alamat	Jl. Imam Bonjol No. 12
Kota	SEMARANG
No. Telpon	(0274) 586945
Kontak	ALIF GUFRON

At the bottom of the window is a toolbar with the following buttons: a set of four navigation arrows (back, left, right, forward), "Baru" (New), "Simpan" (Save), "Batal" (Cancel), "Koreksi" (Correct), "Hapus" (Delete), and "Keluar" (Exit).

Gambar 5.4 Form data pemasok *page frame* memasukkan data pemasok

Langkah-langkah pembuatan *form page frame* memasukkan data pemasok telah diterangkan pada sub bab 5.4.1, yaitu mengenai pembuatan *label*, *text*, dan *command button*.

Pada *form* data pemasok terdapat juga program bantu yang berfungsi untuk melihat daftar pemasok yang telah dimasukkan, seperti terlihat pada gambar 5.5.

No. Pemasok	Nama Pemasok	Alamat Pemasok	Kota
PS001	PT. KALBE FARMA	Jl. Imam Bonjol No. 12	SEMARANG

Gambar 5.5 Form data pemasok *page frame* daftar pemasok

Langkah-langkah pembuatan *form page frame* daftar pemasok telah diterangkan pada sub bab 5.4.1, yaitu mengenai pembuatan *grid* dan *page frame*.

5.4.3 Implementasi Form Data Dokter

Form dokter (*Fdokter,scx*) berfungsi untuk mencatat atau menyimpan semua data dokter. Berkas dokter tersimpan dalam *database* dokter.dbf. Tampilan *form* dokter *page frame* pendataan dokter seperti pada gambar 5.6

Pemasukan Data Dokter		Daftar Dokter
Kode Dokter	DK001	
Nama Dokter	dr. DIDI DEVITO, Phd.	
Spesialisasi	Kandungan	
Alamat	Jl. Kalurang KM. 14 Tegal Sar	
Kota	Sleman	
Telpon	896119	

Toolbar: Baru, Simpan, Batal, Koreksi, Hapus, Keluar

Gambar 5.6 Form data dokter *page frame* pendataan dokter

Langkah-langkah pembuatan *form page frame* pemasukkan data dokter telah diterangkan pada sub bab 5.4.1, yaitu mengenai pembuatan *label, text, dan command button*.

Pada *form* data dokter terdapat juga program bantu yang berfungsi untuk melihat daftar dokter yang telah dimasukkan, seperti terlihat pada gambar 5.7.

Kd Dokter	Nama Dokter	Alamat	Kota
D9.001	dr. DIDI DEVITO, Phd.	Jl. Kalurang KM. 14 Tegal Sar	Sleman

Keluar

Gambar 5.7 Form data dokter *page frame* daftar dokter

Langkah-langkah pembuatan *form page frame* daftar dokter telah diterangkan pada sub bab implementasi *form* obat, yaitu mengenai pembuatan *grid* dan *page frame*.

5.4.4 Implementasi Form Data Pasien

Form data pasien (**Fpasien.scx**) berfungsi untuk mencatat atau menyimpan semua data pasien. Berkas pasien tersimpan dalam *database* pasien.dbf. Tampilan *form* data pasien *page frame* pendataan pasien seperti pada gambar 5.8.

The image shows a software window titled "Pasien" with a tabbed interface. The active tab is "Pendaftaran Data Pasien". The form contains the following data:

Kode Pasien	P5001
Nama Pasien	Paimin Bartes
Sex	Laki-laki
Usia	25
Alamat	Jl. Wonosari Km. 20 Gasakan No
Kota	Wonosari
No. Telepon	54757

At the bottom of the window is a toolbar with the following buttons: "Batal", "Simpan", "Batal", "Koreksi", "Koreksi", and "Keluar".

Gambar 5.8 Form data pasien *page frame* pendataan pasien

Langkah-langkah pembuatan *form page frame* memasukkan data pasien telah diterangkan pada sub bab 5.4.1, yaitu mengenai pembuatan *label*, *text*, dan *command button*.

Pada *form* data pasien terdapat juga *page frame* daftar pasien yang berfungsi untuk melihat daftar pasien yang telah dimasukkan, seperti terlihat pada gambar 5.9.

No. R.M.	Nama Pasien	Sex	Usia	Alamat
P0101	Paimin Bartes	Laki-laki	25	Jl. Wonosari Km. 20 Gasakar

Gambar 5.9 Form data pasien *page frame* daftar pasien

Langkah-langkah pembuatan *form page frame* daftar pasien telah diterangkan pada sub bab 5.4.1, yaitu mengenai pembuatan *grid* dan *page frame*.

5.4.5 Implementasi Form Cari Obat

Form cari obat (**Fcari_obat.scx**), *form* ini berfungsi untuk membantu dalam pencarian data-data obat yang telah dimasukkan ke *database* obat.dbf, bentuk *form*-nya dapat dilihat pada gambar 5.10.

Pencarian Data Obat

Masukan Kode Obat Masukan Nama Obat

Kode Obat	Nama Obat	Bentuk	Harga Beli	Harga Jual	Stok	Kadaluarsa
K01001	Panadol	Tablet	250	260	0	12-12-2002

Gambar 5.10 Form cari obat

Langkah-langkah pembuatan *form* cari obat adalah sebagai berikut :

- a. Mengaktifkan visual FoxPro.
- b. Membuat *form* kosong, dengan cara sebagai berikut :
 - 1) Klik menu *File*, dan pilih *New*, kemudian dalam kotak pilihan pilih *form* dan klik *New File*.
 - 2) Memberi judul *form*, aktifkan *properties*, kemudian klik pada *caption* ketik dengan nama **Cari Obat**.
- c. Membuat *check box* Masukan Kode Obat adalah sebagai berikut :
 - 1) Klik objek *check box* pada *form control*, kemudian drag ke *form*.

- 2) Mengaktifkan *properties* pilih *caption*, kemudian mengetikkan **Masukan Kode Obat**
 - 3) Membuat tulisan menjadi tebal pilih *FontBold*
 - 4) Untuk membuat *checkbox*, masukan Nama Obat ulangi langkah 1,2 dan 3.
- d. Membuat *label*, *grid* dan *commonbotton* telah dijelaskan pada sub bab implementasi *form* obat.

Prosedur-prosedur yang ada pada *form* pencarian data obat adalah prosedur untuk memilih kriteria pencarian baik itu berdasarkan kode obat maupun nama obat dan prosedur untuk melakukan pencarian data obat.

```

**PRESEDUR PILIHAN BERDASARKAN KODE OBAT
if thisform.cek1.value=1
thisform.txtkd_obat.enabled=.t.
  thisform.txtkd_obat.setfocus()
else
thisform.txtkd_obat.enabled=.f.
endif

```

```

**PRESEDUR PILIHAN BERDASARKAN NAMA OBAT
if thisform.cek2.value=1
thisform.txtnm_obat.enabled=.t.
thisform.txtnm_obat.setfocus()
else
thisform.txtnm_obat.enabled=.f.
endif

```

```

**PROSEDUR CARI DATA OBAT BERDASARKAN KODE DAN NAMA
nama_cari=thisform.txtnm_obat.value
no_cari=thisform.txtkode.value
if thisform.cek1.value=0 and thisform.cek2.value=0
  MessageBox("Masukan nomor atau nama obat !!!",48,"Perhatian")
endif
if thisform.cek1.value=1 and thisform.cek2.value=0
  sele obat
  set filter to upper(kd_obat)=upper(RTRIM(no_cari)) and !deleted()
  thisform.refresh()
endif
if thisform.cek1.value=0 and thisform.cek2.value=1
  sele obat

```

```

set filter to upper(nm_obat)=upper(RTRIM(nama_cari)) and !deleted()
thisform.refresh()
endif
if thisform.cek1.value=1 and thisform.cek2.value=1
sele obat
set filter to upper(kd_obat)=upper(RTRIM(no_cari)) and;
upper(nm_obat)=upper(RTRIM(nama_cari)) and;
!deleted()
thisform.refresh()
endif

```

5.4.6 Implementasi Form Cari Pemasok

Form cari pemasok (**Fcari_pemasok.scx**), *form* ini berfungsi untuk membantu dalam pencarian data-data pemasok yang telah dimasukan ke *database* *distrib.dbf*, bentuk *form*-nya dapat dilihat pada gambar 5.11.

Cari Data Pemasok

Pencarian Data Pemasok

Masukan Kode Pemasok Masukan Nama Pemasok

Kd Pemasok	Nama Pemasok	Alamat	Kota	Kontak Person
P0001	PT. KALBE FARMA	Jl. Imam Bonjol No. 12	SEMARANG	(0274) 565945

Cari Data Keluar

Gambar 5.11 Form cari pemasok

Langkah-langkah implementasi *form* cari pemasok telah dijelaskan pada sub bab 5.4.5, yaitu mengenai pembuatan *label*, *command button*, *check box*, *grid* dan *text*.

Prosedur yang terdapat pada implementasi *form* cari pemasok adalah prosedur untuk memilih kriteria pencarian baik itu berdasarkan kode pemasok maupun nama pemasok dan prosedur untuk melakukan pencarian data pemasok.

****PRESEDUR PILIHAN BERDASARKAN KODE PEMASOK**

```
if thisform.cek1.value=1
  thisform.txtkode_dis.enabled=.t.
  thisform.txtkode_dis.setfocus()
else
  thisform.txtkode_dis.enabled=.f.
endif
```

****PRESEDUR PILIHAN BERDASARKAN NAMA PEMASOK**

```
if thisform.cek2.value=1
  thisform.txtnama_dis.enabled=.t.
  thisform.txtnama_dis.setfocus()
else
  thisform.txtnama_dis.enabled=.f.
endif
```

****PROSEDUR CARI DATA PEMASOK BERDASARKAN KODE DAN NAMA**

```
nama_cari=thisform.txtnama_dis.value
no_cari=thisform.txtKode_dis.value
if thisform.cek1.value=0 and thisform.cek2.value=0
  MessageBox("Masukan Kode Pemasok atau Nama Pemasok !!!",48,"Perhatian")
endif
if thisform.cek1.value=1 and thisform.cek2.value=0
  sele distrib
  set filter to upper(Kode_dis)=upper(RTRIM(no_cari)) and !deleted()
  thisform.refresh()
endif
if thisform.cek1.value=0 and thisform.cek2.value=1
  sele distrib
  thisform.refresh()
endif
if thisform.cek1.value=1 and thisform.cek2.value=1
  sele distrib
  set filter to upper(Kode_dis)=upper(RTRIM(no_cari)) and;
  !deleted()
endif
```

5.4.7 Implementasi Form Cari Dokter

Form cari dokter (**Fcari_dok.scx**), form ini berfungsi untuk membantu dalam pencarian data-data dokter yang telah dimasukkan ke *database* dokter.dbf, bentuk form-nya dapat dilihat pada gambar 5.12.

Pencarian Data Dokter

Masukan Kode Dokter Masukan Nama Dokter

Kd Dokter	Nama Dokter	Spesialis	Alamat	No Telepon
Dk001	dr. DIDI DEWITO, PhD	Kardiologi	Jl. Kalsariaj KM. 14 Tegol Sari	090119

Cari Data Keluar

Gambar 5.12 Form cari dokter

Langkah-langkah implementasi *form* cari dokter telah dijelaskan pada sub bab 5.4.5, yaitu mengenai pembuatan *label*, *command button*, *check box*, *grid* dan *text*.

Prosedur yang terdapat pada implementasi *form* cari dokter adalah prosedur untuk memilih kriteria pencarian baik itu berdasarkan kode dokter maupun nama dokter dan prosedur untuk melakukan pencarian data dokter.

```

**PRESEDUR PILIHAN BERDASARKAN KODE DOKTER
if thisform.cek1.value=1
    thisform.txtkod_dokter.enabled=.t.
    thisform.txtkod_dokter.setfocus()
else
    thisform.txtkod_dokter.enabled=.f.
endif

**PRESEDUR PILIHAN BERDASARKAN NAMA DOKTER
if thisform.cek2.value=1
    thisform.txtnm_dokter.enabled=.t.
    thisform.txtnm_dokter.setfocus()
else
    thisform.txtnm_dokter.enabled=.f.
endif

**PROSEDUR CARI DATA DOKTER BERDASARKAN KODE DAN NAMA
nama_cari=thisform.txtnm_dokter.value
no_cari=thisform.txtKod_dokter.value
if thisform.cek1.value=0 and thisform.cek2.value=0
    Messagebox("Masukan Kode Dokter atau Nama Dokter !!!",48,"Perhatian")
endif
if thisform.cek1.value=1 and thisform.cek2.value=0
    sele dokter
    set filter to upper(Kod_dokter)=upper(RTRIM(no_cari)) and !deleted()
    thisform.refresh()
endif
if thisform.cek1.value=0 and thisform.cek2.value=1
    sele dokter
    set filter to upper(nm_dokter)=upper(RTRIM(nama_cari)) and !deleted()
    thisform.refresh()
endif
if thisform.cek1.value=1 and thisform.cek2.value=1
    sele dokter
    set filter to upper(Kod_dokter)=upper(RTRIM(no_cari)) and;
    upper(nm_dokter)=upper(RTRIM(nama_cari)) and;
    !deleted()
    thisform.refresh()
endif

```

5.4.8 Implementasi Form Pembelian Obat

Form pembelian obat (**Fbeli.scx**), *form* ini berfungsi untuk melakukan transaksi pembelian obat ke pemasok. Berkas pembelian disimpan ke *database* *beli.dbf* dan *itembeli.dbf*, bentuk *form*-nya dapat dilihat pada gambar 5.13.

5.4.9 Implementasi Nota Penjualan

Form nota penjualan (**Fjual.scx**), form ini berfungsi untuk melakukan transaksi penjualan obat ke pasien. Berkas penjualan obat disimpan ke *database* jual.dbf dan itemjual.dbf, bentuk *form*-nya dapat dilihat pada gambar 5.14.

nota Penjualan

Pendataan Penjualan Obat

Nama Dokter: dr. DIDI DEVITO, Phd. Tanggal Jual: 22-04-1998
 Nama Pasien: Paimin Bartes No Nota Jual: 98040003

KB Obat	Nama Obat	QTY	Harga	Jumlah
KO 001	Paradol	10	260	0

Sub Jumlah: 0
 Bayar: 0
 Kembali: 0

10

K < > H Baru Cetak Keluar

Gambar 5.14 Form nota penjualan

Langkah-langkah implementasi *form* nota penjualan telah diterangkan pada sub bab 5.4.1 dan sub bab 5.4.5, yaitu mengenai pembuatan *label*, *command button*, *grid* dan *text*.

Adapun penambahan yang terdapat pada *form* nota penjualan adalah modifikasi *text*, seperti pada gambar 5.15.



Gambar 5.15 Modifikasi *text* pada *form* nota penjualan

Fungsi dari *text* ini untuk memperjelas apa yang dituliskan dan proses apa yang terjadi.

Prosedur-prosedur yang ada ada pada *form* nota penjualan adalah prosedur pembayaran dan prosedur menghitung sub jumlah.

****PROSEDUR PEMBAYARAN**

```
thisform.text6.value=thisform.text5.value - thisform.text1.value
thisform.text2.inputmask="999,999,999 .-"
thisform.text2.value=thisform.text6.value
```

****PROSEDUR MENGHITUNG SUB JUMLAH**

```
Sum Jumlah_jual for Itemjual.No_jual = Jual.No_jual to x
Thisform.text1.value=x
```

5.4.10 Implementasi Form Return Obat

Form return obat (**Fretur.scx**), *form* ini berfungsi untuk melakukan transaksi pengembalian obat ke pemasok dikarenakan obat rusak. Berkas pengembalian obat disimpan ke *database* retur.dbf dan itemretur.dbf, bentuk *form*-nya dapat dilihat pada gambar 5.16.

Return

Pendataan Pengembalian Obat

Nomor Retur:

Tanggal Retur:

Nomor Faktur: 01060001

Nama Suplier:

Kode Obat	Nama Obat	QTY	Harga	Jumlah

Alasan Kembali:

Obat Rusak

Obat Kadaluarsa

Sub Jumlah:

Gambar 5.16 Form return obat

Langkah-langkah implementasi *form return* obat telah diterangkan pada sub bab 5.4.1 dan sub bab 5.4.5, yaitu mengenai pembuatan *label*, *command button*, *grid* dan *text*.

Adapun penambahan yang terdapat pada *form return* obat adalah *option group*.

Langkah-langkah pembuatan *option group*.



- 1) Klik objek *option group* pada *form control*, kemudian drag ke *form*
- 2) Mengaktifkan *properties* pilih *buttoncount*, kemudian isi dengan 2.

- 3) Mengaktifkan *option* pertama di *properties* pilih *caption*, kemudian mengetikkan **Obat Rusak**, dengan langkah yang sama, kemudian mengetikkan **Obat Kadaluarsa**.
- 4) Membuat tulisan menjadi tebal pilih *FontBold*

Prosedur-prosedur yang terdapat pada *form* return obat adalah prosedur untuk menghitung jumlah yang ada di *grid* dan prosedur menghitung sub jumlah.

```

**PROSEDURE MENGHITUNG JUMLAH
Replace itemretur.qty_retur with this.value
Replace itemretur.jumlah_retur with this.value * Obat.Hbeli_obat
IF this.value > Obat.Stok_obat .OR. this.value =0
    MESSAGEBOX("RETUR TIDAK BISA DIPENUHI!",32+4,"Konfirmasi")
    KEYBOARD'{RIGHTARROW}'
else
Locate for empty (Kode_obat)
IF !FOUND()
    APPEND BLANK
ENDIF

**PROSEDURE MENGHITUNG SUB JUMLAH
Sum jumlah_retur FOR itemretur.no_retur = retur.no_retur TO X
Thisform.Text1.value = X

```

5.4.11 Implementasi Form User

Form user (userx.scx), *form* ini berfungsi untuk mensetuap hak akses pengguna aplikasi sietem informasi farmasi. Berkas hak akses disimpan ke *database* userx.dbf, bentuk *form*-nya dapat dilihat pada gambar 5.17.

Gambar 5.17 Form user

Langkah-langkah implementasi *form user* adalah sebagai berikut :

- a. Mengaktifkan visual FoxPro.
- b. Membuat *form* kosong, dengan cara sebagai berikut :
 - 1) Klik menu *File*, dan pilih *New*, kemudian dalam kotak pilihan pilih *form* dan klik *New File*.
 - 2) Memberi judul *form*, aktifkan *properties*, kemudian klik pada *caption* ketik dengan nama **User**.
- c. Membuat *Frame* dengan cara sebagai berikut :
 - 1) Klik objek *container* pada *form control*, kemudian *drag* ke *form*.
 - 2) Mengaktifkan *properties* pilih *picture*, kemudian memilih *background* yang diinginkan.

- d. Langkah-langkah Membuat *text*, *label*, *command button* dan *check box* telah dijelaskan pada sub bab implementasi *form* obat serta sub bab implementasi *form* cari obat.

Prosedur pada implementasi *form user*, untuk pengisian nomor, nama dan *password*.

**** PENGISIAN NOMOR, NAMA, PASSWORD**

Set filter to

Go Bottom

Akh = Val(Right(no,3)) + 1001 &&

Kodenya = Right(Alt(Str(Akh)),3) &&

Nomornya='U'+Kodenya &&

locate for empty(no)

If !Found() &&

Append Blank &&

Endif &&

**** PENGISIAN PILIHAN**

Replace no with Nomornya

Replace setobat with .F.

Replace setpasien with .F.

Replace setdokter with .F.

Replace setdistrib with .F.

Replace setuser with .F.

Replace setjual with .F.

Replace setbeli with .F.

Replace setretur with .F.

Replace setcaobat with .F.

Replace setcadis with .F.

Replace setcadok with .F.

Thisform.Frame1.TxtNama.setFocus() &&

Thisform.Refresh() &&

5.4.12 Implementasi Form Akses Aplikasi Farmasi RSUD Muhammadiyah

Form akses aplikasi farmasi RSUD Muhammadiyah (**flogin.scx**), *form* ini berfungsi untuk hak akses penggunaan aplikasi sistem informasi farmasi, bentuk *form*-nya dapat dilihat pada gambar 5.18.

Gambar 5.18 Form Akses Aplikasi Farmasi RSUD Muhammadiyah

Tombol *Login* digunakan untuk masuk ke aplikasi setelah pemakai mengisi hak aksesnya yaitu nama dan *password*, sedangkan tombol batal digunakan untuk keluar dari aplikasi sistem informasi farmasi.

Langkah-langkah implementasi Akses Aplikasi Farmasi RSUD Muhammadiyah telah diterangkan pada sub bab implementasi *form* obat dan sub bab implementasi *form* cari obat, yaitu mengenai pembuatan *label*, *command button* dan *text*.

Prosedur pada implementasi Akses Aplikasi Farmasi RSUD Muhammadiyah adalah prosedur untuk *login* ke menu-menu aplikasi yang telah ditentukan pada *form user*.

****PROSEDUR LOGIN KE MENU FILE**

```
If setobat=.F.
  Set skip OF bar 1 of file.T.
Endif
If setpasien=.F.
  Set skip OF bar 2 of file.T.
Endif
If setdokter=.F.
  Set skip OF bar 3 of file.T.
Endif
If setdistrib=.F.
  Set skip OF bar 4 of file.T.
Endif
```

****PROSEDUR LOGIN KE MENU TRANSAKSI**

```
If setjual=.F.
  Set skip OF bar 1 of transaksi.T.
Endif
```

```
If setbeli=.F.  
    Set skip OF bar 2 of transaksi T.  
Endif  
If setretur=.F.  
    Set skip OF bar 3 of transaksi T.  
Endif  
  
**PROSEDUR LOGIN KE MENU CETAK  
If setctkobat=.F.  
    Set skip OF bar 1 of cetak T.  
Endif  
If setctkpasien=.F.  
    Set skip OF bar 2 of cetak T.  
Endif  
If setctkdokter=.F.  
    Set skip OF bar 3 of cetak T.  
Endif  
If setctkdistrib=.F.  
    Set skip OF bar 4 of cetak T.  
Endif  
If setctkbeli=.F.  
    Set skip OF bar 5 of cetak T.  
Endif  
If setctkjual=.F.  
    Set skip OF bar 6 of cetak T.  
Endif  
If setctkretur=.F.  
    Set skip OF bar 7 of cetak T.  
Endif  
  
**PROSEDUR LOGIN KE MENU CARI  
If setcaobat=.F.  
    Set skip OF bar 1 of cetak T.  
Endif  
If setcapas=.F.  
    Set skip OF bar 2 of cetak T.  
Endif  
If setcadok=.F.  
    Set skip OF bar 3 of cetak T.  
Endif  
If setcadis=.F.  
    Set skip OF bar 4 of cetak T.  
Endif
```

5.4.13 Implementasi Form Restore Data

Form restore data (frestore.sex), form ini berfungsi untuk memperbaiki atau memulihkan kembali direktori dari *backup* data ataupun dari penghapusan file-file di dalam direktori tersebut, bentuk *form*-nya seperti pada gambar 5.19.

Gambar 5.19 Form restore data

Langkah-langkah implementasi *form restore* telah diterangkan pada sub bab 5.4.1 dan sub bab 5.4.10, yaitu mengenai pembuatan *label*, *command button*, *option group* dan *text*.

Prosedur-prosedur yang ada pada implementasi *form restore* data adalah prosedur memasukkan direktori asal, prosedur memasukkan direktori tujuan dan prosedur *restore* data.

Prosedur memasukkan direktori asal

```
ThisForm.text1.value = getdir(",Pilih directory asal')
```

Prosedur memasukkan direktori tujuan

```
ThisForm.txtfilename.value = getdir(",Pilih directory untuk restore')
```

**PROSEDUR RESTORE DATA OBAT

```
if thisform.op.op1.value=1
  IF not FILE(alltrim(thisform.txtfilename.value)+"obat.dbf")
    messagebox("File tabel obat yang akan di restore"+chr(13)+;
      "Pada directory "+alltrim(thisform.txtfilename.value)+" "+chr(13)+;
      "Tidak ditemukan..... !!!",64,"Perhatian....")
  else
    copy file alltrim(thisform.txtfilename.value)+"obat.dbf" to alltrim(thisform.text1.value)+;
      "obat.dbf" copy file alltrim(thisform.txtfilename.value)+;
      "obat.cdx" to alltrim(thisform.text1.value)+"obat.cdx"
  endif
endif
```

**PROSEDUR RESTORE DATA PEMASOK

```
if thisform.op.op2.value=1
  IF not FILE(alltrim(thisform.txtfilename.value)+"distrib.dbf")
    messagebox("File tabel distributor yang akan di restore"+chr(13)+;
      "Pada directory "+alltrim(thisform.txtfilename.value)+" "+chr(13)+;
      "Tidak ditemukan..... !!!",64,"Perhatian....")
  else
    copy file alltrim(thisform.txtfilename.value)+"distrib.dbf" to alltrim(thisform.text1.value)+;
      "distrib.dbf" copy file alltrim(thisform.txtfilename.value)+;
      "distrib.cdx" to alltrim(thisform.text1.value)+;"distrib.cdx"
  endif
endif
```

**PROSEDUR RESTORE DATA DOKTER

```
if thisform.op.op3.value=1
  IF not FILE(alltrim(thisform.txtfilename.value)+"dokter.dbf")
    messagebox("File tabel dokter yang akan di restore"+chr(13)+;
      "Pada directory "+alltrim(thisform.txtfilename.value)+" "+chr(13)+;
      "Tidak ditemukan..... !!!",64,"Perhatian....")
  else
    copy file alltrim(thisform.txtfilename.value)+"dokter.dbf" to alltrim(thisform.text1.value)+;
      "dokter.dbf" copy file alltrim(thisform.txtfilename.value)+;
      "dokter.cdx" to alltrim(thisform.text1.value)+;"dokter.cdx"
  endif
endif
```

**PROSEDUR RESTORE DATA PASIEN

```
if thisform.op.op4.value=1
  IF not FILE(alltrim(thisform.txtfilename.value)+"pasien.dbf")
    messagebox("File tabel pasien yang akan di restore"+chr(13)+;
      "Pada directory "+alltrim(thisform.txtfilename.value)+" "+chr(13)+;
      "Tidak ditemukan..... !!!",64,"Perhatian....")
  else
```

```

        copy file alltrim(thisform.txtfilename.value)+" pasien.dbf" to alltrim(thisform.text1.value)+;
        " pasien.dbf" copy file alltrim(thisform.txtfilename.value)+;
        " pasien.cdx" to alltrim(thisform.text1.value)+;" pasien.cdx"
    endif
endif

**PROSEDUR RESTORE DATA PEMBELIAN
if thisform.op.op5.value=1
    IF not FILE(alltrim(thisform.txtfilename.value)+"beli.dbf")
        messagebox("File tabel beli yang akan di restore"+chr(13)+;
        "Pada directory "+alltrim(thisform.txtfilename.value)+" "+chr(13)+;
        "Tidak ditemukan..... !!!",64,"Perhatian....")
    else
        copy file alltrim(thisform.txtfilename.value)+" dokter.dbf" to alltrim(thisform.text1.value)+;
        " beli.dbf" copy file alltrim(thisform.txtfilename.value)+;
        " beli.cdx" to alltrim(thisform.text1.value)+;" beli.cdx"
    endif
endif

**PROSEDUR RESTORE DATA PENJUALAN
if thisform.op.op6.value=1
    IF not FILE(alltrim(thisform.txtfilename.value)+"jual.dbf")
        messagebox("File tabel jual yang akan di restore"+chr(13)+;
        "Pada directory "+alltrim(thisform.txtfilename.value)+" "+chr(13)+;
        "Tidak ditemukan..... !!!",64,"Perhatian....")
    else
        copy file alltrim(thisform.txtfilename.value)+" dokter.dbf" to alltrim(thisform.text1.value)+;
        " jual.dbf" copy file alltrim(thisform.txtfilename.value)+;
        " jual.cdx" to alltrim(thisform.text1.value)+;" jual.cdx"
    endif
endif

**PROSEDUR RESTORE DATA PENGEMBALIAN
if thisform.op.op7.value=1
    IF not FILE(alltrim(thisform.txtfilename.value)+"retur.dbf")
        messagebox("File tabel retur yang akan di restore"+chr(13)+;
        "Pada directory "+alltrim(thisform.txtfilename.value)+" "+chr(13)+;
        "Tidak ditemukan..... !!!",64,"Perhatian....")
    else
        copy file alltrim(thisform.txtfilename.value)+" dokter.dbf" to alltrim(thisform.text1.value)+;
        " retur.dbf" copy file alltrim(thisform.txtfilename.value)+;
        " retur.cdx" to alltrim(thisform.text1.value)+;" retur.cdx"
    endif
endif

```

5.4.14 Implementasi Form Backup Data

Form backup data (fbackup.sex), form ini berfungsi untuk membackup data-data dalam file yang diperlukan pada sistem informasi farmasi, bentuk form-nya seperti pada gambar 5.20.

Gambar 5.20 Form backup data

Langkah-langkah implementasi *form backup* data telah diterangkan pada sub bab 5.4.1 dan sub bab 5.4.10, yaitu mengenai pembuatan *label*, *command button*, *option group* dan *text*.

Prosedur-prosedur yang ada pada implementasi *form backup* data adalah prosedur memasukkan direktori asal, prosedur memasukkan direktori tujuan dan prosedur *backup* data.

Prosedur memasukkan direktori asal



```
ThisForm.text1.value = getdir(", 'Pilih directory asal')
```

Prosedur memasukkan direktori tujuan



```
ThisForm.txtfilename.value = getdir(", 'Pilih directory untuk backup')
```

****PROSEDUR BACKUP DATA OBAT**

```
if thisform.op.op1.value=1
  IF not FILE(alltrim(thisform.txtfilename.value)+"obat.dbf")
    messagebox("File tabel obat yang akan di backup "+chr(13)+
      "Pada directory "+alltrim(thisform.txtfilename.value)+" "+chr(13)+
      "Tidak ditemukan..... !!!",64,"Perhatian....")
  else
    copy file alltrim(thisform.txtfilename.value)+"obat.dbf" to alltrim(thisform.text1.value)+
      "obat.dbf" copy file alltrim(thisform.txtfilename.value)+
      "obat.cdx" to alltrim(thisform.text1.value)+"obat.cdx"
  endif
endif
```

****PROSEDUR BACKUP DATA PEMASOK**

```
if thisform.op.op2.value=1
  IF not FILE(alltrim(thisform.txtfilename.value)+"distrib.dbf")
    messagebox("File tabel distributor yang akan di backup "+chr(13)+
      "Pada directory "+alltrim(thisform.txtfilename.value)+" "+chr(13)+
      "Tidak ditemukan..... !!!",64,"Perhatian....")
  else
    copy file alltrim(thisform.txtfilename.value)+"distrib.dbf" to alltrim(thisform.text1.value)+
      "distrib.dbf" copy file alltrim(thisform.txtfilename.value)+
      "distrib.cdx" to alltrim(thisform.text1.value)+,"distrib.cdx"
  endif
endif
```

****PROSEDUR BACKUP DATA DOKTER**

```
if thisform.op.op3.value=1
  IF not FILE(alltrim(thisform.txtfilename.value)+"dokter.dbf")
    messagebox("File tabel dokter yang akan di backup "+chr(13)+
      "Pada directory "+alltrim(thisform.txtfilename.value)+" "+chr(13)+
      "Tidak ditemukan..... !!!",64,"Perhatian....")
  else
    copy file alltrim(thisform.txtfilename.value)+"dokter.dbf" to alltrim(thisform.text1.value)+
      "dokter.dbf" copy file alltrim(thisform.txtfilename.value)+
      "dokter.cdx" to alltrim(thisform.text1.value)+,"dokter.cdx"
  endif
endif
```

****PROSEDUR BACKUP DATA PASIEN**

```

if thisform.op.op4.value=1
  IF not FILE(alltrim(thisform.txtfilename.value)+"pasien.dbf")
    messagebox("File tabel pasien yang akan di backup "+chr(13)+;
      "Pada directory "+alltrim(thisform.txtfilename.value)+" "+chr(13)+;
      "Tidak ditemukan..... !!!",64,"Perhatian...")
  else
    copy file alltrim(thisform.txtfilename.value)+" pasien.dbf" to alltrim(thisform.text1.value)+;
    " pasien.dbf" copy file alltrim(thisform.txtfilename.value)+;
    " pasien.cdx" to alltrim(thisform.text1.value)+;" pasien.cdx"
  endif
endif

**PROSEDUR BACKUP DATA PEMBELIAN
if thisform.op.op5.value=1
  IF not FILE(alltrim(thisform.txtfilename.value)+"beli.dbf")
    messagebox("File tabel beli yang akan di backup "+chr(13)+;
      "Pada directory "+alltrim(thisform.txtfilename.value)+" "+chr(13)+;
      "Tidak ditemukan..... !!!",64,"Perhatian...")
  else
    copy file alltrim(thisform.txtfilename.value)+" dokter.dbf" to alltrim(thisform.text1.value)+;
    " beli.dbf" copy file alltrim(thisform.txtfilename.value)+;
    " beli.cdx" to alltrim(thisform.text1.value)+;" beli.cdx"
  endif
endif

**PROSEDUR BACKUP DATA PENJUALAN
if thisform.op.op6.value=1
  IF not FILE(alltrim(thisform.txtfilename.value)+"jual.dbf")
    messagebox("File tabel jual yang akan di backup "+chr(13)+;
      "Pada directory "+alltrim(thisform.txtfilename.value)+" "+chr(13)+;
      "Tidak ditemukan..... !!!",64,"Perhatian...")
  else
    copy file alltrim(thisform.txtfilename.value)+" dokter.dbf" to alltrim(thisform.text1.value)+;
    " jual.dbf" copy file alltrim(thisform.txtfilename.value)+;
    " jual.cdx" to alltrim(thisform.text1.value)+;" jual.cdx"
  endif
endif

**PROSEDUR BACKUP DATA PENGEMBALIAN
if thisform.op.op7.value=1
  IF not FILE(alltrim(thisform.txtfilename.value)+"retur.dbf")
    messagebox("File tabel retur yang akan di backup "+chr(13)+;
      "Pada directory "+alltrim(thisform.txtfilename.value)+" "+chr(13)+;
      "Tidak ditemukan..... !!!",64,"Perhatian...")
  else
    copy file alltrim(thisform.txtfilename.value)+" dokter.dbf" to alltrim(thisform.text1.value)+;
    " retur.dbf" copy file alltrim(thisform.txtfilename.value)+;
    " retur.cdx" to alltrim(thisform.text1.value)+;" retur.cdx"
  endif
endif
endif
endif

```

5.4.15 Implementasi Form Hapus Data

Form hapus data (**fperawat.scx**), form ini berfungsi untuk menghapus data-data dalam *file* yang tidak diperlukan lagi pada sistem informasi farmasi, bentuk form-nya seperti pada gambar 5.21.

Gambar 5.21 Form hapus data

Langkah-langkah implementasi *form* perawatan telah diterangkan pada sub bab 5.4.1 dan sub bab 5.10, yaitu mengenai pembuatan *label*, *command button*, *option group* dan *text*.

Prosedur yang ada pada implementasi *form* hapus data adalah prosedur untuk melakukan hapus data, prosedurnya adalah sebagai berikut :

****PROSEDUR HAPUS DATA OBAT**

```
if thisform.op.op1.value=1
    sele obat
    delete all
    pack
    reindex
endif
```

****PROSEDUR HAPUS DATA PEMASOK**

```
if thisform.op.op2.value=1
    sele distrib
    delete all
    pack
    reindex
endif
```

****PROSEDUR HAPUS DATA DOKTER**

```
if thisform.op.op3.value=1
    sele dokter
    delete all
    pack
    reindex
endif
```

****PROSEDUR HAPUS DATA PASIEN**

```
if thisform.op.op4.value=1
    sele pasien
    delete all
    pack
    reindex
endif
```

****PROSEDUR HAPUS DATA PEMBELIAN**

```
if thisform.op.op5.value=1
    sele beli
    delete all
    pack
    reindex
endif
```

****PROSEDUR HAPUS DATA PENJUALAN**

```
if thisform.op.op6.value=1
    sele jual
    delete all
    pack
    reindex
endif
```

```

** PROSEDUR HAPUS DATA PENGEMBALIAN
if thisform.op.op7.value=1
    sele retur
    delete all
    pack
    reindex
endif

** PROSEDUR HAPUS DATA SETUP HAK AKSES
if thisform.op.op8.value=1
    sele userx
    delete all
    pack
    reindex
endif

```

5.4.16 Implementasi Pembuatan Bentuk Laporan

Program laporan dibuat sebagai dokumentasi dari semua kegiatan yang menyangkut sistem informasi farmasi. Laporan ini membantu dalam mengawasi jalannya kegiatan mulai dari obat masuk sampai obat keluar ke pasien. Program laporan tersebut terdiri dari 7 laporan, yaitu laporan persediaan obat, laporan daftar dokter, laporan daftar pemasok, laporan daftar pasien, laporan pembelian obat, laporan penjualan obat, laporan pengembalian obat. Adapun bentuk-bentuk dari laporan dapat dilihat pada bagian lampiran.

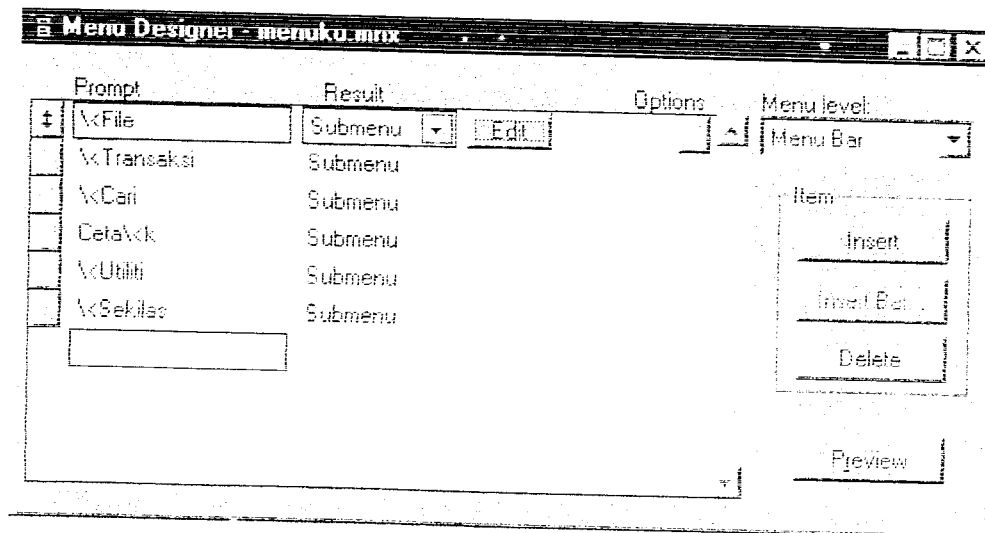
5.4.17 Implementasi Menu Utama

Selain fasilitas-fasilitas yang telah dijelaskan pada sub bab fasilitas pengembangan Foxpro juga menyediakan fasilitas menu *designer* yang dapat digunakan untuk mengakses *function* dalam yang tersedia di sistem.

Fungsi dari menu utama adalah untuk mengatur seluruh jalannya program, mulai dari menampilkan menu berupa *bar*, *pull down* menu dan semua dialog yang ada.

Langkah-langkah implementasi menu utama adalah sebagai berikut :

- a. Mendefinisikan menubar, dengan cara sebagai berikut :
 - 1) Klik menu *File* dan pilih *New* sehingga muncul kotak dialog *New*.
 - 2) Pada kotak dialog *New* pilih *Menu*, kemudian klik *New File* sehingga muncul tampilan kotak dialog *New Menu* seperti pada gambar 5.21.
 - 3) Ketik nama menubar yang akan dipakai dalam sistem informasi farmasi yaitu *File*, *Transaksi*, *Cari*, *Cetak*, *Utility*, *Sekilas*, kemudian pada kolom *result* isi dengan *Submenu*.



Gambar 5.21 Kotak dialog menu designer

- b. Memasukkan submenu dari masing-masing menubar, dengan cara sebagai berikut :

- 1) Letakkan kursor pada baris menubar *File*.
- 2) Klik tombol *Create* yang ada disebelah kanan kolom *Result* dan kemudian mendefinisikan sub-submenu untuk menubar *File*. Kemudian pada kolom *Result* isikan *command* dan pada kolom *option* isikan *directory* diikuti dengan nama program yang diinginkan.
- 3) Memberikan *short key* (Alt) dengan cara menambahkan \< diawal kata yang akan dibuat *short key* (Alt).
- 4) Menambah garis pemisah antar submenu yaitu dengan cara isikan \-.
- 5) Mengulangi langkah 1-4 untuk mendefinisikan menubar selanjutnya.
- 6) Setelah menyimpan kemudian lakukan *generate* pada menu yang dibuat.

BAB VI

ANALISIS KINERJA PERANGKAT LUNAK

6.1 Pendahuluan

Aplikasi Sistem Informasi Farmasi berguna untuk membantu dalam melakukan kegiatan yang ada di bagian farmasi, mulai dari proses pemasukkan obat dari pemasok sampai pengeluaran obat ke pasien serta pembuatan laporan-laporan dari masing-masing kegiatan dalam proses.

Sesuai dengan rumusan masalah, Aplikasi Sistem Informasi Farmasi ini diterapkan untuk memberikan informasi tentang obat yang ada secara cepat, tepat dan akurat untuk pasien dan petugas bagian farmasi itu sendiri serta untuk menghindari adanya kerangkapan kode obat.

Dengan adanya permasalahan di atas maka perlu diterapkan Aplikasi Sistem Informasi Farmasi, karena aplikasi ini menggunakan sistem relasi antar *database* yang ada sehingga memudahkan dalam melakukan semua kegiatan yang ada dan terdapat fasilitas pencarian data.

Sebelum menganalisis sistem terhadap masalah yang telah dijelaskan di atas, maka program harus bebas terlebih dahulu dari kesalahan-kesalahan yang mungkin dapat terjadi.

6.2. Analisis Kerja Terhadap Akses Aplikasi

Pertama kali dalam menggunakan aplikasi ini adalah memasukkan *password*, yaitu untuk masuk ke aplikasi sesuai dengan haknya. Hak akses ini berfungsi untuk mengatur penggunaan aplikasi sesuai dengan haknya masing-masing. Pengisian nama dan *password* sesuai dengan *setup* hak akses yang telah ditentukan. Tampilan *form setup* hak akses seperti pada gambar 5.17.

6.3 Analisis Kerja Terhadap Berkas Obat

Pengolahan data obat pada aplikasi Sistem Informasi Farmasi ini merupakan pengolahan data yang pertama kali. Data obat ini merupakan induk dari semua kegiatan yang menyangkut pengolahan obat di farmasi yang tersimpan dalam *database* obat.dbf. Tampilan *form* pendataan obat seperti terlihat pada gambar 5.2. Pada pengisian data obat ini terdiri dari kode obat, nama obat, jenis, resep, pabrik, harga beli, dan harga jual.

Untuk melihat data yang telah dimasukan dapat dilihat pada daftar data obat. Sedangkan pengisian laba, tanggal kadaluarsa, dan stok obat otomatis terisi pada saat pengisian data pembelian obat dari pemasok.

6.4 Analisis Kerja Terhadap Berkas Pemasok

Berkas pemasok digunakan sebagai dokumentasi data pemasok dan tersimpan dalam *database* distrib.dbf yang nantinya digunakan sebagai laporan. Data pemasok ini digunakan untuk pemesanan kembali sejumlah obat dengan menghubungkan

langsung pemasok yang diinginkan. Tampilan *form* data pemasok dapat dilihat pada gambar 5.4.

Pada pengisian data pemasok ini terdiri dari kode pemasok, nama pemasok, alamat, kota, nomor telepon dan kota. Pendataan data pemasok ini sama dengan pendataan data obat, yaitu terdapat fasilitas untuk melihat daftar pemasok.

6.5 Analisis Kerja Terhadap Berkas Dokter

Berkas dokter digunakan sebagai dokumentasi data dokter dan tersimpan dalam *database* dokter.dbf yang nantinya digunakan sebagai laporan. Data dokter ini digunakan untuk pengeluaran resep/penjualan obat. Tampilan *form* pendataan dokter dapat dilihat pada gambar 5.6.

Data-data dokter yang dimasukkan pada *form* pendataan dokter adalah kode dokter, nama dokter, jenis kelamin, spesialis, alamat, kota dan telepon. Dalam pengoperasian pengolahan data dokter juga terdapat fasilitas untuk melihat daftar dokter.

6.6 Analisis Kerja Terhadap Berkas Pasien

Berkas pasien ini digunakan untuk pengeluaran resep/penjualan obat sebagai dokumentasi data pasien dan tersimpan dalam *database* pasien.dbf yang nantinya digunakan sebagai laporan. Tampilan *form* pendataan pasien dapat dilihat pada gambar 5.7

6.7 Analisis Kerja Terhadap Berkas Pembelian

Berkas ini digunakan untuk menyimpan data pembelian yang dilakukan. Pendataan data transaksi pembelian ini tersimpan dalam *database* beli.dbf yang nantinya digunakan sebagai laporan

Pengisian data pada *form* pembelian terdiri dari no.faktur, tgl terima, nama pemasok, dan kontak, kelima data ini akan terisi secara otomatis. Sedangkan data jatuh tempo dan kode pemasok diisikan secara manual. Untuk pengisian data yang ada di *grid*, yaitu kode obat, *quantity*, tanggal kadaluarsa, dan diskon diisikan secara manual, sedangkan untuk nama obat, harga beli, harga jual dan jumlah akan terisi dengan otomatis. Data penjualan ini akan digunakan untuk menentukan jumlah persediaan obat. Tampilan *form* pendataan pembelian obat dapat dilihat pada gambar 5.13.

6.8 Analisis Kerja Terhadap Berkas Penjualan

Pendataan transaksi penjualan tersimpan dalam *database* jual.dbf yang nantinya digunakan sebagai laporan. Data penjualan ini digunakan juga untuk menentukan jumlah persediaan obat.

Data-data yang diisikan pada *form* penjualan adalah tanggal penjualan terisi secara otomatis sesuai dengan tanggal pada saat penggunaan dan nomor nota diisikan secara otomatis. Sedangkan nama pasien, nama dokter diambil dari berkas pasien dan berkas dokter. Pada pengisian data di *grid* untuk kode obat, *quantity* diisikan secara manual, sedangkan nama obat harga dan jumlah akan terisi secara otomatis.

Tampilan *form* pendataan penjualan obat/nota penjualan dapat dilihat pada gambar 5.14

6.9 Analisis Kerja Terhadap Berkas Return

Berkas ini digunakan untuk menyimpan data obat-obat rusak/cacat yang akan dikembalikan ke pada pemasok. Pendataan data pengembalian ini tersimpan dalam *database* retur.dbf yang nantinya digunakan sebagai laporan

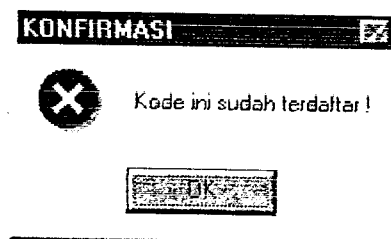
Data-data yang diisikan pada *form return* adalah nomor *return* yang terisi secara otomatis, nomor faktur diambil dari berkas pembelian, tanggal *return* terisi secara otomatis sesuai dengan tanggal pengoperasian sedangkan nama pemasok diambil dari berkas pemasok. Pada pengisian data di *grid* untuk kode obat, *quantity* diisikan secara manual, sedangkan nama obat harga dan jumlah akan terisi secara otomatis. Tampilan *form* pendataan *return* pembelia obat dapat dilihat pada gambar 5.15

6.10 Pesan atau Informasi

Pesan-pesan yang muncul pada saat program dijalankan atau pada saat kesalahan dalam memasukan data bertujuan untuk mempermudah menjalankan program aplikasi. Pesan-pesan yang terdapat dalam program aplikasi sistem informasi farmasi antara lain :

1. Kerangkapan kode obat

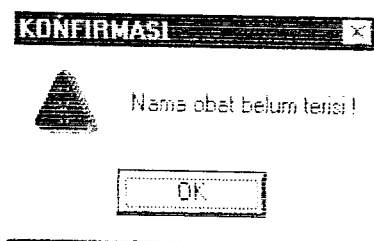
Untuk menghindari kerangkapan pengisian kode obat, maka ditambahkan peringatan yang berupa konfirmasi kesalahan, seperti pada gambar 6.1.



Gambar 6.1 Konfirmasi kesalahan pengisian kerangkapan kode obat

2. Konfirmasi kesalahan pengisian data

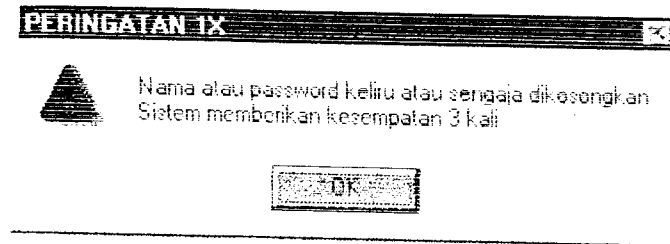
Pada pengisian data obat yang terdiri dari kode obat, nama obat, jenis, resep, pabrik, harga beli, dan harga jual terdapat konfirmasi kesalahan jika salah satu dari data belum terisi. Hal ini untuk menghindari kesalahan atau kurang lengkapnya dalam pengisian data obat. Salah satu contoh konfirmasi kesalahan apabila nama obat belum terisi ditunjukkan seperti pada gambar 6.2.



Gambar 6.2 Konfirmasi kesalahan pengisian data

3. Peringatan pengisian *password*

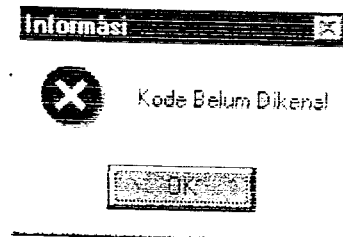
Untuk menghindari adanya pemakai yang tidak bertanggung jawab dalam menjalankan aplikasi ini, maka pada penggunaan *password* dibatasi pengisiannya tiga (3) kali kesalahan, dan apabila kesalahan sampai tiga (3) kali maka akan keluar dari sistem. Peringatan yang berupa kesalahan ditunjukkan pada gambar 6.3.



Gambar 6.3 Peringatan kesalahan pemasukkan *password*

4. Informasi pembelian obat

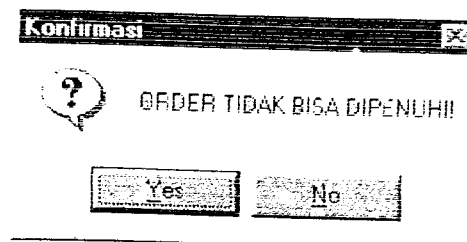
Pada pembelian obat, kode obat yang dimasukan harus sesuai dengan kode obat yang ada pada pendataan data obat. Tampilan informasi apabila kesalahan pada pengisian kode obat seperti pada gambar 6.4.



Gambar 6.4. Informasi pembelian obat

5. Konfirmasi penjualan obat

Penjualan obat harus sesuai dengan stok yang ada pada berkas obat dan apabila penjualan lebih besar dari stok yang ada, maka akan ada konfirmasi seperti pada gambar 6.5.



Gambar 6.5. Konfirmasi penjualan obat

6.11 Analisis Sistem Terhadap Masalah

Setelah selesai pengujian program langkah selanjutnya pengujian sistem terhadap masalah yang dihadapi pada penulisan tugas akhir ini.

1. Analisis Integritas Data

Integritas data atau keterkaitan antara *file* satu dengan yang lain dapat menangani masalah keterlambatan antara data satu dengan yang lain disimpan dalam satu *file*. Sistem dapat digunakan secara bersama-sama baik pengguna maupun *administrator* apabila terjadi *update* data.

Maka data yang berkaitan dengan yang lain juga mengalami perubahan secara otomatis. Contoh pada pembelian akan berpengaruh kepada persediaan obat.

2 Analisis Antarmuka (*interface*) Sistem

Antarmuka menjadi hal yang sangat penting dalam suatu program yang mempunyai antarmuka yang baik harus bersifat ramah pengguna (*user friendly*/bersifat ramah pengguna), artinya mudah digunakan serta pengguna tidak banyak menghabiskan waktu untuk menyelesaikan pekerjaannya yang menggunakan sistem ini.

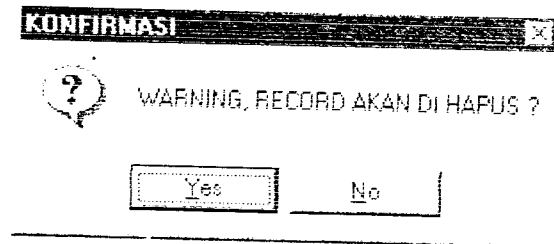
Antarmuka yang digunakan pada sistem informasi farmasi ini dibuat semaksimal mungkin bersifat ramah pengguna, hal ini dilihat dari segi.

a. Level keahlian

Perangkat lunak dikembangkan untuk memenuhi kebutuhan pengguna. Pengguna sistem yang ada pada sistem informasi farmasi dibagi menjadi dua :

- 1) Pegawai sebagai *user*, pegawai mempunyai hak menggunakan sistem, mempunyai hak melihat dan tidak dapat mengubah atau mengupdate data.
 - 2) Pegawai sebagai *administrator*, pegawai yang berhak dalam pengolahan data sistem informasi farmasi, mengetahui fungsi-fungsi sistem, menangani input/output dan kerja sitem yang ada.
2. Interaksi manusia dan komputer

Perangkat lunak bersifat komunikatif, artinya mudah dimengerti oleh pengguna, contoh dalam pengolahan apabila akan menghapus data maka akan tampil kotak-kotak pesan seperti pada gambar 6.6.



Gambar 6.6 Tampilan kotak pesan

Perangkat lunak juga menampilkan suatu pesan-pesan kesalahan, apabila terjadi kesalahan seperti telah dijelaskan di atas.

3. Input data

Penginputan data didesain dengan seramah mungkin (*user friendly*/bersifat ramah pengguna), penginputan atau *update* data, langsung disimpan dengan cara mengklik tombol perintah simpan.

6.12 Evaluasi Sistem

Dalam pembuatan atau perancangan sistem, tentunya terdapat beberapa kelebihan dan kelemahan. Adapun kelebihan maupun kekurangan tersebut adalah sebagai berikut :

1. Kelebihan

- a. Adanya pesan-pesan yang muncul pada saat program dijalankan, sehingga akan mempermudah dalam menjalankan program aplikasi.
- b. Data yang dimasukan dapat disimpan, diperbaharui dan dipergunakan kembali.
- c. Pembuatan laporan yang mudah dan cepat
- d. Adanya hak akses yang dapat menjaga data-data dari pemakai yang tidak bertanggung jawab.

2. Kekurangan

- a. Belum adanya dokumen mengenai sistem yang dibuat seperti, struktur data, prosedur dan petunjuk penggunaan sistem sehingga akan mengalami kesulitan untuk perubahan atau memodifikasi sistem.
- b. Untuk melakukan *update* atau hapus *record* pada tabel yang memiliki *record* yang banyak akan mengalami kesulitan untuk mencari *record*, jika *record* yang akan dihapus tersebut berada di tengah-tengah *file* dikarenakan pada *form* pencarian informasi.

BAB VII

PENUTUP

7.1 Kesimpulan

Kesimpulan yang dapat dikemukakan pada akhir penulisan ini adalah :

1. Aplikasi sistem informasi farmasi yang dikembangkan terdapat fasilitas untuk melihat daftar persediaan obat, sehingga semua persediaan obat yang tersisa dapat dilihat dengan cepat dan mudah guna memudahkan dalam melakukan transaksi pembelian dan penjualan obat.
2. Penambahan data baru akan terkunci secara otomatis sehingga memberikan keamanan data yang bersifat tunggal seperti kode pemasok, kode pasien, nomor faktur dan nomor *return*.
3. Untuk menghindari kesalahn dalam pengisian data baru seperti data obat, data pemasok, data dokter, data pasien, data pembelian, data penjualan dan data pengembalian obat deiberikan pesan-pesan kesalahan atau konfirmasi-konfirmasi lainnya.
4. Aplikasi sistem informasi farmasi yang dikembangkan mampu memberikan kemudahan bagi pegawai rumah sakit maupun pasien, antara lain memiliki antarmuka yang memudahkan dalam pengoperasiannya, serta mampu digunakan dengan *multi user*.

7.2 Saran

Penulis memberikan saran-saran sebagai berikut :

1. Perlu adanya pembaharuan perangkat keras (*upgrade*) yang ada di RSU Muhammadiyah Yogyakarta untuk menunjang proses pengolahan data yang lebih maksimal.
2. Untuk menjaga keamanan basis data dari kerusakan yang disebabkan karena sistemnya atau karena virus, maka sebaliknya melakukan *backup* terhadap basis data satu bulan sekali
3. Pegawai yang berhubungan langsung dengan sistem ini harus bisa merawat sistem dengan melakukan *backup* data, hapus data, dan *restore* data, agar sistem bisa berjalan dengan baik.
4. Untuk mengatasi kelemahan-kelemahan sistem yang telah disebutkan pada bab sebelumnya (BAB VI) adalah sebagai berikut :
 - a. Membuat menu *help* yang isinya dokumen struktur data, prosedur dan petunjuk menggunakan sistem.
 - b. Pada *form* pencarian tidak hanya menampilkan informasi, tetapi juga diberikan perintah untuk mengupdate dan menghapus data.

DAFTAR PUSTAKA

- [DAV93] Gordon B. Davis, *Sistem Informasi Manajemen*. PT. Pustaka Binaman, Pressindo, 1993.
- [FAT99] Fathansyah, *Basis Data*, Bandung : Informatika, 1999
- [HAR93] Hartianto Kristanto, *Konsep dan Perancangan Database*. Yogyakarta : Andi Offset, 1993.
- [INS97] P. Insap Santosa, *Interaksi Manusia dan Komputer*. Yogyakarta : Andi Offset, 1997.
- [MOE86] Moekijai, *Pengantar Sistem Informatika Manajemen*. Bandung : PT. Remaja Rosdakarya, 1986.
- [TOG95] Simatupang .M Togar, *Teori Sistem*. Yogyakarta . Andi Offset, 1995.



RUMAH SAKIT UMUM PKU MUHAMMADIYAH

Jln. KH. Ahmad Dahlan No. 20 Yogyakarta 55122

Telepon : (0274) 512653, 512654, 513871
IGD : (0274) 566635
Faximile : (0274) 566129
E-mail : pku@yogya.wasantara.net.id

Rekening Bank :
Bank Mandiri Cabang Yogyakarta
RSU PKU Muhammadiyah Yogyakarta
AC 009.2043.435

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

SURAT KETERANGAN

NO : 1688 /E-IV/PI.24.6/VIII/2001

Assalamu'alaikum wr. wb.

Direktur RSU PKU Muhammadiyah Yogyakarta menerangkan bahwa :

Nama : Winarsih
NIM : 95 351 100
Institusi : Jurusan Tehnik Informatika, Fakultas Teknologi Industri,
Universitas Islam Indonesia Yogyakarta

Telah menyelesaikan penelitian di RSU PKU Muhammadiyah Yogyakarta dengan judul :
**"Sistim Informasi Farmasi
Di RSU PKU Muhammadiyah Yogyakarta"**

Demikian surat keterangan ini dibuat, untuk dapat dipergunakan sebagaimana mestinya.

Wassalamu'alaikum wr. wb.

Yogyakarta, 9 Agustus 2001
Direktur,

dr. H. Muhammad Iqbal, Sp. PD
NBM. 753.483