

SURAT TUGAS
1379/B.01/LPPM-UBSI/VII/2024

Tentang

**Pelindungan Ciptaan di Bidang Ilmu Pengetahuan, Seni dan Sastra Berdasarkan Undang-Undang
Nomor 28 Tahun 2014 tentang Hak Cipta**
Nomor dan Tanggal Permohonan : EC00202464140, 11 Juli 2024
Nomor Pencatatan : 000639495

**PADA SURAT PENCATATAN CIPTAAN KEMENTERIAN HUKUM DAN HAK ASASI
MANUSIA REPUBLIK INDONESIA**

Program Komputer

Judul Ciptaan :

**RANCANG BANGUN PROGRAM POINT OF SALES BERBASIS CLIENT SERVER (Studi
Kasus Pada K24 Pharmacy)**

MEMUTUSKAN

- Pertama : Kepada saudara
Melyani S.Pd, MM
Sebagai Pencipta yang mempublikasikan karyanya.
- Kedua : Mempunyai tugas sbb:
Melaksanakan Tugas yang diberikan dengan penuh rasa tanggung jawab.
- Ketiga : Keputusan ini berlaku sejak tanggal ditetapkan, dengan ketentuan apabila
dikemudian hari terdapat kekeliruan akan diubah dan diperbaiki sebagaimana
mestinya.

Jakarta, 4 Juli 2024

LPPM Universitas Bina Sarana Informatika

Ketua



Dr. Taufik Baidawi, M.kom

Tembusan

- Rektor Universitas Bina Sarana Informatika
- Arsip
- Ybs

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202464140, 11 Juli 2024

Pencipta

Nama : Melyani, Ahmad Rafik dkk

Alamat : Perumahan Bunga Rampai VII GG 7, RT 006 RW 010 No.20, Perumnas Klender Jakarta Timur 12860, Duren Sawit, Jakarta Timur, DKI Jakarta, 12860

Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : Melyani, Ahmad Rafik dkk

Alamat : Perumahan Bunga Rampai VII GG 7, RT 006 RW 010 No.20, Perumnas Klender Jakarta Timur 12860, Duren Sawit, Jakarta Timur, DKI Jakarta, 12860

Kewarganegaraan : Indonesia

Jenis Ciptaan : Program Komputer

Judul Ciptaan : RANCANG BANGUN PROGRAM POINT OF SALES BERBASIS CLIENT SERVER (Studi Kasus Pada K24 Pharmacy)

Tanggal dan tempat diumumkan untuk pertama kali : 11 Juli 2024, di Jakarta Timur
di wilayah Indonesia atau di luar wilayah Indonesia

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000639495

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
u.b

Direktur Hak Cipta dan Desain Industri

IGNATIUS M.T. SILALAH
NIP. 196812301996031001

Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Melyani	Perumahan Bunga Rampai VII GG 7 , RT 006 RW 010 No.20, Perumnas Klender Jakarta Timur 12860 , Duren Sawit, Jakarta Timur
2	Ahmad Rafik	Perumahan Bunga Rampai IX Blok 25 No. 133 Malaka Jaya, Duren Sawit Jakarta Timur 13460, Duren Sawit, Jakarta Timur
3	Mety Titin Herawaty	Grand. Residance Cluster Tirtyasa AC 5 No 2 Bekasi Timur 17320, Bekasi Timur, Bekasi
4	Amalia Tresna Fadhillah	Jalan Cipinang Muara 1 No.53, Jatinegara, Jakarta Timur, 13420, Jatinegara, Jakarta Timur
5	Muhamad Khadafi Firman	Jalan Cipinang Muara 1 No.53, Jatinegara, Jakarta Timur, 13420, Jatinegara, Jakarta Timur
6	Anjas Ramadhani	Jalan Kp Cirewed Rt03/01 No139 Desa Sukadamai Cikupa Tangerang 15710, Cikupa, Tangerang
7	Faizal Roni	Jl. Yapen 2 Blok QB-15 Nusaloka, Perumahan BSD, Serpong Tangerang Selatan , Serpong, Tangerang Selatan
8	Rahayu Swastika	Jl. I Barat No. 9 RT.011/06 Kebon Baru, Tebet Jakarta Selatan 12830, Tebet, Jakarta Selatan
9	Slamet Maryoso	Jl. Beringin 53 RT 01/I Bancong Kec. Wonoasri Kab. Madiun Jawa Timur 63157, Wonoasri, Madiun

LAMPIRAN PEMEGANG

No	Nama	Alamat
1	Melyani	Perumahan Bunga Rampai VII GG 7 , RT 006 RW 010 No.20, Perumnas Klender Jakarta Timur 12860 , Duren Sawit, Jakarta Timur
2	Ahmad Rafik	Perumahan Bunga Rampai IX Blok 25 No. 133 Malaka Jaya, Duren Sawit Jakarta Timur 13460, Duren Sawit, Jakarta Timur
3	Mety Titin Herawaty	Grand. Residance Cluster Tirtyasa AC 5 No 2 Bekasi Timur 17320, Bekasi Timur, Bekasi
4	Amalia Tresna Fadhillah	Jalan Cipinang Muara 1 No.53, Jatinegara, Jakarta Timur, 13420, Jatinegara, Jakarta Timur
5	Muhamad Khadafi Firman	Jalan Cipinang Muara 1 No.53, Jatinegara, Jakarta Timur, 13420, Jatinegara, Jakarta Timur
6	Anjas Ramadhani	Jalan Kp Cirewed Rt03/01 No139 Desa Sukadamai Cikupa Tangerang 15710, Cikupa, Tangerang
7	Faizal Roni	Jl. Yapen 2 Blok QB-15 Nusaloka, Perumahan BSD, Serpong Tangerang Selatan , Serpong, Tangerang Selatan

8	Rahayu Swastika	Jl. I Barat No. 9 RT.011/06 Kebon Baru, Tebet Jakarta Selatan 12830, Tebet, Jakarta Selatan
9	Slamet Maryoso	Jl. Beringin 53 RT 01/I Bancong Kec. Wonoasri Kab. Madiun Jawa Timur 63157, Wonoasri, Madiun



Menu dan Paduan Penggunaan

RANCANG BANGUN PROGRAM POINT OF SALES BERBASIS CLIENT SERVER

(Studi Kasus Pada K24 Pharmacy)

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Informasi memegang peranan penting dalam kehidupan manusia. Sistem pengolahan data dengan komputer semakin banyak digunakan untuk pengolahan data dengan cepat, tepat dan relevan dan juga efisiensi kerja. Dunia usaha seperti apotek sangatlah membutuhkan sistem informasi. Banyak berbagai macam obat keluar masuk disebuah Apotek tiap harinya. Untuk mengetahui stok obat yang masih tersedia dalam apotek sangatlah kompleks, harus di bantu dengan kartu stok. Hal ini menyulitkan apoteker untuk mengetahui stok ataupun sirkulasi obat yang ada dalam apotek. Petugas apotek akan mencatat keluar masuknya obat, membuat laporan- laporan persediaan obat, membuat laporan-laporan penjualan obat yang dibuat tiap hari ataupun tiap bulannya. Pembuatan laporan tersebut dilakukan secara manual, akan menyita banyak waktu dan tenaga dalam pembuatan laporan- laporan. Dengan adanya sistem informasi berbasis komputer, nantinya permasalahan yang dikerjakan secara manual dapat terkomputerisasi, sehingga dalam operasional apotek dapat sangat membantu.

1.2 Rumusan Masalah

Dari topik diatas, maka rumusan permasalahan dalam penulisan tugas akhir ini adalah sebagai berikut:

“ Bagaimana membangun sebuah Sistem Informasi Apotek yang dapat membantu operasional Apotek khususnya dalam bidang penjualan? ”

1.3 Batasan Masalah

Batasan masalah tugas akhir sistem informasi Apotek yang dibuat, dibatasi pada sistem proses pelayanan transaksi penjualan obat yang meliputi penjualan obat dengan resep dokter ataupun obat non resep dan pembuatan laporan-laporan harian ataupun bulanan, laporan-laporan ini meliputi :

- Laporan transaksi obat non resep
- Laporan transaksi obat resep
- Laporan obat psikotropika dan narkotika
- Laporan obat keras.
- Laporan obat bebas dan bebas terbatas
- Laporan obat stok minimal/habis

1.4 Tujuan

Tujuan dari pembuatan Sistem Informasi Apotek “ K24 PHARMACY ” Bekasi ini adalah :

1. Meningkatkan efisiensi operasi apotek antara lain petugas dapat dengan mudah membuat laporan penjualan dengan mudah dan cepat.
2. Melibatkan peran serta komputer dalam dunia farmasi khususnya apotek.
3. Meningkatkan manajemen keuangan apotek.
4. Menyediakan laporan-laporan penjualan tepat pada waktunya.
5. Pembuatan perangkat lunak sistem informasi berbasis *Client-Server* pada K24 Pharmacy.

1.5 Metodologi Penelitian

Metodologi yang digunakan dalam penulisan tugas akhir ini adalah pendekatan terstruktur yaitu analisis dan desain terstruktur. Pada pendekatan terstruktur ini, tahap-tahap yang digunakan sebagai berikut :

1. Metodologi Kepustakaan (*Literatur*)

Pengumpulan data yang dilakukan dengan cara membaca referensi berupa buku-buku, laporan-laporan, karya ilmiah, makalah di internet dan sebagainya yang berhubungan dengan penulisan tugas akhir ini.

2. Menetapkan topik masalah.
3. Pengumpulan data

Pada langkah ini penulis melakukan observasi dengan mengamati sistem yang ada, selain itu juga melakukan wawancara langsung dengan apoteker dan juga mengumpulkan dokumen atau bahan yang yang diperlukan.

4. Analisis Sistem (*Systems Analysis*)

Tahap ini penulis menganalisa sistem apotek yang ada dan mengembangkan menjadi sistem yang terkomputerisasi. Cara yang digunakan adalah dengan mengidentifikasi kebutuhan dan mengumpulkan data yang diperlukan sesuai dengan kebutuhan sistem yang nantinya akan dikembangkan. Alat yang digunakan dengan diagram arus data dan HIPO diagram.

5. Perancangan sistem (*Systems Design*)

Perancangan sistem apotek yang terkomputerisasi nantinya akan meliputi perancangan input, perancangan output, perancangan database. Untuk desain basis data alat yang digunakan secara konseptual (*Conceptual Database Design*) menggunakan diagram ER, desain basis data secara logika (*Logical Database Design*) menggunakan normalisasi database, desain fisik (*Physical Database Design*) yang merupakan penjabaran dari konsep desain logika secara terinci dengan bentuk implementasi dengan menggunakan sistem pengolahan *database*.

6. Implementasi

Mengimplementasikan rancangan program sistem informasi apotek ke dalam pemrograman berbasis data. Program aplikasi yang digunakan untuk pengembangan Point of Sales pada Apotek ini, menggunakan *Microsoft Visual Basic 6.0* dan DBMS dari *Microsoft SQL Server 2005*

7. Pengujian (*Testing*)

Tahap ini menfokuskan pada pengujian program yang dibuat.

BAB II

LANDASAN TEORI

2.1 KONSEP DASAR SISTEM

Terdapat dua kelompok dalam mendefinisikan sistem, yaitu menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Menurut Jerry FitzGerald, Ardra F. FitzGerald dan Warren D. Stalling, Jr., suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi didalam sistem. Prosedur (*procedure*) didefinisikan oleh Ricard F. Neuschel sebagai berikut. Suatu prosedur adalah suatu urutan operasi klerikal (tulis menulis), biasanya melibatkan orang didalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi. Menurut Jerry FitzGerald, Ardra F. FitzGerald dan Warren D. Stalling, Jr., Suatu prosedur adalah urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa (*what*) yang harus dikerjakan, siapa (*who*) yang mengerjakan, kapan (*when*) dikerjakan dan bagaimana (*how*) mengerjakannya.

Pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sebagai berikut ini.

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu. Pendekatan sistem yang merupakan kumpulan-kumpulan dari elemen-elemen atau komponen-komponen atau subsistem-subsistem merupakan definisi yang lebih luas. Definisi ini lebih banyak diterima, karena kenyataannya suatu sistem dapat terdiri dari beberapa subsistem atau sistem bagian. Sebagai misal sistem apotek dapat terdiri dari subsistem-subsistem, yaitu subsistem apotek penjualan, subsistem apotek pembelian, subsistem apotek penggajian dan lain sebagainya. Komponen-komponen atau subsistem-subsistem dalam suatu sistem tidak dapat berdiri sendiri. Komponen-komponen atau subsistem-subsistem saling berinteraksi dan saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai.

Suatu sistem mempunyai maksud tertentu. Maksud dari suatu sistem adalah untuk mencapai suatu tujuan (*goal*) dan ada juga yang menyebutkan bahwa sistem untuk mencapai suatu sasaran (*objectives*). Goal biasanya dihubungkan dengan ruang lingkup yang lebih luas dan sasaran dalam ruang lingkup yang lebih sempit.

2.1.1. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat sifat tertentu, yaitu mempunyai komponen-komponen (*komponents*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*) dan sasaran (*objectives*) atau tujuan (*goal*).

- **Komponen Sistem**

Suatu sistem terdiri dari sejumlah sistem yang saling berinteraksi, yang artinya saling bekerjasama membentuk suatu kesatuan. Komponen- komponen sistem atau elemen- elemen sistem dapat berupa suatu subsistem atau bagian bagian dari sistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu. Dan mempengaruhi suatu proses secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yang disebut supra sistem. Misalnya suatu perusahaan dapat disebut sebagai sistem dan industri yang merupakan sistem yang lebih besar dapat disebut dengan supra sistem. Kalau dipandang industri sebagai sistem maka perusahaan dapat disebut sebagai subsistem. Demikian juga bila perusahaan dipandang sebagai suatu sistem, maka sistem akutansi adalah subsistemnya. Kalau sistem akutansi dipandang sebagai suatu sistem, maka perusahaan adalah supra sistem dan industri adalah supra dari supra sistem.

- **Batas Sistem**

Batas sistem (*Boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

- **Lingkungan Luar Sistem**

Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluarbatas dari sistem yang mempengaruhi sistem. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

- **Penghubung Sistem**

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya membentuk satu kesatuan. Keluaran (*output*) dari suatu sistem akan menjadi masukan (*input*) untuk subsistem yang lainnya melalui penghubung.

- **Masukan Sistem**

Masukan (*input*) adalah energi yang dimasukkan kedalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukkan sinyal input (*signal input*). Maintenance input adalah energi yang masuk supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh dalam program komputer, program adalah maintenance input yang digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah menjadi informasi.

- **Keluaran Sistem**

Keluaran (*output*) adalah hasil dari energi yang diolah dan diklarifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.

- **Pengolah Sistem**

Suatu sistem dapat menjadi suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

- **Sasaran Sistem**

Suatu sistem pastilah mempunyai tujuan (*goal*) atau sasaran (*objektive*). Sasaran dari suatu sistem sangat menentukan sekali, masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem akan berhasil bila mengenai sasaran atau tujuannya.

2.2 KONSEP DASAR INFORMASI

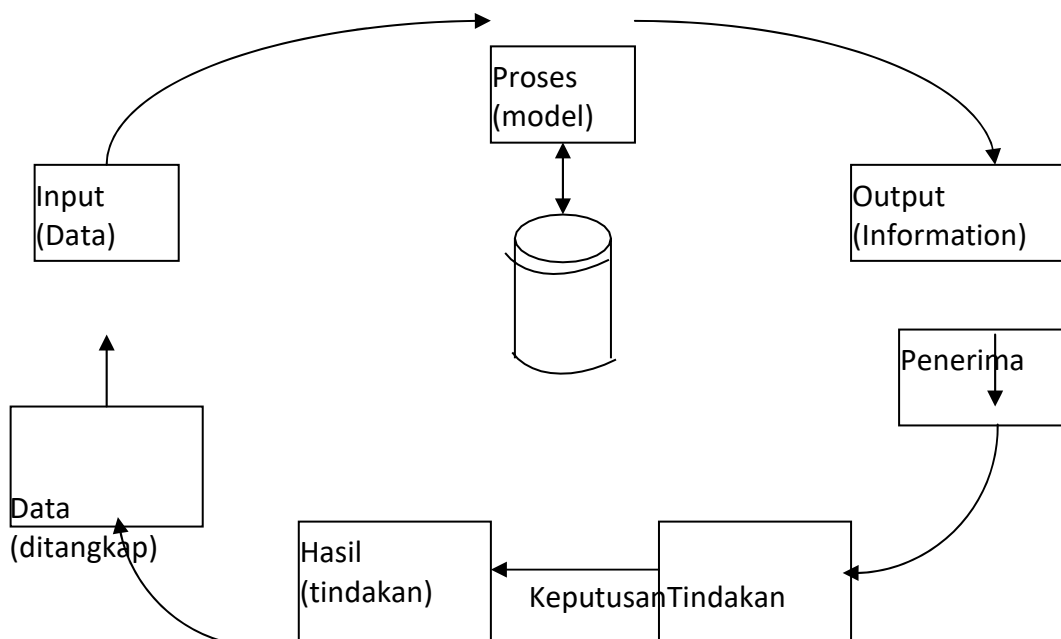
Suatu sistem yang kurang mendapatkan informasi akan menjadi luruh, kerdil dan akhirnya terakhir. Robert N. Anthony dan John Dearden menyebut keadaan dari sistem dalam hubungannya dengan berakhirnya dengan istilah *entropy*. Informasi yang berguna bagi sistem akan menghindari proses *entropy* yang disebut dengan *negative entropy* atau *negentropy*.

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah suatu yang terjadi pada saat yang tertentu. Kesatuan nyata (*fact dan entity*) adalah berupa suatu obyek nyata seperti tempat, benda dan orang yang betul-betul ada dan terjadi.

2.2.1. Siklus Informasi

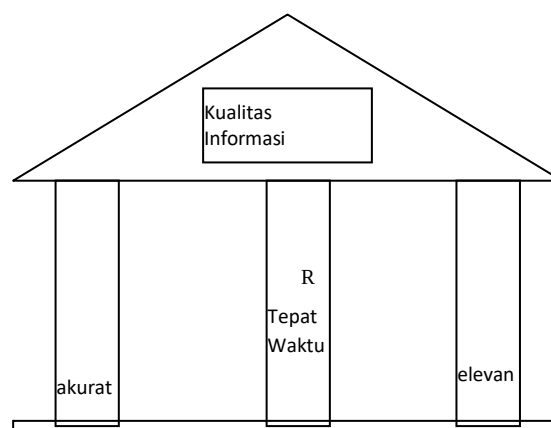
Data merupakan bentuk yang masih mentah yang belum dapat berceritera banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu medel untuk menghasilkan informasi. Data yang diolah menghasilkan suatu informasi menggunakan suatu model proses yang tertentu. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus ini oleh John Burch disebut dengan siklus informasi (*informaiton cycle*). Sikklus ini juga disebut dengan siklus pengolahan data (*data procesing cycle*).



Gambar. 2.1. Siklus informasi

2.2.2. Kualitas Informasi

Kualitas dari informasi (*quality of information*) tergantung dari tiga hal, yaitu informasi yang akurat (*accurate*), tepat pada waktunya (*timeliness*) dan relevan (*relevance*). John Burch dan Gary Grutnitski menggambarkan kualitas dari informasi dengan bentuk bangun yang ditunjang oleh tiga buah pilar.



Gambar 2.2. Pilar kualitas Informasi

- **Akurat**, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Dari sumber informasi ke penerima kemungkinan banyak gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut untuk itu informasi haruslah akurat.
- **Tepat pada waktunya**, berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang usang tidak ada nilainya, karena informasi merupakan landasan dalam pengambilan keputusan.

- **Relevan**, berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

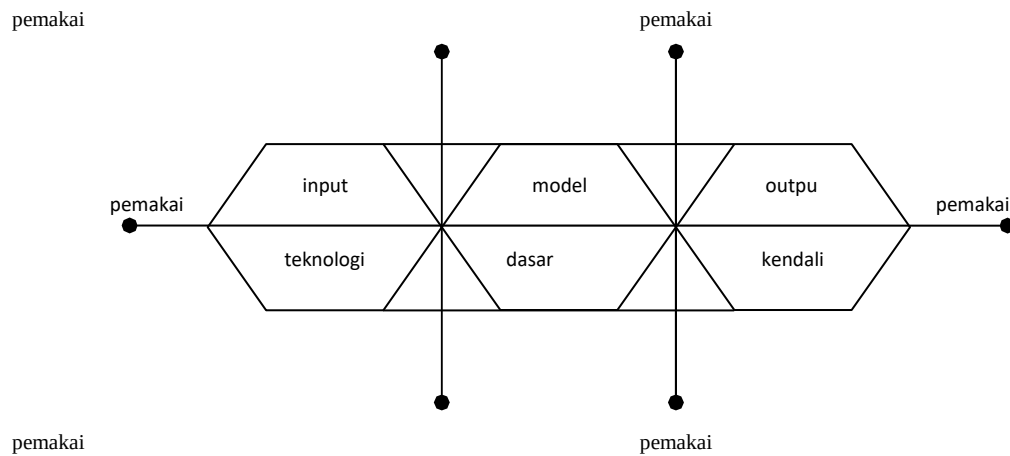
2.3 KONSEP DASAR SISTEM INFORMASI

Sistem informasi didefinisikan oleh Robert A. Leitch dan K. Roscoe Davis sebagai berikut :

Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi , bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

2.3.1. Komponen Sistem Informasi

John Burch dan Gary Grudnitski komponen sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah block bangunan (*building block*), yaitu blok masukan (*input block*), block model (*model block*), block keluran (*output block*), block teknologi (*technology block*), blok basis data (*database block*), dan blok kendali (*controls block*). Sebagai suatu sistem keenam blok tersebut saling berinteraksi membentuk satu kesatuan untuk mencapai sasarnya.



Gambar 2.3. Blok sistem informasi yang berinteraksi

- **Blok Masukan**

Input mewakili data yang masuk kedalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

- **Blok Model**

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

- **Blok Keluaran**

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

- **Blok Teknologi**

Teknologi merupakan alat (*tool-box*) dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari tiga bagian utama, yaitu teknisi (*humanware atau brainware*), perangkat lunak (*software*) dan perangkat keras (*hardware*).

- **Blok Kendali**

Banyak hal dapat merusak sistem informasi, seperti misalnya bencana alam, kecurangan-kecurangan, kesalahan-kesalahan, sabotase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah, ataupun bila terlanjur kesalahan-kesalahan dapat langsung cepat diatasi.

2.3.2. Sistem Informasi Manajemen

Sistem Informasi Manajemen (*Management Information Systems* atau sering dikenal sebagai *MIS*) merupakan penerapan sistem informasi didalam organisasi untuk mendukung informasi-informasi yang dibutuhkan oleh semua tingkatan manajemen. SIM (Sistem Informasi Manajemen) didefinisikan oleh George M. Scott sebagai berikut :

Suatu SIM adalah kumpulan dari interaksi-interaksi sistem-sistem informasi yang menyediakan informasi baik untuk kebutuhan manajerial maupun kebutuhan operasi.

Menurut Barry E. Cushing :

Suatu SIM adalah kumpulan dari manusia dan sumber-sumber daya modal didalam suatu organisasi yang bertanggung jawab mengumpulkan dan mengolah data untuk menghasilkan informasi yang berguna untuk semua tingkatan manajemen didalam kegiatan perencanaan dan pengendalian.

Menurut Frederick H. Wu :

SIM adalah kumpulan dari sistem-sistem yang menyediakan informasi untuk mendukung manajemen.

Menurut Gordon B. Davis :

SIM adalah sistem manusia / mesin yang menyediakan informasi untuk mendukung operasi manajemen dan fungsi pengambilan keputusan dari organisasi.

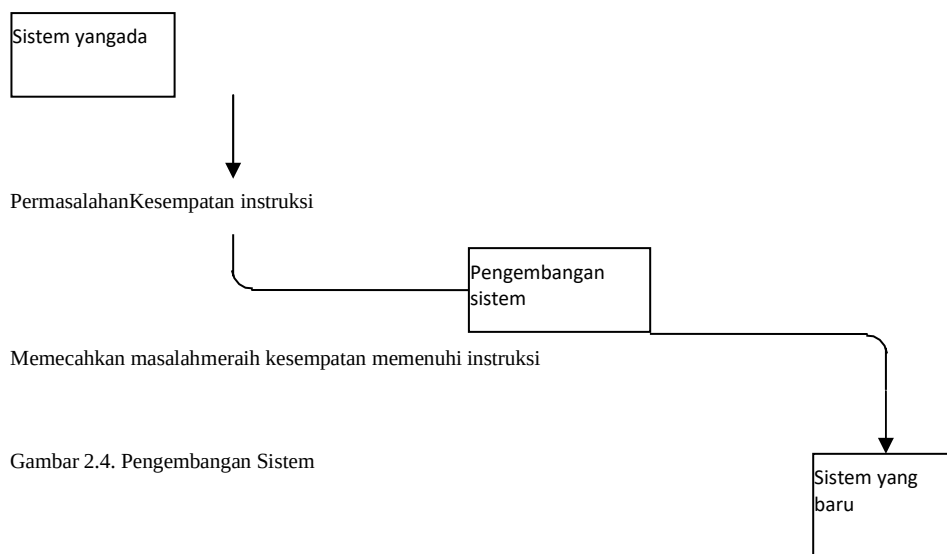
Dari definisi yang diberikan oleh Gordon B. Davis, elemen non- komputer adalah sistem manusia dan elemen komputer adalah mesin. Gordon B. Davis juga menegaskan bahwa SIM juga berhubungan dengan pengolahan informasi yang berbasis pada komputer (*computer-based information processing*). Dari beberapa definisi tersebut diatas dapat dirangkum bahwa MIS :

1. Kumpulan dari interaksi sistem-sistem informasi.
2. Menghasilkan informasi yang berguna untuk semua tingkatan manajemen.

SIM merupakan suatu sistem yang melakukan fungsi-fungsi untuk menyediakan semua informasi yang mempengaruhi semua operasi organisasi. SIM merupakan kumpulan dari sistem-sistem informasi.

2.4 PERLUNYA PENGEMBANGAN SISTEM

Pengembangan sistem (*system development*) dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang ada. Karena adanya permasalahan, kesempatan atau instruksi, maka sistem yang baru perlu dikembangkan untuk memecahkan permasalahan-permasalahan yang timbul, meraih kesempatan-kesempatan yang ada atau memenuhi instruksi yang diberikan.



Gambar 2.4. Pengembangan Sistem

Dengan telah dikembangkan sistem yang baru, maka diharapkan akan terjadi peningkatan-peningkatan di sistem yang baru. Peningkatan-peningkatan ini berhubungan dengan :

1. *Performance* (kinerja) peningkatan terhadap kinerja (hasil kerja) sistem yang baru sehingga menjadi lebih efektif. Kinerja dapat diukur dari *throughput* dan *respon time*. *Throughput* adalah jumlah pekerjaan yang dapat dilakukan suatu saat tertentu. *Respon time* adalah rata-rata waktu yang

tertunda diantara dua transaksi atau pekerjaan ditambah dengan waktu *respon* untuk menanggapi pekerjaan tersebut.

2. *Information* (informasi), peningkatan terhadap kualitas informasi yang disajikan.
3. *Economy* (ekonomis), peningkatan terhadap manfaat-manfaat atau keuntungan-keuntungan atau penurunan-penurunan biaya yang terjadi. Ekonomis berhubungan dengan jumlah sumber daya yang digunakan.
4. *Control* (pengendalian), peningkatan terhadap pengendalian untuk mendeteksi dan memperbaiki kesalahan-kesalahan serta kecurangan kecurangan yang dan akan terjadi.
5. *Efficiency* (efisiensi), peningkatan terhadap efisien operasi. Efisien berhubungan dengan bagaimana sumber daya tersebut digunakan dengan pemborosan yang paling minimum. Efisien dapat diukur dari outputnya dibagi dengan input.
6. *Service* (pelayanan), peningkatan terhadap pelayanan yang diberikan oleh sistem.

2.5 SIKLUS HIDUP PENGEMBANGAN SISTEM

Proses pengembangan sistem melewati beberapa tahapan dari mulai sistem itu direncanakan sampai sistem tersebut diterapkan, dioperasikan dan dipelihara. bila operasi sistem yang dikembangkan masih timbul kembali permasalahan- permasalahan yang kritis serta tidak dapat diatasi dalam tahap pemeliharaan sistem, maka perlu dikembangkan kembali suatu sistem untuk mengatasi dan

proses ini kembali ke tahap pertama, yaitu tahap perencanaan sistem. Siklus ini disebut dengan siklus hidup suatu sistem (*systems life cycle*). Daur atau siklus hidup dari pengembangan sistem merupakan suatu bentuk yang digunakan untuk menggambarkan tahap utama dan langkah-langkah didalam tahap tersebut dalam proses pengembangannya.

Di *systems life cycle*, tiap-tiap dari pengembangan sistem dibagi menjadi beberapa tahap kerja. Tiap-tiap tahapan ini mempunyai karakteristik tersendiri. Tahap utama siklus pengembangan sistem dapat terdiri dari tahapan perencanaan sistem (*systems planing*), analisis sistem (*systems analysis*), desain sistem (*systems design*), seleksi sistem (*systems selection*), implementasi sistem (*systems implementation*) dan perawatan sistem (*systems maintenance*). Tahap-tahap ini sebenarnya merupakan tahapan dalam pengembangan sistem teknik (*engineering systems*). Tampak bahwa daur hidup dari sistem teknik dan sistem informasi dapat sama atau mirip. Hal ini tidaklah kebetulan, karena proses pengembangan sistem informasi adalah proses teknik dan proses semacam ini harus mengikuti langkah-langkah yang sama serta prinsip-prinsip umum dari sistem teknik. Istilah *software engineering* merupakan proses pengembangan perangkat lunak yang merupakan subsistem dari pengembangan sistem informasi. Jika ditinjau kembali siklus pengembangan sistem sejak tahun 1970 yang diusulkan oleh beberapa penulis, maka akan didapatkan kesamaan didalam siklus teknik.

Siklus pengembangan sistem menurut J. F. Kelly, *Computerized Management Information Systems*, (MacMilan, 1970), dikutip oleh A. Ziya Aktas, *Structured Analysis & Design of Information Systems*, (NJ: Prentice-Hall, 1987):

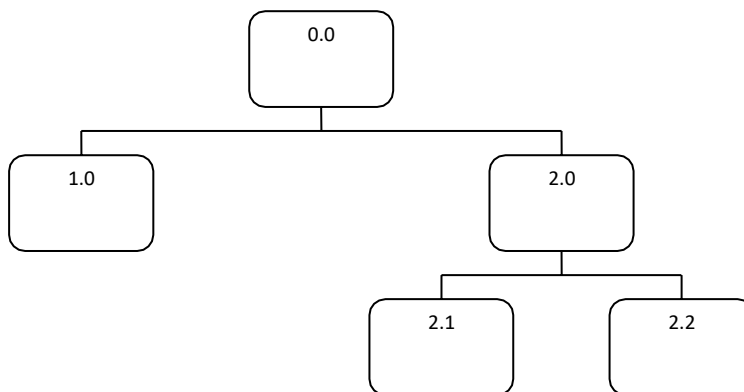
1. Penelitian sistem (system survey).
 - a. Definisi ruang-lingkup (scope definition).
 - b. Studi penelitian (survey study).
2. Analisis dan desain sistem (system analysis and design).
 - a. Studi penelitian (survey study).
 - b. Pengumpulan data dan analisis (data collection and analysis).
 - c. Desain sistem (system design).
 - d. Rencana implementasi (implementation planning).
3. Pengembangan sistem (system development).
 - a. Pengembangan (development).
 - b. Pengetesan (testing).
 - c. Pengoperasian sistem baru (cutover).
 - d. Perawatan (maintenance).

2.5.1. Alat dan Teknik dalam Pengembangan Sistem

Pengembangan sistem akan membutuhkan alat bantu desain yang digunakan untuk membantu analisis dan perancangan sistem yang akan dikembangkan. Alat-alat pengembangan sistem yang berbentuk grafik (Jogiyanto 1990) diantaranya adalah sebagai berikut :

a) HIPO Diagram

HIPO (*Hierarchy plus Input-Proses-Output*) adalah alat dokumentasi program, khususnya sebagai alat desain dan teknik dokumentasi dalam siklus pengembangan sistem. Diagram pada Gambar 2.5 menggambarkan hubungan dari fungsi-fungsi pada sistem secara berjenjang.

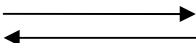


Gambar 2.5. HIPO

b) *Data flow Diagram (DFD)*

DFD adalah diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem. Notasi-notasi tersebut dapat dilihat pada Gambar 2.6. seperti berikut :

Gambar kesatuan luar 

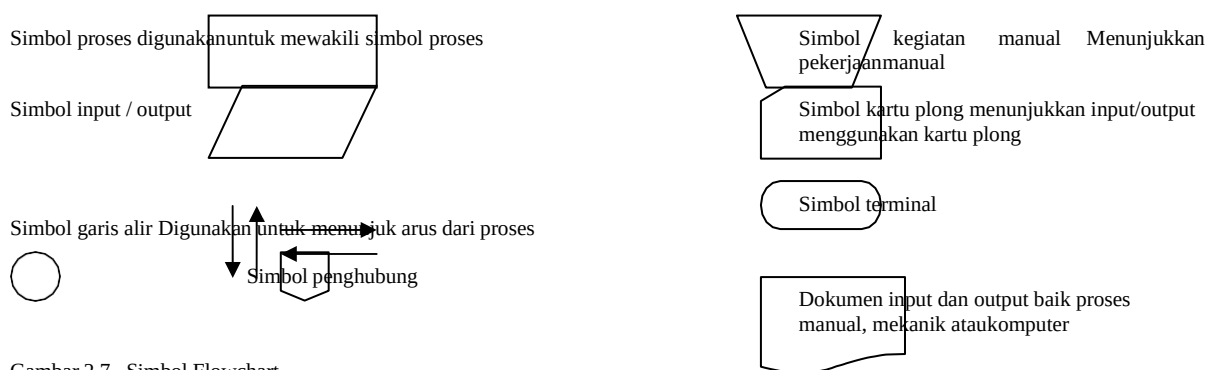
Gambar arus data 

Gambar notasi proses 

Gambar notasi penyimpanan  Gambar 2.6. Notasi DFD

c) Sistem *Flowchart*

Bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Simbol-simbol yang digunakan dalam sistem *flowchart* :



Gambar 2.7. Simbol Flowchart

2.5.2. Perencanaan Sistem

Sebelum sistem informasi dikembangkan umumnya dimulai dengan adanya suatu perencanaan untuk mengembangkan sistem itu. Tanpa adanya sistem perencanaan yang baik, pengembangan sistem tidak dapat berjalan sesuai dengan yang diharapkan. Tanpa adanya kebijakan pengembangan sistem oleh manajemen puncak (*top management*), dukungan dari manajemen puncak ini maka pengembangan sistem tidak akan mendapat dukungan dari manajemen puncak ini. Dukungan dari manajemen puncak sangatlah penting. Kebijakan sistem (*system policy*) merupakan landasan dan dukungan dari manajemen puncak untuk membuat perencanaan sistem. Perencanaan sistem (*system planning*) merupakan pedoman untuk melakukan pengembangan sistem.

Kebijaksanaan untuk mengembangkan sistem informasi dilakukan oleh manajemen puncak karena manajemen menginginkan untuk meraih kesempatan-kesempatan yang ada yang tidak dapat diraih oleh sistem yang lama, atau sistem yang lama mempunyai banyak kelemahan-kelemahan yang perlu diperbaiki (misalnya meningkatkan efektivitas manajemen, meningkatkan pelayanan yang lebih baik kepada pelanggan).

2.5.3. Analisis Sistem

Analisis sistem (*systems analysis*) dapat didefinisikan sebagai :

Penguraian dari sebuah sistem informasi yang utuh kedalam bagian-bagiannya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

Tahap analisis sistem dilakukan setelah tahap perencanaan sistem (*systems planning*) dan sebelum tahap desain sistem (*systems design*). Tahap analisis merupakan tahap yang kritis dan sangat penting, karena kesalahan ditahap ini akan menyebabkan juga kesalahan ditahap selanjutnya. Analisis sistem dalam hal ini, yaitu menganalisis sistem untuk menemukan kelemahan-kelemahannya, sehingga dapat diusulkan perbaikannya.

Didalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analis sistem sebagai berikut.

1. Identify, yaitu mengidentifikasi masalah.
2. Understand, yaitu memahami kerja dari sistem yang ada.

3. Analyze, yaitu menganalisis sistem.
4. Report, yaitu membuat laporan hasil analisis.

2.5.4. Desain Sistem

Setelah tahap analisis sistem selesai dilakukan, maka analisis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Tiba waktunya untuk memikirkan bagaimana membentuk sistem tersebut. Tahap ini disebut dengan desain sistem (*design systems*). Desain sistem dapat dibagi menjadi dua bagian, yaitu desain secara umum (*general systems design*) dan desain sistem terinci (*detailed systems design*). Desain sistem secara umum (*general systems design*) disebut juga dengan desain konseptual (*conceptual design*) atau desain logikal (*logical design*) atau desain secara makro (*macro design*). Desain terinci disebut juga dengan desain secara fisik (*physical systems design*) atau desain internal (*intenal design*).

Desain sistem dapat didefinisikan sebagai berikut ini : Menurut

Robert J. Versello/John Router III :

Tahap setelah analisis dari siklus pengembangan sistem; pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi; menggambarkan bagaimana suatu sistem dibentuk.

Menurut John Burch & Gary Grudnitski:

Desain sistem dapat didefinisikan sebagai gambaran, perancangan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi.

Menurut George M. Scott :

Desain sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti di selesaikan, tahap ini menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah instalasi dari sistem akan benar-benar memuaskan rancang bangun yang telah di tetapkan pada akhir tahap analisis sistem.

Desain sistem dapat diartikan sebagai berikut :

1. tahap setelah analisis dari siklus pengembangan sistem;
2. pendefinisian dari dari kebutuhan kebuuhan fungsional;
3. persiapan rancang bangun untuk implementasi;
4. menggambarkan bagaimana sistem informasi di bentuk;
5. yang dapat berupa gambaran, perencanaan dan pembuatan skets atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi;
6. termasuk menyangkut konfigurasi dari kompomen-komponen perangkat lunak dan perangkat keras dari suatu sistem.

2.5.4.1. Desain Sistem Secara Umum

Desain secara umum bertujuan untuk memberikan gambaran secara umum kepada user tentang sistem yang baru. Desain ini mengidentifikasi komponen-komponen sistem informasi yang akan didesain secara terinci. Desain terinci dimaksudkan untuk program komputer dan ahli teknik lainnya yang akan mengimplementasi sistem. Pada tahap desain secara umum, komponen-komponen

sistem informasi yang didesain adalah model, output, input, database, teknologidan kontrol.

2.5.4.2.Desain Model Secara Umum.

Analisis sistem dapat mendesain model dari sistem informasi yang diusulkan dalam bentuk physical systems dan logical model. Bagan alir sistem (*systems flowchart*) merupakan alat yang tepat digunakan untuk menggambarkan physical systems. Simbol-simbol bagan alir sistem ini menunjukkan secara tepat arti fisiknya, seperti simbol terminal, hard disk, laporan-laporan.

Logical model dari sistem informasi dapat digambar menggunakan diagram arus data (*data flow diagram*). Arus dari data di DAD dapat dijelaskan dengan menggunakan kamus data (*data dictionary*). Pengolahan data dari sistem informasi berbasis komputer membutuhkan metode-metode dan prosedur- prosedur yang akan mendefinisikan urutan kegiatan untuk menghasilkan output dan input yang ada. Metode adalah suatu cara untuk melakukan kegiatan. Suatu prosedur merupakan rencana tahap demi tahap untuk menerapkan suatu metode. Bagan alir sistem (*system flow chart*) merupakan alat berbentuk grafik yang dapat digunakan untuk menunjukan urutan kegiatan dari sistem informasi berbasis komputer.

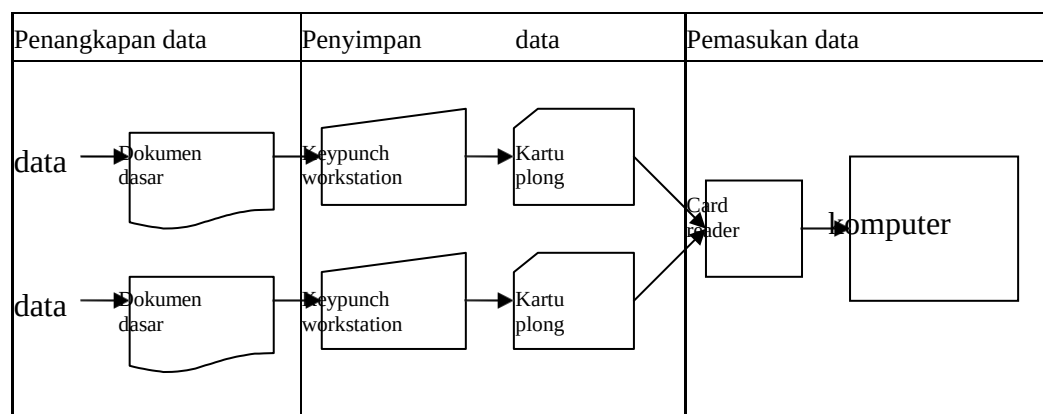
2.5.4.3.Desain Output Secara Umum

Output (keluaran) adalah produk dari sistem informasi yang dapat dilihat. Output dapat berupa hasil di media keras (seperti misalnya kertas, microfilm) atau hasil di media lunak (berupa tampilan di layar video). Output dapat berupa hasil

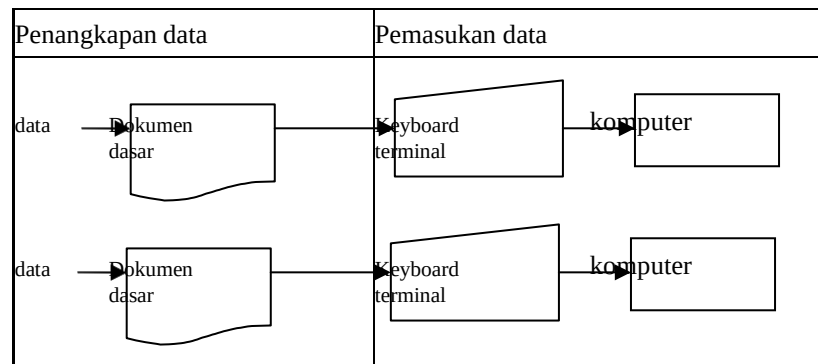
dari suatu proses yang akan digunakan oleh proses lain dan tersimpan di suatu media seperti tape, disk atau kartu.

2.5.4.4. Desain Input Secara Umum

Alat input dibagi menjadi 2 golongan, yaitu alat input langsung (*online input device*) dan alat input tidak langsung (*off line input device*). Alat input langsung merupakan alat input yang langsung dihubungkan dengan CPU, misalnya adalah keyboard, mouse. Alat input tidak langsung adalah alat input yang tidak langsung dihubungkan dengan CPU, misalnya KTC (*key-to-card*), KTT (*key-to-disk*). Input tidak langsung mempunyai tiga tahapan utama yaitu data capture, data preparation dan data entry seperti pada gambar 2.8. input yang menggunakan alat input langsung terdiri dari 2 tahapan utama yaitu data capture dan data entry, seperti terlihat seperti pada Gambar 2.9.



Gambar 2.8. proses input yang menggunakan alat input tidak langsung



Gambar 2.9. proses input yang menggunakan alat input langsung

2.5.4.5 Desain Database Secara Umum

Basis data (database) adalah sebuah objek untuk menyimpan informasi yang kompleks untuk menyimpan informasi yang terstruktur, yang diorganisir dan disimpan dalam suatu cara yang memungkinkan pemakainya dapat mengambil informasi dengan cepat dan efisien. Informasi disimpan dan diambil dari database oleh sebuah program yang disebut dengan database management system (DBMS). DBMS mengelola semua informasi di dalam database, aplikasi bisa mengakses informasi ini melalui pernyataan yang dibuat dengan Structured Query Language (SQL), sebuah bahasa yang menentukan operasi high-level. Operasi ini disebut

dengan query, dan ada dua jenis query; selection query, yang mengambil informasi dari database, dan action query, yang meng-update database.

2.5.4.6. Database Relasional

Database relasional didasarkan pada hubungan antara data-data yang dikandungnya. Database disimpan di dalam tabel, dan tabel mengandung data yang berhubungan, atau entity. Tujuannya adalah untuk menjaga tabel tetap kecil dan

dapat dikelola, serta entity-entity yang terpisah disimpan di dalam tabel-tabel tersendiri.

2.5.4.7. Entity/Relationship (E-R).

Entity Relation Model adalah model yang termasuk dalam *conceptual design* yang mampu menyediakan gagasan yang menggambarkan kebutuhan data dari suatu aplikasi agar mudah dimengerti dan bebas dari kriteria dalam hal pengolahan dan pengorganisasian data pada suatu sistem.

Entity Relation Model menggambarkan database dalam bentuk tabel dan dimensi, dimana setiap kolom sebagai attribte. Attribute menunjukkan item data atau field. Kumpulan nilai dari item data atau field disebut dengan istilah domain.

Tuple (*record*) adalah masing-masing baris dari record didalam tabel. Tiap-tap tuple dapat mempunyai suatu kunci yang unik, field yang menjadi kunci yang unik disebut dengan field kunci (*key field*).

Suatu tuple yang identik sama dengan tuple lain tidak diperbolehkan didalam suatu tabel.

Jadi harus ada suatu kunci yang disebut dengan *primary key*. *Primary key* adalah suatu kolom dimana tidak ada 2 baris yang memiliki nilaisama dalam kolom tersebut. Setiap tabel dapat memiliki lebih dari satu kunci dan kunci-kunci tersebut dikenal sebagai *secondary key* dan *primary key*. *Secondary key* adalah nilai attribut yang mungkin mempunyai unsur yang tidak unik tetapi digunakan juga untuk mengidentifikasi suatu tuple. *Foreign key* adalah nilai

attribute yang bukan merupakan *primary key* pada suatu tuple namun kunci tersebut merupakan *primary key* bagi tuple dalam tabel lain.

Sifat-sifat *Entity Relation Model* (Ir. Harianto Kristanto. 1994,35)

- a. Tidak ada duplikasi baris/ tidak ada baris yang sama
- b. Urutan baris tidak signifikan (urutan bebas).
- c. Urutan kolom tidak signifikan (urutan bebas)
- d. Nilai data bersifat atonik

2.5.4.8. Normalisasi

Ada beberapa aturan dalam perancangan database, yang disebut dengan aturan normalisasi. Aturan–aturan ini akan membantu dalam merancang data base yang normal. Database dianggap normal jika ia tidak mengulangi informasi atau tidak menimbulkan keanehan pada proses update atau penghapusan. Walaupun jumlah aturan ini bervariasi aturan dasar normalisasi sebenarnya hanya ada tiga : aturan normalisasi normal pertama, kedua, dan ketiga. Tabel yang dinormalisasi berdasarkan pada aturan pertama disebut dengan istilah *first normal form* (1NF).

Tabel yang dinormalisasi berdasarkan pada aturan kedua disebut dengan istilah *secon normal form* (2NF). Yang terakhir, tabel yang berbentuk 2NF bisa dinormalisasi berdasarkan aturan ketiga, dalam hal ini akan disebut dengan istilah *third normal form* (3NF). Bentuk normal yang lebih tinggi menangani masalah yang spesifik dan pada situasi yang jarang.

Tujuan utama dalam perancangan database adalah menghindari adanya duplikasi informasi. Bentuk normal pertama aturannya sebuah tabel tidak boleh mengandung kelompok yang berulang. Bentuk normal pertama ini tidak membutuhkan tabel dipecah-pecah kedalam banyak tabel. Ia hanya mengubah beberapa kolom tabel menjadi baris-baris tambahan. Bentuk normal kedua, aturan

normalisasi kedua bahwa setiap field yang tidak bergantung sepenuhnya pada kunci primer harus dipindah ke tabel lain. Bentuk normal ketiga, aturan normalisasi ketiga adalah bahwa tidak ada kebergantungan antara field-field non kunci.

2.5.4.9.DAD (Diagram Arus Data)

Terdapat 2 bentuk DAD (Diagram Arus Data), yaitu diagram arus data fisik (*physical data flow diagram*) dan diagram arus data logika (*logical data flow diagram*). Diagram arus data fisik lebih menekankan pada bagian proses dari sistem diterapkan sedang diagram arus data logika lebih menekankan proses-proses apa yang terdapat di sistem.

1. Diagram Arus Data Fisik

Diagram arus data fisik (DADF) lebih tepat untuk menggambarkan sistem yang ada (sistem yang lama). Penekanan dari DADF adalah bagaimana proses-proses dari sistem diterapkan (dengan cara apa, oleh siapa dan dimana), termasuk proses-proses manual.

2. Diagram Arus Data Logika

Diagram arus data logika (DADL) lebih tepat digunakan untuk menggambarkan sistem yang akan diusulkan (sistem yang baru). DADL tidak menerapkan sistem diterapkan, tetapi menekankan hanya pada logika dari kebutuhan-kebutuhan sistem, yaitu proses-proses apa secara logika yang dibutuhkan oleh sistem. Untuk sistem komputerisasi, penggambaran DADL yang hanya menunjukkan kebutuhan proses dari sistem yang

diusulkan secara logika, proses yang digambarkan hanya merupakan proses-proses secara komputer saja.

2.6 CLIENT - SERVER

Server adalah pemilik informasi yang menyediakan dirinya untuk memberikan *service* atau layanan, sedangkan *Client* berperan sebagai peminta layanan tersebut.

Secara fisik *Client-Server* dibagi menjadi 3 bagian yaitu

- *Client machine* yang merupakan sistem operasi komputer yang menjalankan aplikasi *client*. *Client* melakukan satu atau lebih permintaan layanan.

- *Server machine* yang merupakan sistem operasi komputer yang menjalankan aplikasi komponen *server*. *Server machine* memenuhi permintaan layanan *client* dan mengirimkannya ke *client machine*.
- *Communication Network* yang merupakan fasilitas komunikasi yang memungkinkan satu atau lebih permintaan dikirim dari *Client* ke *Server* dan memungkinkan satu atau lebih pesan yang berisi layanan dikirim dari *Server* ke *Client*.

Kelebihan model *client-server* adalah adanya kemampuan untuk memecah proses menjadi lebih dari satu bagian yaitu *user interface*, *business logic*, dan *data service* dimana antar bagian tersebut harus tetap dapat saling bekerja sama.

2.6.1. Arsitektur *Client-Server*

Arsitektur adalah desain sebuah aplikasi. Arsitektur *Client-Server* merupakan arsitektur terdistribusi yang berarti pemrosesannya dari suatu aplikasi terjadi pada lebih dari satu mesin. Dalam aplikasi *Client-Server* aplikasi dipecah menjadi komponen-komponen yang bekerja sama untuk mencapai tujuan.

Komponen itu disebut *tier* atau tingkat.

Ada dua model arsitektur *Client-server* yaitu *model two tier* dan *model three tier*.

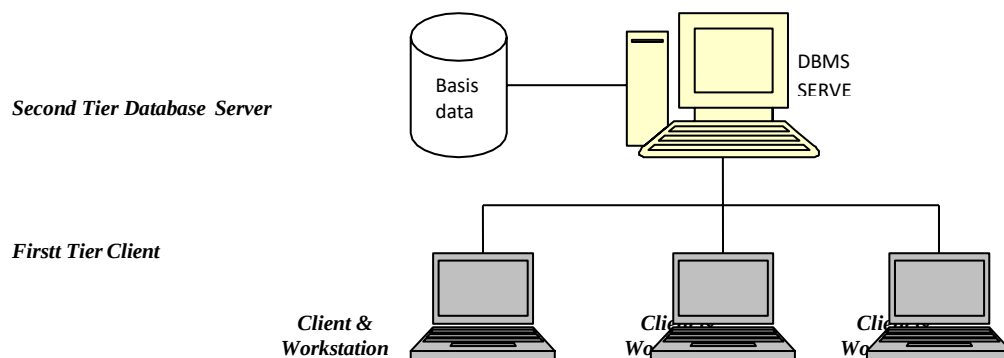
2.6.1.1. Arsitektur *Two Tier*

Terdiri atas sebuah *Server* dan banyak *client* yang dihubungkan melalui jaringan. Aplikasi ditempatkan pada komputer *client* dan mesin database diletakkan pada *server*. *Tier client* disebut juga tingkat presentasi. *Tier client* berisi kode yang akan menampilkan data dan interaksi dengan *user*. Tugas yang terjadi pada *client* adalah antarmuka pengguna, interaksi database. Pengambilan dan modifikasi data, aturan bisnis, dan pengaturan kesalahan.

Tier kedua disebut *Server database* yang bertugas menangani manajemen data, keamanan, *query*, *trigger*, prosedur tersimpan dan penanganan kesalahan. *Model two tier* ini memiliki beberapa keterbatasan yaitu kurangnya skalabilitas (seberapa baik suatu aplikasi bisa menangani kebutuhan yang meningkat), koneksi database yang harus selalu dijaga untuk masing-masing *client* sehingga menghabiskan sumber daya *server*, tidak adanya keterbaruan kode, dan tidak adanya tingkat menengah untuk menangani keamanan dan transaksi.

Arsitektur tradisional *two tier* menyediakan pembagian kerja sebagai berikut

1. *Client (tier 1)* bertanggung jawab terhadap presentasi data kepada user
2. *Server (tier 2)* bertanggung jawab mensuplai layanan data kepada client



Gambar 2.10 Arsitektur *two tier*

Layanan presentasi menangani kegiatan presentasi dan aplikasi bisnis utama sedangkan layanan data menyediakan aplikasi bisnis terbatas khususnya dalam hal validasi dimana *Client* tidak mampu melaksanakan akibat minimnya informasi yang ada pada *Client*.

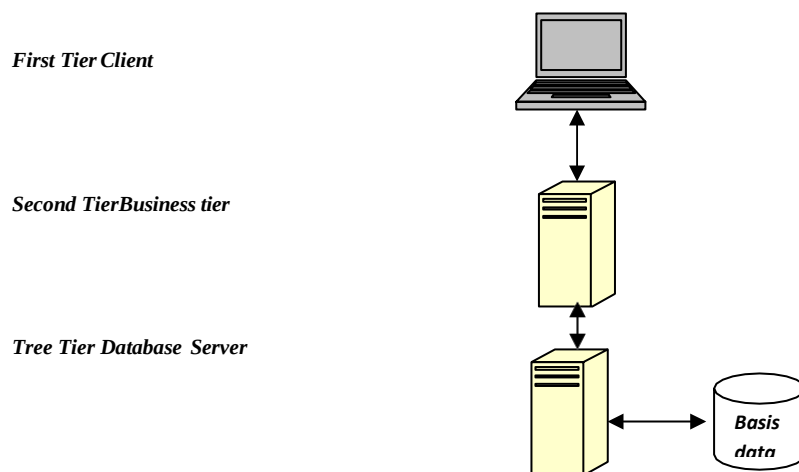
2.6.1.2. Arsitektur *Three Tier*

Untuk menjawab keterbatasan *model two tier* dikembangkan *model three tier* yang memiliki tiga tingkat yaitu tingkat *client (user interface tier)*, tingkat menengah (*business logic tier*), dan tingkat sumber data (*database tier*). Keunggulan *model three tier* adalah skalabilitas, keterbaruan, dan keamanan yang baik. Tingkat *client* menangani semua interaksi *user* dengan

aplikasi. Aplikasi ini bertanggung jawab untuk semua *input user* dan berkomunikasi dengan lapisan bisnis. Tingkat bisnis akan memberlakukan aturan bisnis, memproses data, dan mengelola transaksi.

Arsitektur *Three Tier* ini memiliki tiga tingkat (*tier*) dimana tiap *tier* berpotensi untuk dijalankan dalam *platform berbeda*. Ketiga *tier* tersebut adalah:

1. *User interface tier* bertanggung jawab pada pengiriman masukan dari *user* dan menampilkan atau mencetak hasil kepada *user*.
2. *Business logic tier* bertanggung jawab melakukan pemrosesan kebutuhan untuk menyelesaikan *business problem*
3. *Database tier* bertanggung jawab kepada pengakses data yang disimpan dalam penyimpanan eksternal.



Gambar 2.11 Arsitektur *three tier*

Ada empat konfigurasi pokok yang menggunakan *model three tier*, yaitu

1. *Single computing system model*

Proses ketiga *tier* dari arsitektur *three tier* dieksekusi dalam satu mesin

2. *User interface distribution model*

Dalam *tier* pertama (*user interface tier*) dieksekusi dalam komputer *client* sedangkan proses *tier* kedua (*business logic*) dan *tier* ketiga (*database tier*) dieksekusi komputer *server*.

3. *Data access distribution model*

Proses *tier* pertama dan kedua dieksekusi dalam komputer *client* dan *tier* yang ketiga dieksekusi dalam *server*.

4. *Function distribution model*

Proses masing-masing *tier* dieksekusi oleh komputer yang berbeda. Proses *tier* pertama dieksekusi komputer *client*, sedangkan proses *tier* kedua dan ketiga dieksekusi masing-masing *Server* (*server business logic* dan *database*).

2.7 SQL SERVER

SQL Server adalah DBMS dari *Microsoft*. SQL Server dapat ditingkatkan dengan baik dan bisa digunakan untuk mengembangkan aplikasi untuk segala hal, mulai dari network kecil hingga ribuan user.

SQL Server memiliki beberapa keistimewaan unik yang tidak boleh diabaikan. Pertama-tama, SQL Server memiliki bahasa pemrograman tersendiri yang disebut Transact-SQL (T-SQL). T-SQL adalah ekstensi dari SQL dan kemampuannya begitu tinggi di mana bisa melakukan hampir semua yang dapat dilakukan dengan Visual Basic. T-SQL tidak memiliki antar muka user

tertapi mendukung banyak fungsi manipulasi data serta pernyataan flow-control. Di samping itu juga bisa mengakses tabel-tabel database melalui SQL. Pada dasarnya, T-SQL menggabungkan kekuatan SQL dengan struktur bahasa pemrograman lebih tradisional. Kelebihan dari T-SQL adalah dijalankan pada *server* dan dapat memanipulasi tabel secara local. Dengan cara memindahkan informasi dari *server* ke client dan memprosesnya di client.

2.7.1. Pengantar SQL

SQL singkatan dari *Structured Query Language*. Dalam bahasa Inggris sering dibaca sebagai SEQUEL. SQL merupakan bahasa query standar yang digunakan untuk mengakses basis data relasional. Standarisasi internasional terhadap SQL pertama kali dilakukan oleh ANSI (*American National Standards Institution*), melalui publikasi Database Language SQL (ANSI X3.136-1986). Saat ini, ANSI dan ISO (*International Standards Organization*) merupakan dua organisasi yang membuat standarisasi terhadap SQL.

2.7.1.1. Elemen SQL

Elemen dasar SQL mencakup pernyataan, nama, tipe data, konstanta, ekspresi, dan fungsi bawaan

2.7.1.2. Pernyataan

Pernyataan adalah perintah SQL yang meminta sesuatu tindakan kepada DBMS. SQL memiliki kira-kira 30 pernyataan. Beberapa pernyataan dasar SQL dapat dilihat pada tabel 2.1

Tabel 2.1 Pernyataan-pernyataan dasar SQL

Pernyataan	Keterangan
ALTER	Menambah struktur tabel

COMMIT	Mengakhiri sebuah eksekusi transaksi
CREATE	Menciptakan tabel, indeks, atau pandangan
DELETE	Menghapus baris pada tabel
DROP	Menghapus tabel, indeks, atau pandangan
GRANT	Menguasakan hak terhadap basis data kepada pengguna atau group pengguna
INSERT	Menambah baris pada tabel
REVOKE	Membatalkan hak terhadap basis data
ROLLBACK	Mengembalikan ke keadaan semula sekiranya suatu transaksi gagal dilaksanakan
SELECT	Memilih baris dan kolom pada tabel
UPDATE	Mengubah nilai pada sebuah baris

2.7.1.3. Nama

Nama digunakan sebagai identitas bagi objek-objek pada DBMS. Contoh

objek pada DBMS adalah tabel, kolom, dan pengguna. Setiap data memiliki tipe data. Beberapa tipe data standar dapat dilihat pada tabel 2.2. beberapa tipe data perluasan (spesifik terhadap DBMS tertentu) dapat dilihat pada tabel 2.3

Tabel 2.2 Tipe data standar

Tipe Data	Keterangan
CHAR	Untuk menyatakan deretan karakter (string). Misalnya, untuk menyatakan nama orang, nama jalan, atau nama kota
INTEGER	Untuk menyatakan bilangan bulat

NUMERIC	Untuk menyatakan bilangan real
---------	--------------------------------

Tabel 2.3 Tipe data perluasan

Tipe Data	Keterangan
VARCHAR	Untuk menyatakan string yang panjangnya bervariasi
MONEY	Untuk menyatakan uang
BOOLEAN	Menyatakan tipe logis (True atau False)
BLOB	Menyatakan data biner (gambar, suara, dan sebagainya)
SERIAL atau AUTOINCREMENT	Untuk menyatakan nilai urut

2.8 APOTEK

2.8.1. Pengertian Apotek

Yang dimaksud dengan apotek dalam Peraturan Pemerintah no. 26 tahun 1965 adalah suatu tempat tertentu dimana dilakukan usaha-usaha dalam bidang farmasi dan pekerjaan kefarmasian. Pengertian apotek ini diperbarui dalam Peraturan Pemerintah No.25 tahun 1980. Dalam peraturan pemerintah ini yang disebut apotek adalah suatu tempat tertentu, tempat dilakukan pekerjaan , kefarmasian adalah pembuatan termasuk pengendalian mutu sediaan farmasi , pengamanan peng gandaan, penyimpanan dan distribusi obat, pengelolaan obat pelayanan obat atas resep dokter, pelayanan informasi obat, serta pengembangan obat, bahan obat dan obat tradisional.

2.8.2. Tujuan dan Fungsi Apotek

Peraturan Pemerintah No. 25 tahun 1980 ini menjelaskan bahwa tujuan dan fungsi apotek antara lain :

- a. Tempat pengabdian profesi seorang apoteker yang telah mengucapkan jabatan.
- b. Sarana farmasi yang melakukan peracikan, pegubahan bentuk, pencampuran dan penyerahan obat atau bahan obat.
- c. Sarana penyaluran pembekalan farmasi yang harus menyebarkan obat yang diperlukan masyarakat secara meluas dan merata

BAB III

ANALISA DAN DESAIN SISTEM

Apotek merupakan salah satu pelayanan penjualan obat kepada masyarakat, dimana penyaluran obat terhadap masyarakat dan terjadi suatu transaksi penjualan. Apotek “K24 PHARMACY” Bekasi yang terletak di Jalan S.Parman No.36 Bekasi yang dikelola oleh Ibu Dra. Yuani Widjayanti memiliki kekuasaan penuh untuk mengolah apotek dan menentukan kebijaksanaan semua kegiatan apotek.

3.1 Analisa Sistem Berjalan

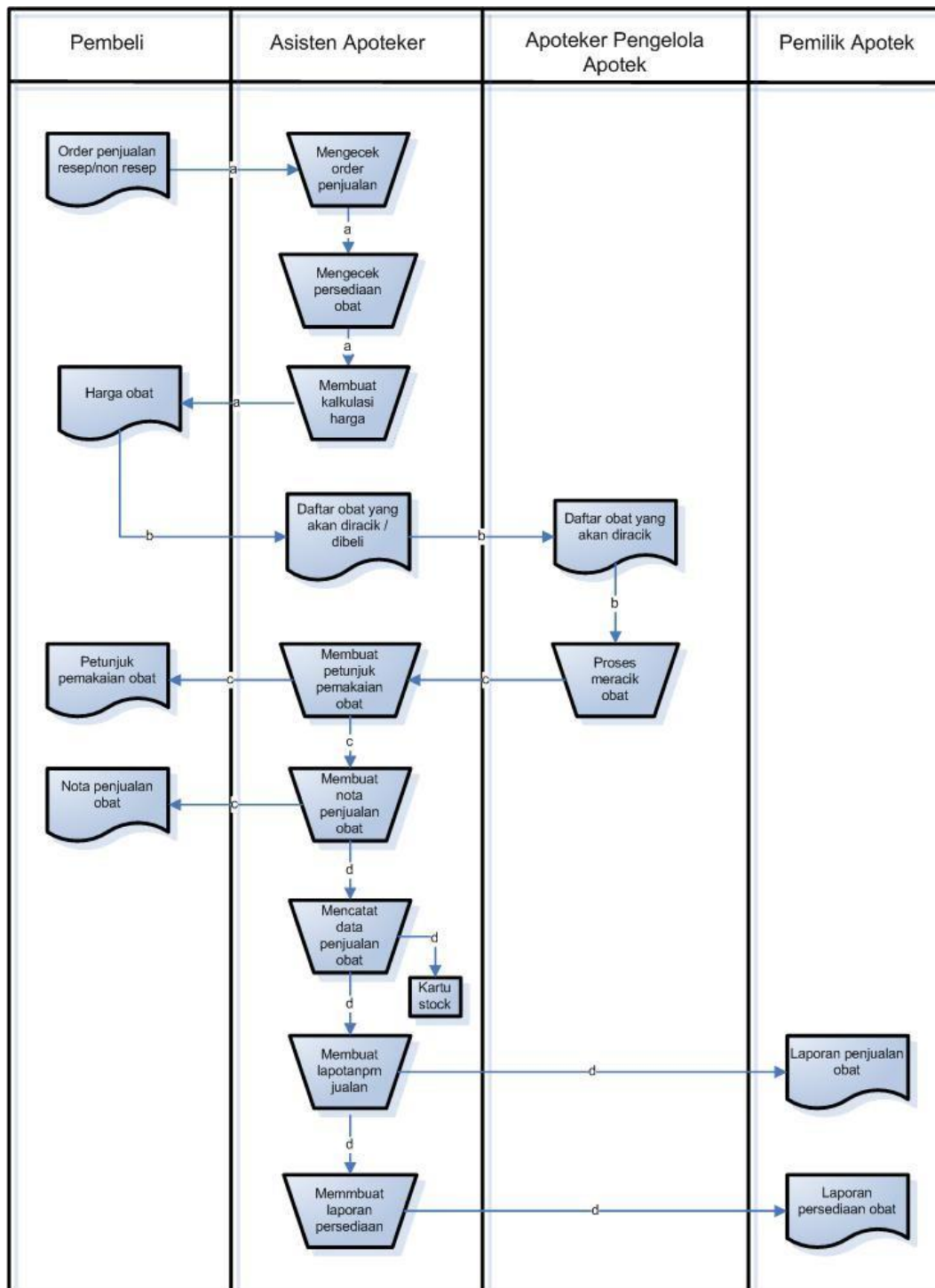
3.1.1 Sistem Penjualan Obat

Pelayanan kebutuhan obat terhadap masyarakat yang dilakukan oleh apotek “K24 PHARMACY” dibedakan menjadi dua yaitu : penjualan obat dengan resep dokter dan tanpa resep dokter. Setiap transaksi dengan resep dokter harus dicatat identitas siapa pasien dan dokter yang memberi resep dan obat apa yang diberikan. Untuk transaksi penjualan non resep tidak perlu dicatat identitasnya. Penjualan obat baik itu dengan resep dokter maupun tidak, petugas apotek akan mencatat nama obat dan jumlah obat yang terjual. Ini semua dilakukan untuk mengetahui nama dan jumlah obat yang terjual setiap harinya dan digunakan untuk pembuatan laporan mengenai penjualan obat dan untuk mengetahui stok obat yang masih ada untuk tiap harinya. Petugas apotek membuat laporan laporan

persediaan obat, membuat laporan-laporan penjualan obat. Pembuatan laporan tersebut dilakukan secara manual.

Semua kegiatan yang dilakukan secara manual tersebut sangatlah tidak efisien dan dimungkinkan juga adanya ketidak cermatan atau tidak telitinya seorang petugas apotek dalam pemasukan data-data yang diperlukan untuk suatu penyajian sebuah informasi. Informasi yang valid yaitu data yang akurat, tepat waktu dan relevan merupakan suatu tuntutan yang tidak bisa dielakkan lagi. Dalam hal ini penulis mencoba untuk membantu dengan mengubah sistem yang lama yaitu sistem yang manual menjadi sistem terkomputerisasi. Dengan adanya sistem terkomputerisasi ini diharapkan pekerjaan akan lebih mudah dan akan mengurangi adanya ketidak telitian dan ketidak cermatan petugas apotek dalam penyajian informasi dan semua pekerjaan itu akan dilakukan secara efisien pembuatan laporan-laporan akan dapat dilakukan dengan cepat.

3.1.2 Diagram Prosedur Sistem Berjalan



Gambar 3.1. Diagram Prosedur Penjualan Obat Sistem Lama

Keterangan diagram prosedur penjualan obat sistem lama pada Gambar 3.1. adalah sebagai berikut :

- a) Seorang pembeli memberikan data-datanya berupa surat resep dari dokter, atau permintaan obat non resep kepada asisten apoteker. Asisten apoteker kemudian akan mengecek persediaan obat. Jika memenuhi asisten apoteker akan membuat kalkulasi harga obat dan memberikan daftar harga obat kepada pembeli untuk mendapatkan persetujuan pembeli.
- b) Dengan menyetujui harga obat, pembeli membayar dan kemudian asisten apoteker membuat daftar obat yang akan diracik atau resep dari dokter dan diserahkan kepada APA (Apoteker Pengelola Apotek) untuk proses meracik obat.
- c) Selesai meracik obat, diserahkan kepada asisten apoteker yang kemudian membuat petunjuk pemakaian obat dan membuat nota penjualan obat dan diserahkan kepada pembeli, berupa petunjuk pemakaian obat, obat dan nota penjualan obat
- d) Asisten akan mencatat data penjualan obat dengan membuat catatan di kartu stock, serta membuat laporan penjualan dan laporan persediaan obat dan nantinya diserahkan pada pemilik apotek pada periode tertentu, berupa laporan persediaan obat dan laporan penjualan obat.

3.2 Analisa Sistem Usulan

3.1.3 Sistem Penjualan Obat

Pada sistem yang baru ini akan membuat sistem lama yang dikerjakan secara manual akan menjadi sistem yang terkomputerisasi.

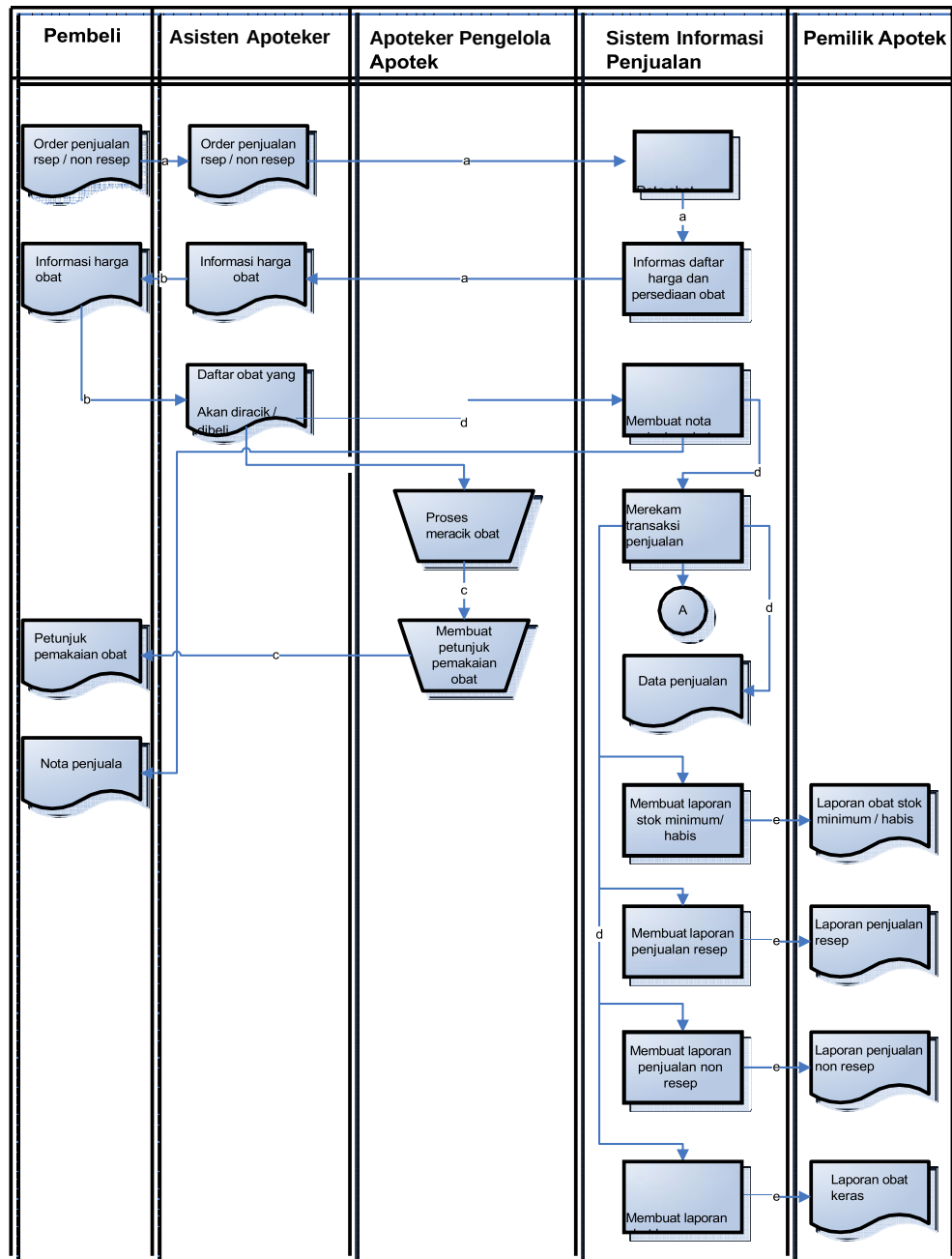
Pelayanan kebutuhan obat terhadap masyarakat yang dilakukan oleh apotek “K24 PHARMACY” dibedakan menjadi dua yaitu, penjualan obat dengan resep dokter dan tanpa resep dokter. Setiap transaksi penjualan dengan resep dokter, obat yang menggunakan resep dokter itu adalah obat-obatan golongan keras dan golongan psikotropika. Setiap transaksi ini petugas apotek akan memasukkan data dikomputer, antara lain identitas dokter, identitas pasien dan obat-obatan yang diresepkan oleh dokter. Penjualan obat non resep, digunakan untuk melayani penjualan obat-obatan golongan bebas dan golongan terbatas. Pada penjualan obat non resep ini hanya perlu input kode obat ataupun nama obat dan jumlah obat yang dibeli oleh konsumen tanpa harus mencatat identitas dari konsumen.

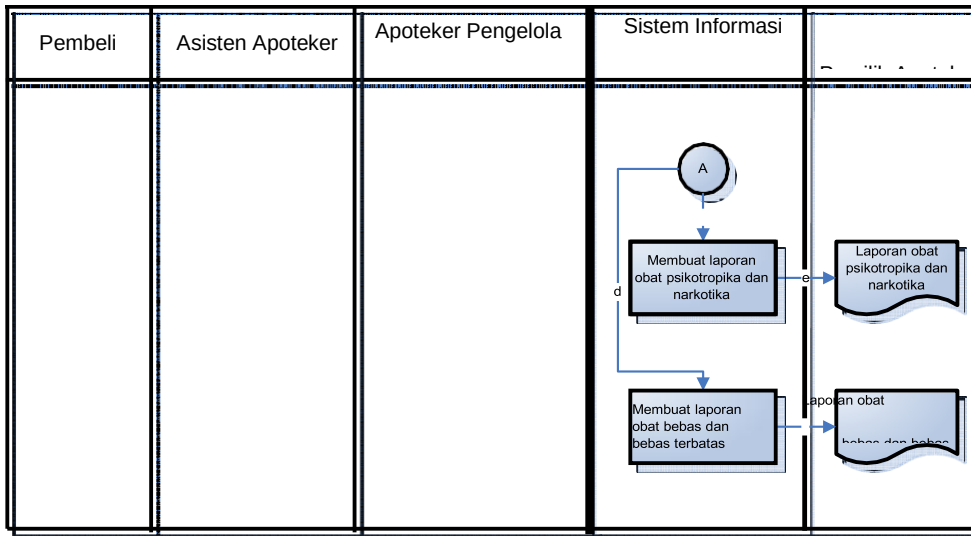
Nota transaksi penjualan obat dengan resep dokter dan non resep akan dibedakan. Transaksi penjualan obat dengan resep dokter nomor nota akan ditambahkan dengan huruf “NR” sedangkan nota transaksi penjualan non resep nomor nota akan ditambahkan dengan huruf “TNR”.

Data persediaan/stok obat yang ada dalam apotek akan terekam dalam *database* obat. Data persediaan obat bisa ditampilkan setiap saat sehingga petugas apotek akan mengetahui jumlah persediaan stok obat. Setiap kali ada transaksi

penjualan obat maka data stok obat yang ada akan langsung berkurang dengan sendirinya.

3.1.4 Diagram Prosedur Sistem Usulan





Gambar 3.2. Diagram Prosedur Penjualan Obat Sistem Baru

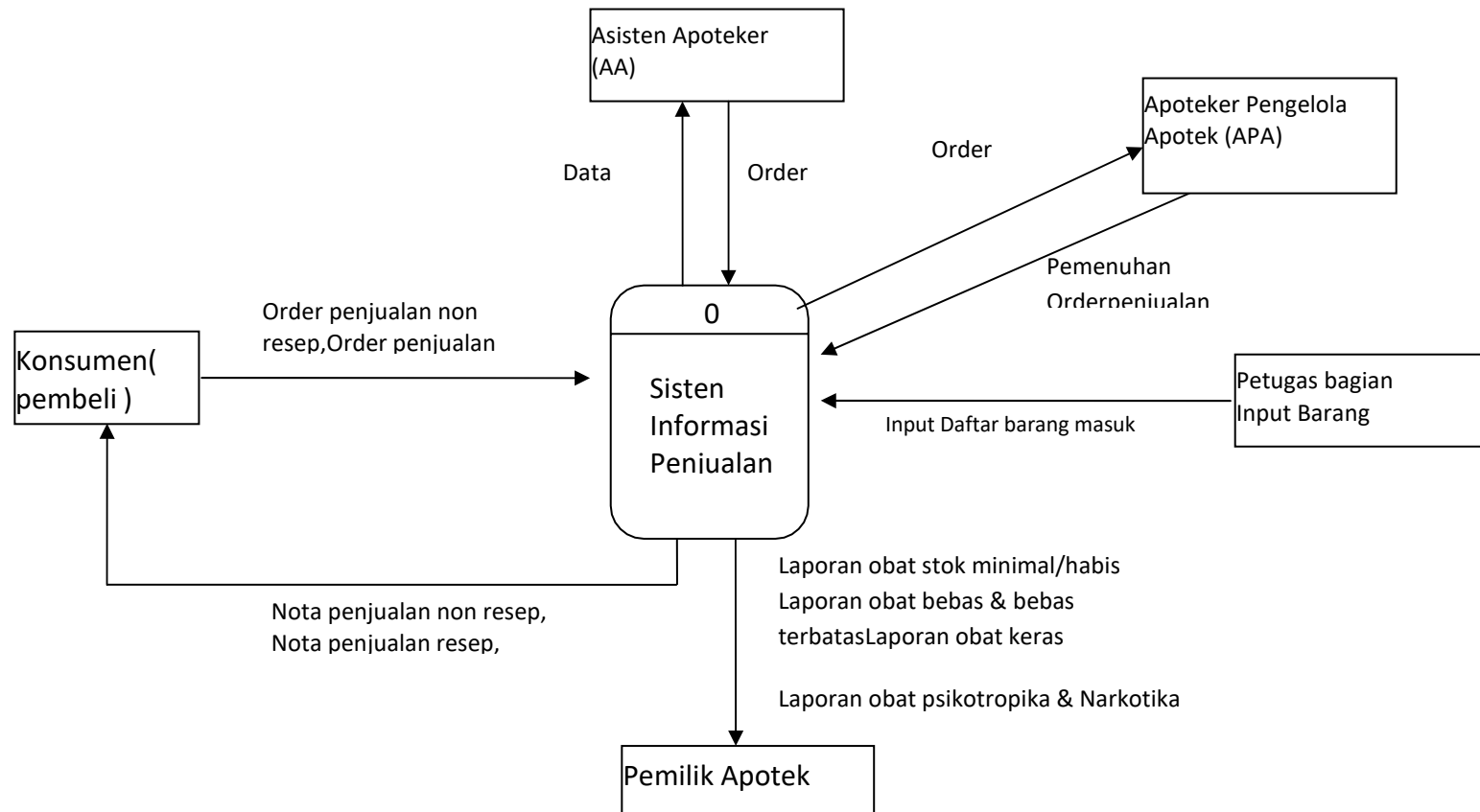
Keterangan diagram prosedur penjualan obat sistem lama pada Gambar 3.2. adalah sebagai berikut :

- Seorang pembeli memberikan data-datanya berupa surat resep dari dokter, atau permintaan obat non resep kepada asisten apoteker. Asisten apoteker kemudian akan mengecek persediaan obat di komputer. Jika persediaan obat ada maka informasi harga obat disampaikan ke pembeli.
- Pembeli menerima informasi harga obat, jika setuju dengan harganya, maka pembeli membayar dan Apoteker Pengelola Apotek akan mulai meracik obat dan menyiapkan obat yang dibeli.
- Selesai meracik obat, kemudian membuat petunjuk pemakaian obat dan pembeli menerima petunjuk pemakaian obat dan obat.
- Komputer membuat nota penjualan obat, kemudian pembeli menerima nota penjualan obat. Komputer merekam nota penjualan obat, yang nantinya digunakan untuk membuat laporan transaksi penjualan resep,

membuat laporan penjualan non resep, membuat laporan obat keras, membuat laporan obat psikotropika dan narkotika, membuat laporan obat bebas dan bebas terbatas dan membuat laporan stok obat minimum/habis .

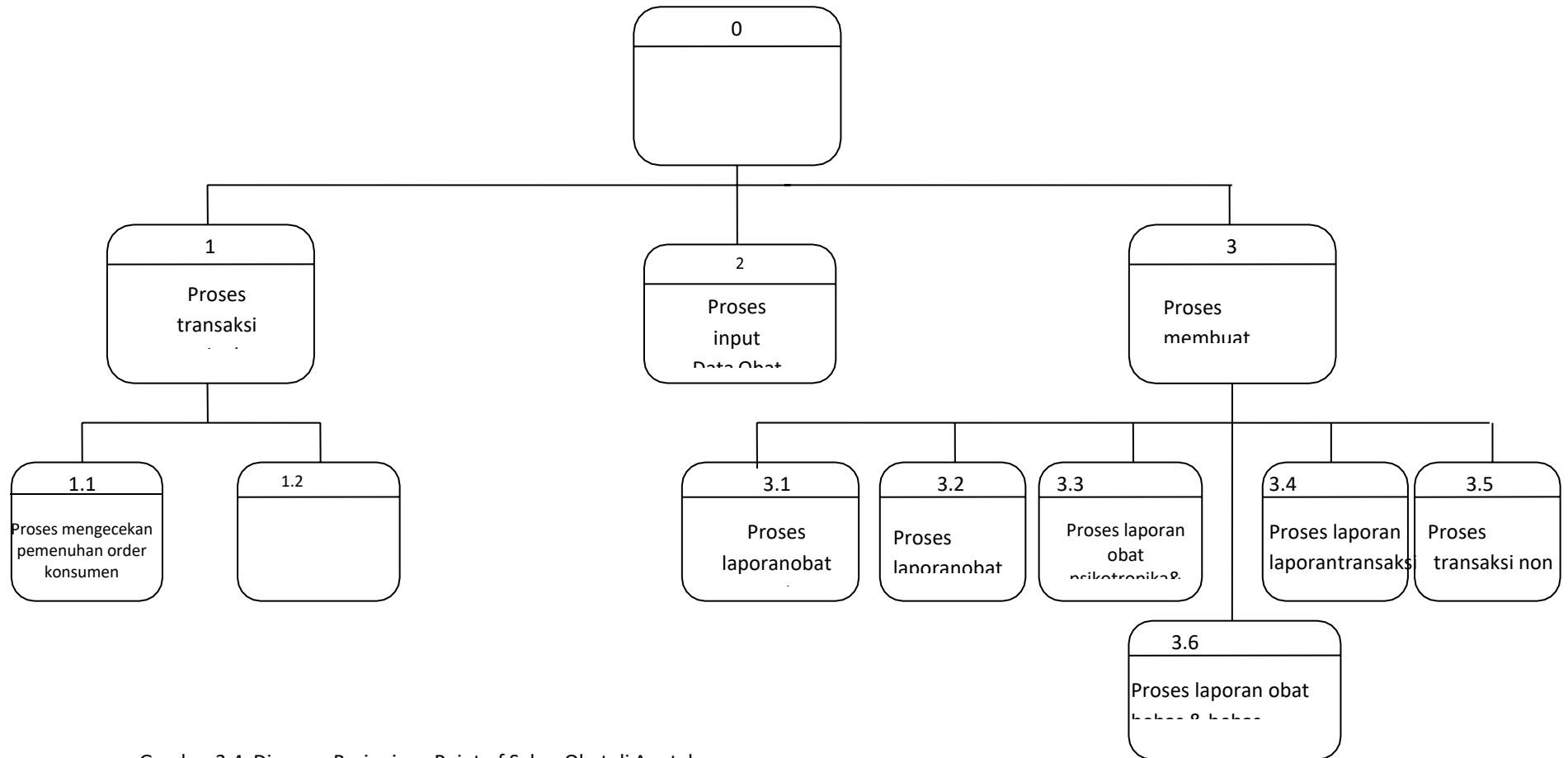
- e) Pemilik apotek menerima laporan penjualan obat berupa laporan transaksi penjualan resep, laporan penjualan non resep, laporan obat keras, laporan obat psikotropika dan narkotika, laporan obat bebas dan bebas terbatas dan laporan stok obat minimum/habis .

3.1.5 Diagram Konteks (Context Diagram)



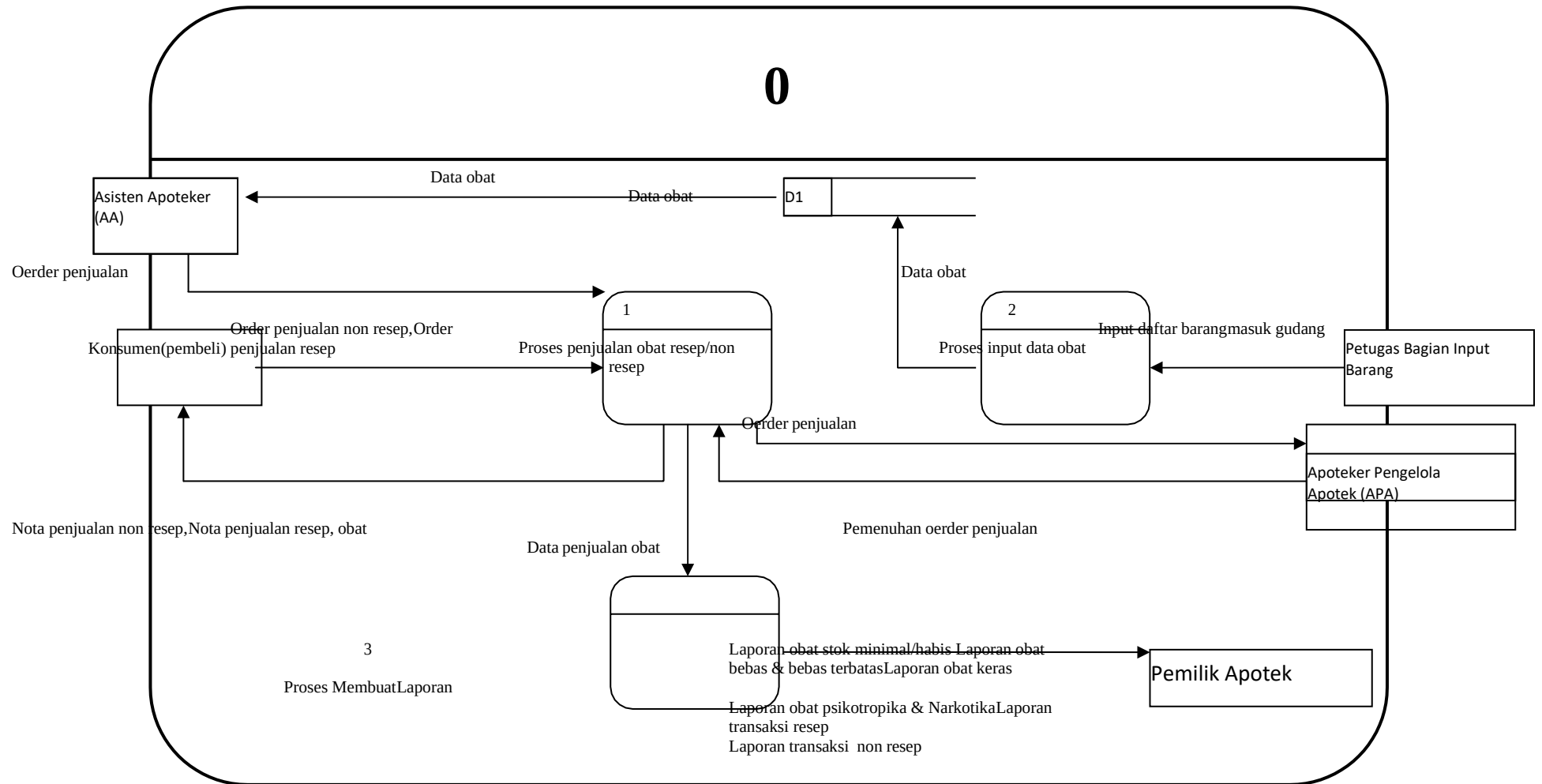
Gambar 3.3. Diagram Konteks Point of Sales Apotek

3.1.6 Diagram berjenjang (hierarchy Chart)



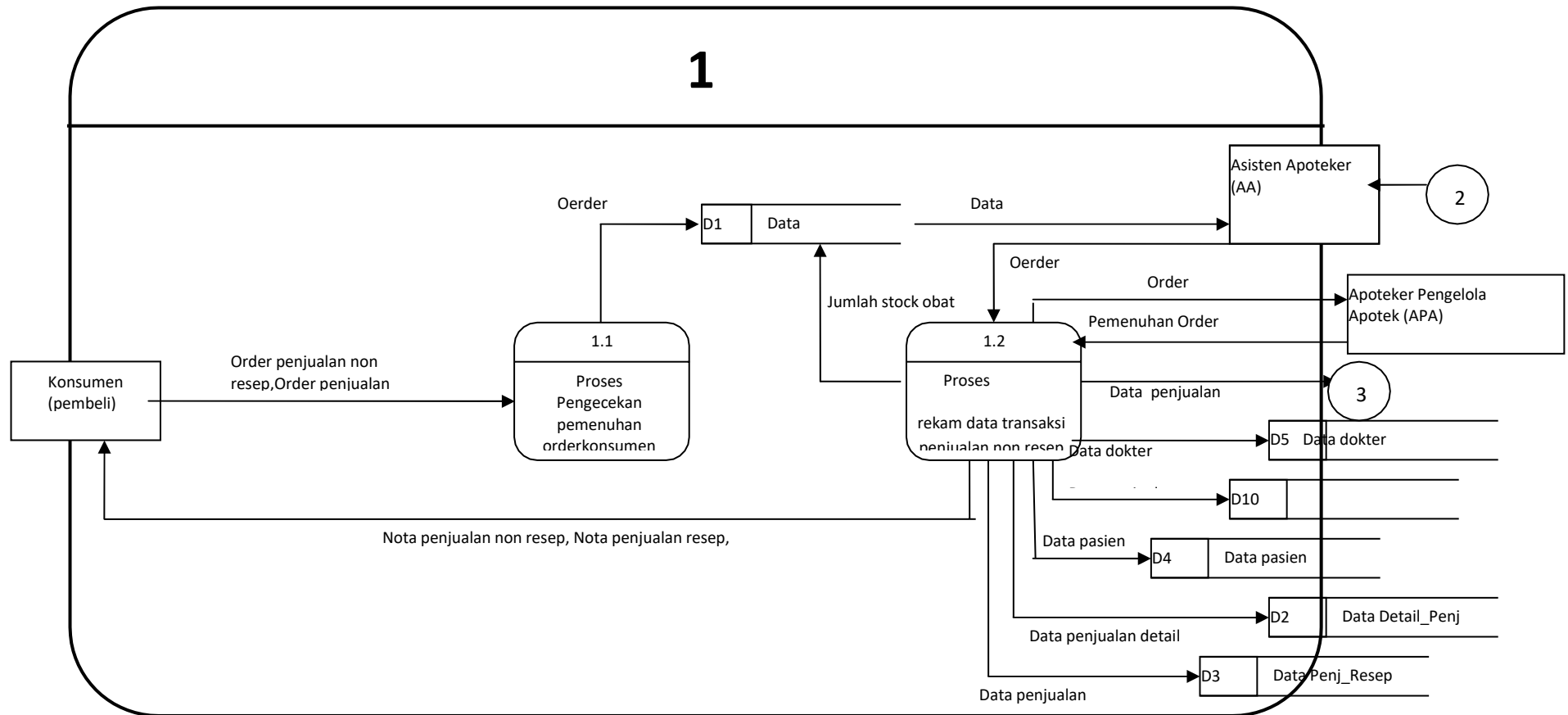
Gambar 3.4. Diagram Berjenjang Point of Sales Obat di Apotek

3.1.7 Diagram Arus Data Level 0



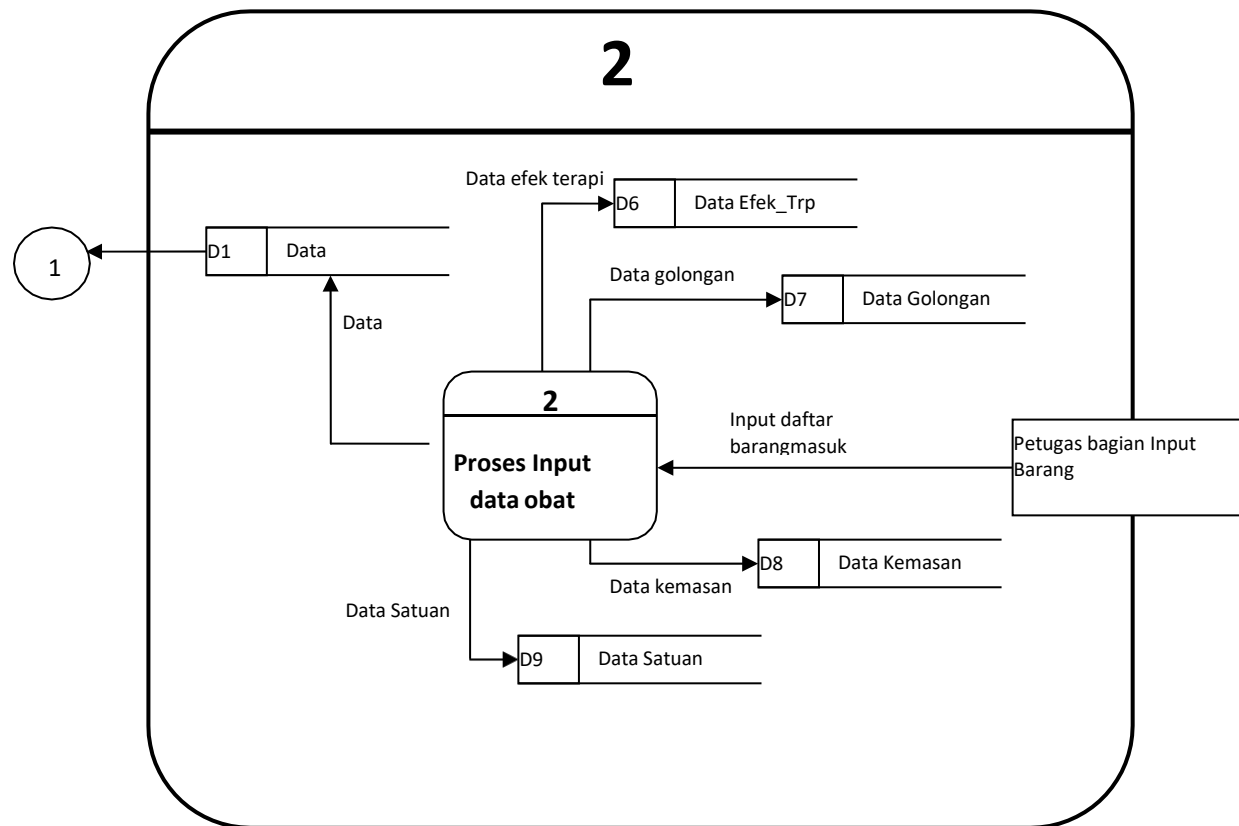
Gambar 3.5. Diagram Arus Data Level 0 Point of Sales Apotek

3.1.8 Diagram Arus Data Level 1 Proses 1 Penjualan Obat



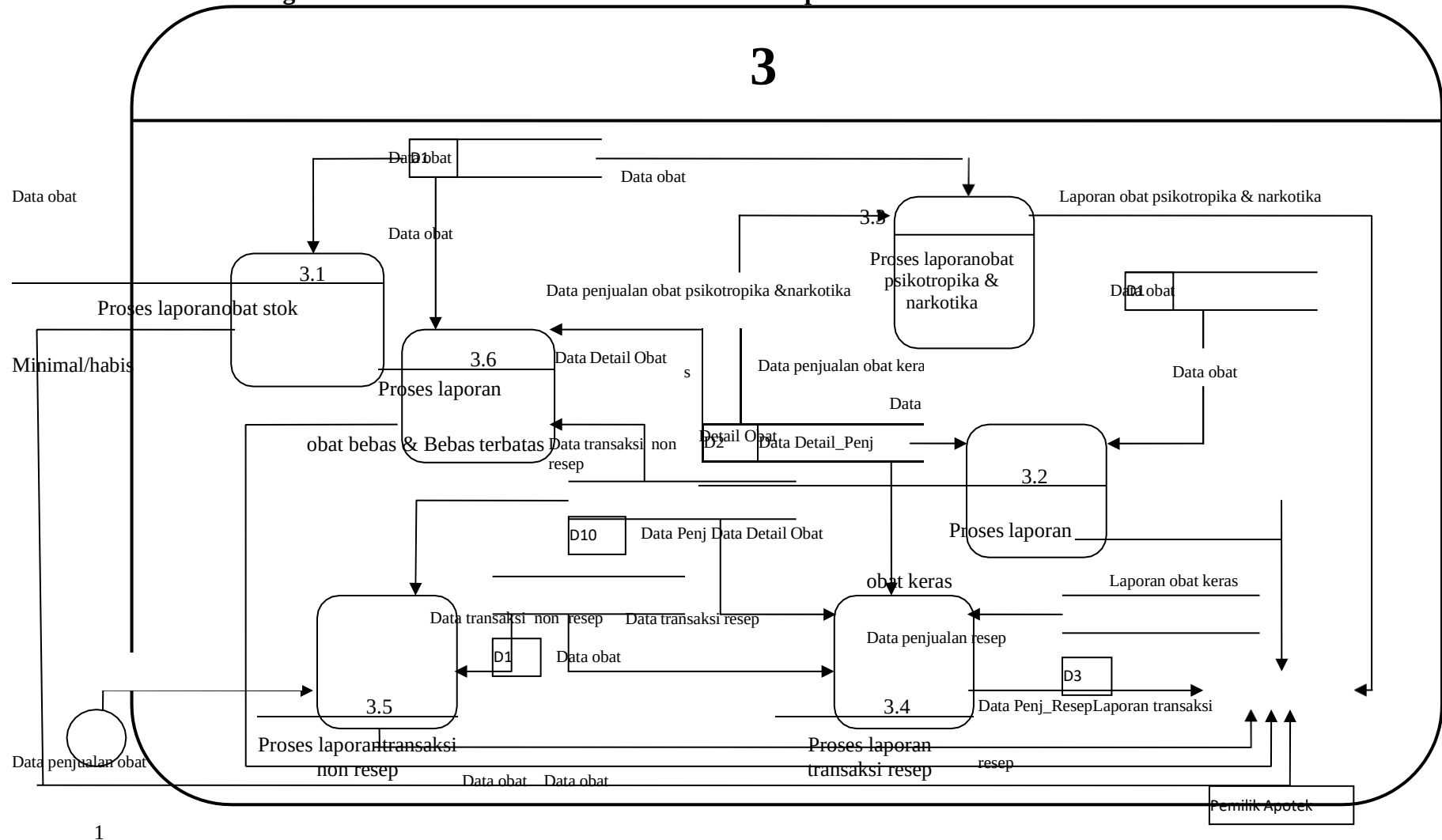
Gambar 3.6. Diagram Arus Data Level 1 Proses 1 Penjualan Obat

3.1.9 Diagram Arus Data Level 1 Proses 2 Rekam Data Stock



Gambar 3.7. Diagram Arus Data Level 1 Proses 2 Rekam Data Stock

3.1.10 Diagram Arus Data Level 1 Proses 3 Pembuatan Laporan



Laporan obat bebas dan bebas terbatas
 Laporan obat stok minimal / habis

Gambar 3.8. Diagram Arus Data Level 1 Proses 3 Pembuatan laporan

manajemen *database* yang dipakai. Tabel-tabel yang direncanakan antara lain sebagai berikut :

a. Tabel Obat

Nama	Type	Width	Keterangan
kode_obat (*)	varchar	15	Kode obat
Id_user	varchar	6	Id user
kode_efek_t(**)	varchar	6	Kode efek terapi
kode_gol(**)	varchar	6	Kode golongan
kode_kemasan(**)	varchar	6	Kode kemasan
kode_satuan(**)	varchar	6	Kode satuan
nama_obat	varchar	50	Nama obat
satatus	varchar	15	Status obat resep/non resep
stk_brg	varchar	4	Stock barang
harga_beli	money		Harga beli
harga_jual	money		Harga jual

Tabel 3.1. Struktur Tabel Obat

b. Tabel Detail_Penj

Nama	Type	Width	Keterangan
kode_obat(**)	varchar	6	Kode obat
no_nota(**)	varchar	15	Nomor nota
jumlah_brg	numeric	10	Jumlah barang yang dibeli
total	Money		Jumlah barang yang dibeli kali harga per item barang

Tabel 3.2. Struktur Tabel Detail_Penj

c. Table Penj

Nama	Type	Width	Keterangan
no_nota(*)	varchar	15	Nomor nota
Id_user(*)	varchar	6	Kode user
tgl_trans	smalldatetime		Tanggal transaksi

Tabel 3.3. Struktur Tabel Penj

GrandTotal	money		
------------	-------	--	--

Tabel 3.3. Struktur Tabel Penj

d. Table Penj_Resep

Nama	Type	Width	Keterangan
no_nota(*)	varchar	15	Nomor nota
kode_resep	varchar	15	Kode Resep
kode_pas(**)	varchar	15	Kode Pasien
kode_doc(**)	varchar	15	Kode Dokter

Tabel 3.4. Struktur Tabel Penj_Resep

e. Tabel Efek_Trp

Nama	Type	Width	Keterangan
kode_efek_t(*)	nchar	6	Kode efek terapi
efek_terapi	varchar	50	Nama efek terapi

Tabel 3.5. Struktur Tabel Efek_Trp

f. Tabel Golongan

Nama	Type	Width	Keterangan
kode_gol(*)	nchar	6	Kode golongan
nama_gol	varchar	50	Nama golongan

Tabel 3.6. Struktur Tabel Golongan

g. Tabel Kemasan

Nama	Type	Width	Keterangan
kode_kemasan(*)	nchar	6	Kode kemasan
Nama_kemasan	varchar	50	Nama kemasan

Tabel 3.7. Struktur Tabel Kemasan

h. Tabel Satuan

Nama	Type	Width	Keterangan
------	------	-------	------------

Tabel 3.8. Struktur Tabel Satuan

kode_satuan(*)	nchar	6	Kode satuan
nama_satuan	varchar	15	Nama satuan

Tabel 3.8. Struktur Tabel Satuan

i. Tabel Dokter

Nama	Type	Width	Keterangan
kode_doc(*)	varchar	15	Kode dokter
nama_doc	varchar	50	Nama dokter
almt_doc	varchar	50	Alamat dokter
telepon_doc	varchar	15	Telepon dokter

Tabel 3.9. Struktur Tabel Dokter

j. Tabel Pasien

Nama	Type	Width	Keterangan
kode_pas(*)	varchar	15	Kode pasien
nama_pas	varchar	50	Nama pasien
umur	varchar	3	Umur pasien
jenis_kelamin	varchar	1	Jenis Kelamin
almt_pas	varchar	50	Alamat pasien
telepon_pas	varchar	15	Telepon pasien

Tabel 3.10. Struktur Tabel Pasien

k. Tabel Login

Nama	Type	Width	Keterangan
Id_user(*)	varchar	6	Kode user
Nama_lkp	varchar	50	Nama lengkap
nama_user	varchar	15	Nama user
status	varchar	10	Group
password	varchar	10	Password

Tabel 3.11. Struktur Tabel Login

Desain Interface

3.1.11 Login KeModul Apotek

Memasukkan Id User, Nama User, Status dan Password pada tempat yang tersedia.

The diagram shows a login window with a title bar 'LOGIN'. Inside, there is a container with four input fields. Labels 'Id User', 'Nama User', 'Status', and 'Password' are positioned to the left of the first, second, third, and fourth fields respectively. At the bottom, there are two buttons labeled 'LOGIN' and 'KELUAR'.

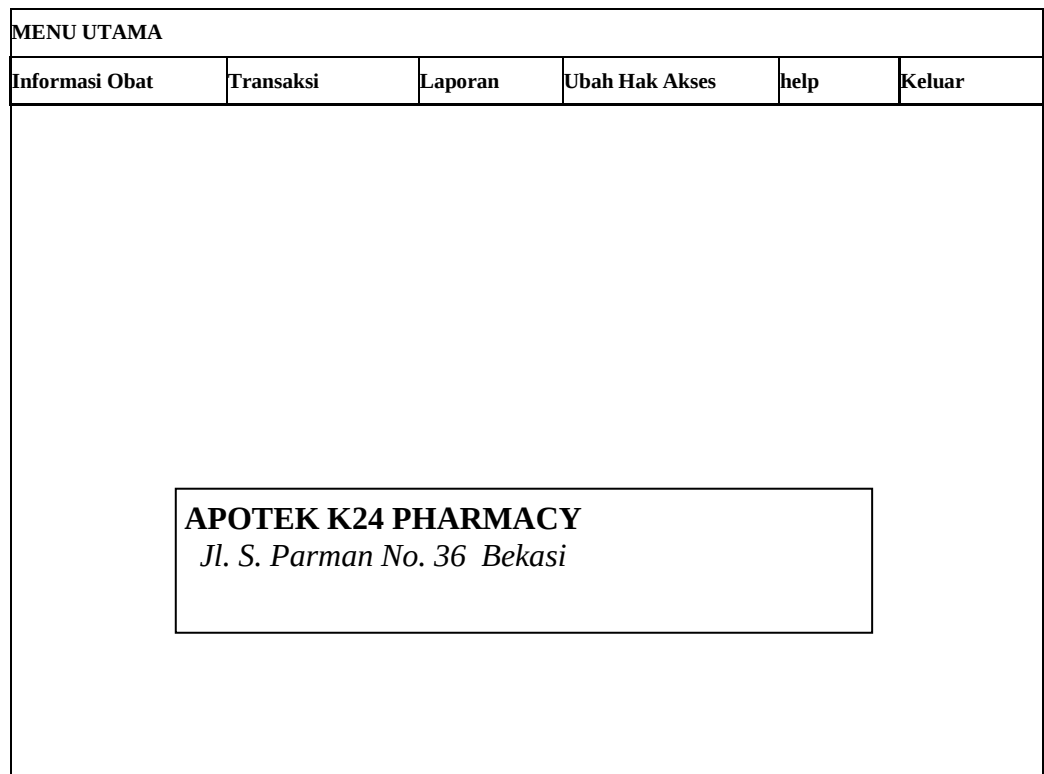
Gambar 3.11. Desain Interface Login User

Jika Is User, Nama, Status User dan Password yang dimasukkan telah terdaftar, maka akan muncul tampilan Menu Utama.atau Lay ar Utama

3.1.12 Layar Utama

Berikut ini adalah desain layar utama Point of Sales apotek.

Layar utama ini berisi menu-menu yang ada dalam Point of Sales obat di apotek.

Gambar 3.12. Desain *Inteface* Layar Utama

3.1.13 Menu Bar

a. Bar Menu Utama

Informasi Obat	Transaksi	Laporan	Ubah Hak Akses	Help	Keluar
----------------	-----------	---------	----------------	------	--------

Gambar 3.13. Desain Bar Menu Utama

b. Bar Menu Informasi Obat

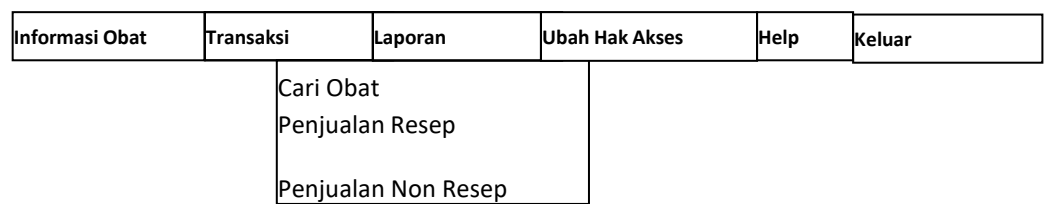
Informasi Obat	Transaksi	Laporan	Ubah Hak Akses	Help	Keluar
Data Obat Detail Obat >					

Gambar 3.14. Desain Bar Menu Informasi Obat



Gambar 3.15. Desain Bar Menu Informasi Obat Sub Menu Detail Obat

c. Bar Menu Transaksi



Gambar 3.16. Desain Bar Menu Transaksi

d. Bar Menu Laporan, Ubah Hak Akses, Help dan Keluar



Gambar 3.17. Desain Bar Menu Laporan, Ubah Hak Akses, Help dan Keluar

Desain Input

3.1.14 Desain Detail Obat

DATA OBAT								
	Kode Obat	Nama Obat	Golongan	Efek terapi	Stok	Harga Beli	Harga Jual	Status
>								

Gambar 3.18. Desain Input Master Obat

3.8.1.1 Desain Input DATA BARU

Data Baru berfungsi untuk memasukkan data seluruh obat-obatan yang tersedia di apotek. Setiap memasukkan data obat, dilengkapi dengan mencantumkan jumlah stok harga jual, harga beli, efek terapi, golongan, kemasan serta satuan obatnya. Data obat yang telah dimasukkan, akan digunakan dalam proses transaksi penjualan obat.

TAMBAH DATA OBAT			
Kode obat			
Mama Obat			
Kode>Nama User			
Kode/Efek terapi			
Kode/Kemasan			
Kode/Satuan			
Kode/Golongan			
Stok Obat			
Harga Beli			
SIMPAN			
Harga Jual			
KELUAR			
Status			

Gambar 3.19. Desain Input Tambah Data Obat

3.8.1.2 Desain EDIT Data Obat

Edit data berfungsi untuk meng-Edit data obat dan nantinya form ini jugadigunakan untuk penambahan data stok obat.

EDIT DATA OBAT			
Kode obat			
Mana Obat			
Kode>Nama User			
Kode/Efek terapi			
Kode/Kemasan			
Kode/Satuan			
Kode/Golongan			
Stok Obat			
Harga Beli			
Harga Jual			
Status	EDIT KELUAR ▼		

Gambar 3.20. Desain Edit Data Obat

3.8.1.3 Desain HAPUS Data

Hapus data berfungsi untuk menghapus data pilih data yang diinginkan kemudian klik hapus maka akan tampil informasi sebagai berikut.

Perhatian	
Apakah yakin anda akan menghapus data ini?	
Yes	No

Gambar 3.21. Desain informasi Hapus Data Obat

3.8.1.4 Desain OBAT TERLARIS

Obat terlaris berfungsi untuk mengetahui obat yang paling laris terjual.

RANGKING PENJUALAN OBAT			
Keluar			
	Kode Obat	Nama Obat	Total Penjualan
>			

Gambar 3.22. Desain Ranging Penjualan Obat

3.8.1.5 Desain STOK OBAT MINIMUM

Stok obat minimum berfungsi untuk mengetahui jumlah stok obat yang habis 0 atau sisa 1. Hal ini digunakan untuk melakukan pengadaan obat.

DATA OBAT MINIMUM			
Keluar			
	Kode Obat	Nama Obat	Stok barang
>			

Gambar 3.23. Desain Stok Obat Minimum

3.1.15 Desain Input DETAIL OBAT

3.8.2.1. Input Detail Obat Sub Menu Efek Terapi

Input Efek Terapi berfungsi untuk memasukkan data efek terapi obat. Data efek terapi ini akan digunakan pada saat akan memasukkan data obat. Contoh data efek terapi

- Anti biotik
- Anti histamin

DATA EFEK TERAPI		
Kode Efek Terapi	Nama	
Efek Terapi	Kd Efek Trp	Nama Efek Terapi
>		
SIMPAN KELUAR		

Gambar 3.24. Setup Detail Obat Sub Menu Efek Terapi

3.8.2.2. Input Detail Obat Sub Menu Golongan Obat

Input Golongan Obat berfungsi untuk memasukkan data golongan obat. Data golongan obat ini akan digunakan pada saat akan memasukkan data obat. Contoh data golongan obat

:

- Narkotika
- Bebas
- Keras

DATA GOLONGAN OBAT

Kode Golongan

Kode Golongan

Nama Golongan

SIMPAN

KELUAR

Gambar 3.25. Setup Detail Obat Sub Menu Golongan

3.8.2.3. Input Detail Obat Sub Menu Kemasan Obat

Input Kemasan Obat berfungsi untuk memasukkan data kemasan obat. Data kemasan obat ini akan digunakan pada saat akan memasukkan data obat. Contoh data kemasan obat :

- Tablet - Kapsul
- Salep

DATA KEMASAN OBAT

Kode KemasanNama

	Kode Kemasan	Nama Kemasan
Kemasan >		

SIMPAN

KELUAR

Gambar 3.26. Setup Detail Obat Sub Menu Kemasan

3.8.2.4. Input Detail Obat Sub Menu Satuan Obat

Input Satuan Obat berfungsi untuk memasukkan data satuan obat. Data satuan obat ini akan digunakan pada saat akan memasukkan data obat. Contoh data satuan obat :

- Gram
- Miligram
- Cc

DATA SATUAN OBAT		
Kode Satuan		
Nama		
Satuan		
SIMPAN KELUAR		

Gambar 3.27. Setup Detail Obat Sub Menu Satuan

3.1.16 Desain Input Penjualan Resep

Desain input penjualan resep berfungsi untuk melakukan transaksi resep yang diberikan dokter. Transaksi resep akan menyimpan data dokter dan pasien yang ada dalam resep.

Tanggal /No. Notald

/ Nama User No

Resep

Kode/Nama Dokter

INPUT DATA
DOKTER

INPUT
DATA
PASIE

KodeObat

Nama Obat

Harga Satuan/Stok

HAPUS DARI
DAFTAR

MASUKKAN
DAFTAR

	Kode	Nama Obat	Harga Satuan	jumlah	Total
>					

Grand Total

Bayat

CETAK

KELUAR

Gambar 3.28. Desain Input Penjualan Resep

3.8.3.1 Input Data Dokter

Tombol INPUT DATA DOKTER pada Gambar 3.28 akan menampilkan *form* DATA DOKTER seperti pada Gambar 3.29 berikut. *Form* ini berfungsi untuk tambah, edit, hapus data dokter.

DATA DOKTER				
	Kode Dokter	Nama Dokter	Alamat	No. Telepon
>				

Gambar 3.29. Desain Data Dokter

3.8.3.1.1 Tombol Data Baru

Tombol Data Baru berfungsi untuk menambah data baru dokter. Desain input data dokter tampak seperti pada Gambar 3.30.

DATA DOKTER	
KodeDokter Nama	
Dokter Alamat	
DokterTelp.	
SIMPAN KELUAR	

Gambar 3.30. Desain Input Data Dokter

3.8.3.1.2 Tombol Edit

Tombol Edit berfungsi untuk meng-Edit data dokter. Desain Edit datadokter seperti tampak pada Gambar 3.31.

The form is titled "EDIT DATA DOKTER". It contains four input fields for editing doctor data: "KodeDokter Nama", "Dokter Alamat", "DokterTelp.", and an unlabeled field. At the bottom, there are two buttons: "EDIT" and "KELUAR".

Gambar 3.31. Desain Edit Data Dokter

3.8.3.1.3 Hapus Data Dokter

Hapus berfungsi untuk menghapus data Dokter, pilih data yang diinginkan kemudian klik hapus maka akan tampil informasi sebagai berikut.

The dialog box is titled "Perhatian". It contains the text "Apakah yakin anda akan menghapus data dokter ini ?" and two buttons: "Yes" and "No".

Gambar 3.32. Desain Hapus Data Obat

3.1.17 Desain Input Penjualan Non Resep

Desain input penjualan non resep berfungsi untuk melakukan transaksi non resep.

NOTA NON RESEP																													
Tanggal																													
No. Nota																													
KodeObat																													
Nama Obat					MASUKKAN DAFTAR																								
Harga Jumlah																													
Total					HAPUS DARI DAFTAR																								
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 5%;"></th> <th style="width: 15%;">Kode</th> <th style="width: 30%;">Nama Obat</th> <th style="width: 15%;">Harga Satuan</th> <th style="width: 15%;">jumlah</th> <th style="width: 20%;">Total</th> </tr> </thead> <tbody> <tr> <td>></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>							Kode	Nama Obat	Harga Satuan	jumlah	Total	>																	
	Kode	Nama Obat	Harga Satuan	jumlah	Total																								
>																													
<table style="width: 100%;"> <tr> <td style="width: 20%;">Grand Total</td> <td style="width: 20%;"> </td> <td style="width: 20%;"></td> <td style="width: 40%;"></td> </tr> <tr> <td>Bayat</td> <td> </td> <td></td> <td></td> </tr> <tr> <td></td> <td> </td> <td style="text-align: center; vertical-align: middle;">CETAK</td> <td style="text-align: center; vertical-align: middle;">KELUAR</td> </tr> </table>						Grand Total				Bayat						CETAK	KELUAR												
Grand Total																													
Bayat																													
		CETAK	KELUAR																										

Gambar 3.33. Desain Input Penjualan Non Resep

3.1.18 Desain Mencetak Laporan

Desain menu cetak sebagai berikut. Menu cetak berfungsi untuk mencetak laporan-laporan

MENU CETAK	
TRANSAKSI	OBAT KERAS
TRANSAKSI RESEP	OBAT PSIKOTROPIKA & NARKOTIKA
TRANSAKSI NON RESEP	OBAT BEBAS & BEBAS TERBATAS
	OBAT STOK MINIMUM / HABIS
Mulai Sampai ▼	
Nama Operator	
CETAK	KELUAR

Gambar 3.34. Desain Interface Cetak Laporan

3.9.1.5. Laporan Transaksi Penjualan Obat Biasa

DATA OBAT BIASA YANG TERJUAL

Mulai Tanggal : Sampai :

Dicetak pada :

Kode Obat	Nama Obat	Terjual	Jenis Obat	Stock

Jumlah nota

Gambar 3.39 Format Laporan Transaksi Penjualan Obat Biasa

3.9.1.6. Laporan Data Obat Habis/Minimum

DATA OBAT MINIMUM HABIS

Dicetak oleh : Dicetak pada :

Kode Obat	Nama Obat	Stock

Gambar 3.40 Format Laporan Data Obat Habis / Minimum

3.9.2. Nota Resep

APOTEK K24

Tanggal : _____

No. Nota : _____

No. Resep : _____

Nama Pasien : _____

Kode Obat	Nama Obat	Jumlah	Total

Bayar

GrandTotal

Kembali

Gambar 3.41. Nota Resep

3.9.3. Nota Non Resep

APOTEK K24

Tanggal :

No. Nota :

Kode Obat	Nama Obat	Jumlah	Total

Bayar
GrandTotal
Kembali

Gambar 3.42. Nota Non Resep

PERANCANGAN MULTI USER DALAM ARSITEKTUR *CLIENT-SERVER*

Pendekatan arsitektural yang mungkin dijadikan dasar pengembangan Point of Sales pada apotek berdasarkan pada sub bab 2.6 tentang aplikasi multi *user* adalah model *Client-Server*, hal ini dapat melihat beberapa pertimbangan sebagai berikut :

- a) Cakupan sistem yang dikembangkan sempit tidak terlalu besar dan bersifat lokal.
- b) Jumlah *client* yang mungkin mengakses sistem cenderung sama sehingga tidak begitu mengalami perubahan, dikarenakan proses yang ditangani hanyalah barang keluar saja.
- c) Sumber data yang relatif tidak besar sehingga cukup menggunakan satu server untuk pengolahan data.

3.10.1. Desain Aplikasi *Client-Server*

Berdasarkan desain sistem dengan DAD dan desain *database* dengan E-Rdiagram maka sistem akan bekerja pada dua lapisan sebagai berikut :

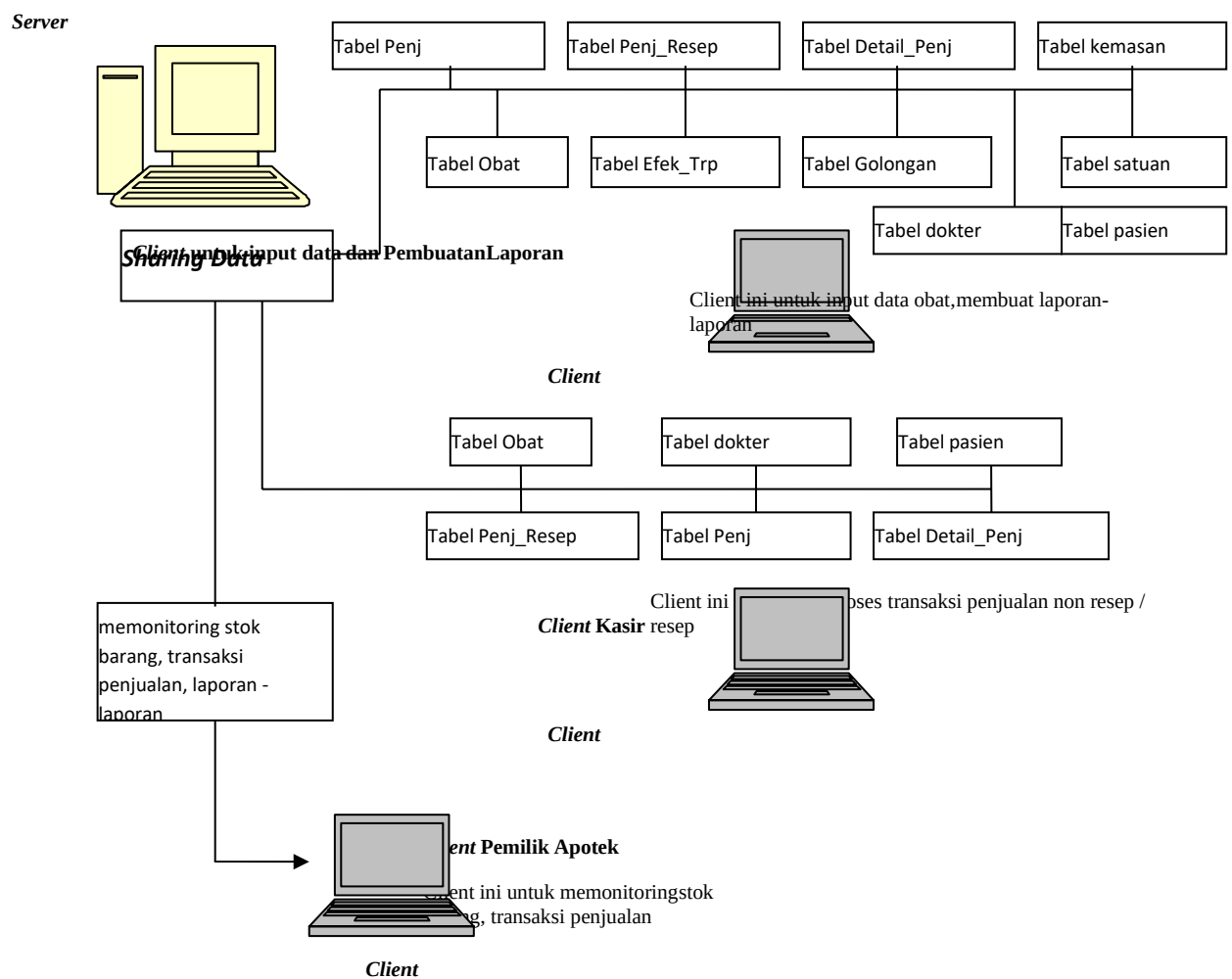
A. Lapisan *Server*

Desain struktur *database* dalam server yaitu :

- Beberapa file obat yang meliputi file efek_terapi, file golongan, file kemasan, dan file satuan.
- Beberapa file transaksi penjualan yang meliputi file Obat, file Penj_Resep, file Penj, file Detail_Penj, file Pasien, dan file Dokter.

B. Lapisan Client

Client adalah tempat layanan antara aplikasi dan *user*, untuk menunjang proses tersebut *client* akan terlibat proses input data dan proses transaksi sebelum data hasil proses dikirim ke *server* untuk disimpan dalam *database server*. Untuk menunjang proses tersebut akan terjadi *sharing* data antara *server* dan *client*. Proses saring data yang terjadi pada lapisan *client* dapat diuraikan pada Gambar 3.43.



Gambar 3.43. Sharing data dari *server* dan *client*

3.10.2 Desain Teknologi untuk *Client-Server*

3.10.2.1. Desain Teknologi Perangkat Keras

Teknologi minimal yang akan digunakan akan meliputi dua bagian yaitu teknologi perangkat keras untuk *server* dan *client*. Adapun teknologi perangkat keras tersebut :

A. Teknologi perangkat keras untuk *Server*

- Processor Pentium IV 1.6 GHz atau di atasnya
- Mainboard Intel 845GL atau di atasnya
- Hardisk 40 GB 7200rpm
- MVM 256MB PC 3200
- Intel Extreme Graphic, LAN onboard
- ASUS CDRW 523252 (untuk Backup data)
- Keyboard + Skroll Mouse
- 15” LG Digital Monitor
- Prolink UPS 500VA

B. Teknologi penunjang untuk *Client*

- Processor Pentium III 667 MHz atau di atasnya
- Mainboard Chipset Intel menyesuaikan dengan procssor
- Hardisk 10 GB 5400rpm
- Visipro SDRAM 128PC 133
- Intel Extreme Graphic, LAN onboard
- ASUS CDROM 52X
- Keyboard + Skroll Mouse

- 15” LG Digital Monitor
- Prolink UPS 500VA

C. Topologi LAN

- Menggunakan topologi “Star”.
- Hub dengan minimal 4 konektor yaitu tiga *client* dan satu *server*.
- Kabel LAN RJ 45.
- Konektor RJ 45 sesuai kebutuhan jumlah koneksi.

3.10.2.2. Desain Teknologi Perangkat Lunak

Teknologi perangkat lunak yang digunakan untuk mengimplementasikan desain, adalah sebagai berikut :

a. Server

Sistem Operasi NT 4 / Windows xp

Basis Data Server *Microsoft. SQL Server 2005*

b. Client

Sistem Operasi Win 9x (98SE) /Win 2000/Windows XP

Aplikasi *Microsoft Visual Basic 6.0*

BAB IV

IMPLEMENTASI

4.1 Aplikasi yang Digunakan

Aplikasi yang digunakan untuk mengimplementasikan atau pengkodean dan membua tabel, *form*, laporan dan validasi pada Sistem Informasi Penjualan Obat di Apotek berbasis *Client-Server* ini adalah :

a. Microsoft SQL Server 2005

Microsoft SQL Server 2005 adalah aplikasi database server yang dipakai untuk menciptakan obyek-obyek basisdata seperti : Tabel, Triger, *Stored Procedure*, *database user*, aturan yang menyangkut validasi data dan menangani masalah keamanan data.

b. *Microsoft Visual Basic 6.0*.

Microsoft Visual Basic 6.0. adalah aplikasi pemrograman yang dipakai untuk pengkodean sebagian besar obyek sistem informasi seperti obyek, *form*, dan *report*.

4.2 Arsitektur Koneksi antar Aplikasi dengan Basisdata

Telah dijelaskan pada BAB III bagian perancangan teknologi bahwa sistem yang akan dibangun menggunakan teknologi *Client-Server Database*. Pada sistem *Client-Server*, *client* dapat meminta bagian data tertentu dengan mengirim statement *SQL Server* sebagai penyedia data.

Komunikasi antar aplikasi (pada *client*) dan data (pada *server*) ini sebenarnya terjadi lewat suatu interface yang disebut *Microsoft ActiveX DataObject* (ADO). ADO memungkinkan untuk membangun aplikasi *client-server* yang dapat mengakses dan memanipulasi data dari sebuah *server* dengan terlebih dahulu menentukan providernya. Pada sistem ini mendeklarasikan variabel koneksi dari ADO Library, yaitu *ADODB.Connection* yang diwakili oleh variabel *K24 PHARMACY*, selanjutnya variabel ini dari objek connection akan membuka database yang telah dideklarasikan lengkap dengan provider yang digunakan dan basis data yang menjadi target adalah *K24 PHARMACY*.

4.3 Membuat Antarmuka (*Interface*)

Antar muka atau interface adalah sarana komunikasi antar sistem dengan user. Antar muka yang ada dalam sistem ini adalah menu dan *form* yang masing-masing mempunyai kegunaan yang berbeda.

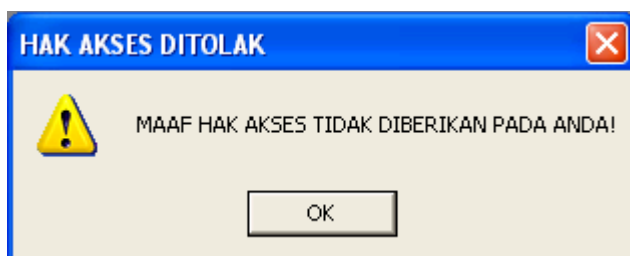
4.3.1 Layar Utama

Tampilan pertama yang akan dijumpai pengguna, adalah tampilan konfirmasi untuk proses login basis data. Pada tampilan ini, *form* akan meminta user untuk memasukkan Id User, Nama User, Status, *Password* dan akan me- *masking password* dengan karakter ‘*’.



Gambar 4.1 form Logim

Jika user meng-*cklik* tombol login maka sistem akan melakukan validasi dengan melihat apakah nama tersebut telah terdaftar pada user di basisdata *server*. Jika nama telah terdaftar, sistem akan mengecek apakah password yang dimasukkan benar. Jika *Id_user*, *Status* dan *Password* dimasukkan salah, maka akan muncul tampilan seperti gambar 4.2.



Gambar 4.2 Hak akses ditolak

Jika *Id_user*, *Status* dan *Password* yang dimasukkan telah terdaftar dan benar, maka akan masuk ke menu utama seperti pada gambar 4.3. Hak akses yang diberikan tergantung group statusnya yaitu seperti yang sudah dijelaskan sebelumnya, terdiri dari tiga status hak yang diberikan yaitu : admin, menejer dan kasir. Pembagian group akan tampak pada tabel 4.1. berikut :

Menu Bar	Menu Item	Group User		
		ADMIN	MNGR	KASIR
Informasi Obat	Data Obat	<i>Enabled</i>	<i>Disable</i>	<i>Disable</i>
	Detail Obat	<i>Enabled</i>	<i>Disable</i>	<i>Disable</i>
Transaksi	Cari Obat	<i>Enabled</i>	<i>Disable</i>	<i>Enabled</i>
	Penjualan Resep	<i>Enabled</i>	<i>Disable</i>	<i>Enabled</i>
	Penjualan Non Resep	<i>Enabled</i>	<i>Disable</i>	<i>Enabled</i>
Laporan		<i>Enabled</i>	<i>Enabled</i>	<i>Disable</i>
Ubah Hak Akses		<i>Enabled</i>	<i>Disable</i>	<i>Disable</i>
Help		<i>Enabled</i>	<i>Enabled</i>	<i>Enabled</i>
Keluar		<i>Enabled</i>	<i>Enabled</i>	<i>Enabled</i>

Tabel 4.1 Pembagian Hak Akses Menu



Gambar 4.3 Layar Utama Aplikasi

4.3.2. Menu

Menu Utama adalah menu yang tersusun secara horisontal pada bagian atas layar utama. Menu Utama seperti pada Gambar 4.3 ini mempunyai enam child menu atau yang disebut Menu Bar yaitu : Informasi Obat, Transaksi, Laporan, Ubah Hak Akses, Help dan Keluar. Menu bar yang terdapat pada Menu Utama akan diuraikan atau dijelaskan satu persatu sebagai berikut :

Sebelum masuk ke Menu Utama user harus login terlebih dahulu untuk dapat masuk ke Menu Utama. Sistem yang dibangun ini mengelompokkan pemakainya kedalam beberapa *group*. Pengelompokan ini bertujuan untuk memisahkan hak akses data karena adanya pembagian tanggung jawab dan pembagian tugas dalam struktur organisasi. Dalam hal ini pengelompokan *group* didasarkan pada tugas serta ruang lingkup kerja. Adapun pembagian *group* meliputi : admin, kasir dan manajer. Strategi yang digunakan untuk mengatur hak akses data adalah lewat pengaturan menu hak akses, karena data diolah lewat *form* dan *form* ini oleh menu. Pembagian hak akses menu dapat dilihat pada tabel 4.1. dan seperti yang telah dijelaskan sebelumnya.

4.3.3. Form

4.3.3.1. Karakteristik Umum

Form merupakan sarana untuk berinteraksi dengan basis data. Lewat *form-form* ini *user* atau pengguna sistem bisa meminta bagian data tertentu, menambah data baru (*insert*), mengubah data (*Update*) dan menghapus data (*Delete*).

Form-form yang ada dalam sistem juga terkoneksi dengan *server*. Koneksi tersebut dilakukan pada *form* FrmLogin dengan menentukan nama koneksi yang telah dibuat. Nama koneksi tersebut adalah MdlKoneksi, setelah koneksi sukses maka *form* dapat digunakan. MdlKoneksi ini bersifat public artinya semua *form* dapat menggunakan untuk mengakses database. Listing program untuk melakukankoneksi ke *server* lewat ADODB. Koneksi ke *server* yang telah dibuat juga harus diputuskan jika *form* Utama Aplikasi ditutup.

4.3.3.2. *Form* pada Menu Bar Informasi Obat

1. Karakteristik *Form* pada Menu Bar Informasi Obat

Form-form yang ada dalam Menu Bar Informasi Obat memiliki fungsi untuk mengetahui Data Obat dan Detail Obat. Hampir semua *form* yang adadalam sistem ini dilengkapi dengan Tombol Navigator Umum yang berfungsi untuk menginstruksikan sistem agar melakukan operasi tertentu. Pada *form* Data Obat terdapat tombol-tombol navigasi. Fungsi masin-masing tombol dapat dilihat pada tabel berikut :

Nama Tombol	Fungsi
Data Baru	berfungsi untuk menambah data obat baru
Edit	untuk mengedit data yang sedang ditampilkan di <i>form</i>
Simpan	Untuk menyimpan data ke simpanan fisik
Hapus	Untuk menghapus data obat
Tambah Stok	Untuk menambah stok obat
Obat Terlaris	Untuk mengetahui obat paling laris
Stok Obat Minimum	Untuk mengetahui stok obat minimum habis 0 atau sisa 10
Keluar	Untuk keluar dari <i>form</i> yang sedang ditampilkan

Tabel 4.2. Fungsi tombol navigator form Data Obat

1.1. *Form* Menu Data Obat

Form Data Obat salah satunya berfungsi untuk memasukkan data obat- obatan yang tersedia di apotek. Setiap memasukkan data obat, dilengkapi dengan mencantumkan jumlah stok minimum, efek terapi, golongan, kemasan serta

satuan obatnya. Data obat yang telah dimasukkan, akan digunakan dalam proses transaksi penjualan obat pada menu Penjualan Resep atau Penjualan Non Resep.

.Form Data Obat dilengkapi dengan tombol navigasi yaitu : Data Baru yang mempunyai fungsi sama dengan Tombol Navigator Umum Tambah yaitu berfungsi untuk menambah data obat baru, tombol Edit berfungsi untuk mengedit data yang sedang ditampilkan di form, tombol Hapus berfungsi untuk menghapus data yang sedang ditampilkan di layar, tombol Obat Terlaris berfungsi untuk menampilkan informasi obat yang paling laris dari obat yang paling banyak terjual, tombol Stok Obat Minimum berfungsi untuk mengetahui jumlah stok obat minimum, jumlah stok obat habis 0 atau sisa 1. data ini digunakan untuk penggandaan obat, tombol Tambah Stok berfungsi untuk memasukkan atau menambah data stok obat. Adapun bentuk form dari Data Obat dapat dilihat pada gambar 4.4 berikut:

The screenshot shows a software window titled "DATA OBAT". Inside, there is a table with four columns: "Kode Obat", "Nama Obat", "Golongan", and "Efek Terap". The table lists various drugs like Albendazol, Alopurinol, Amilorida, Aminofilin, Amitriptilin, Amoksisilin, and Ampisilin. Below the table, it says "Tersedia 146 jenis obat". To the right of this text is a button labeled "TAMBAH STOK". At the bottom of the window, there is a row of six buttons: "DATA BARU", "EDIT", "HAPUS", "OBAT TERLARIS", "STOK OBAT MINIMUM", and "KELUAR".

Kode Obat	Nama Obat	Golongan	Efek Terap
KO-00000001	Albendazol tablet 400 aa	KERAS	Anti Radan
KO-00000002	Alopurinol tablet 100 mg	BEBAS TERBATAS	Anti Biotik
KO-00000003	Amilorida tablet 5 mg (HCl)	BEBAS	Anti Radan
KO-00000004	Aminofilin injeksi 24 mg/ml	NARKOTIKA	Anti Disentr
KO-00000005	Aminofilin tablet 200 mg	BEBAS	Anti Biotik
KO-00000006	Amitriptilin tablet salut 25 mg (HCl)	BEBAS TERBATAS	Anti Biotik
KO-00000007	Amoksisilin Anhidrat tablet 500 mg	PSIKOTROPIKA	Anti Radan
KO-00000008	Amoksisilin Anhidrat kapsul 250 mg	BEBAS	Anti Hestam
KO-00000009	Amoksisilin sirup kering 125 mg/ 5 ml	PSIKOTROPIKA	Anti Biotik
KO-00000010	Ampisilin serbuk injeksi i.m./i.v. 1000 mg/ml	BEBAS	Anti Biotik
KO-00000011	Ampisilin serbuk injeksi i.m./i.v. 500 mg/vial	BEBAS TERBATAS	Anti Jantun
KO-00000012	Amoksisilin sirup kering 125 mg/ 5 ml	BEBAS	Anti Biotik

Gambar 4.4 Data Obat

1.1.a. Tombol Data Baru

Form Data Baru adalah *form* yang digunakan untuk menambah data baru obat. Pada *form* ini mempunyai tombol navigasi umum hanya tombol Simpan. Jika ingin menambah data obat harus meng-*klick* tombol Data Baru, Kemudian isi textbox yang disediakan oleh *form* Tambah Data Obat kemudian meng-*klick* Simpan untuk menyimpan data obat baru pada tabel Obat.

Adapun bentuk dari *form* Data Baru dapat dilihat pada Gambar 4.5 berikut

Gambar 4.5 Layar Utama Aplikasi

Obyek penting lainnya pada *form* ini adalah obyek combo status yang juga merupakan navigator. Obyek combo memuat status obat untuk menfilter obat yaitu untuk obat dengan resep atau non resep.

Untuk pengisian kode dan nama user, meng-click double pada textbox Kode/Nama User kemudian akan tampil obyek listview yang berisi data user pemakai sistem. Adapun tampilan seperti pada gambar 4.4 setelah obyek listview tertampil kemudian user meng-click obyek listview dengan memilih salah satu record atau dengan menekan tombol enter dari keyboard setelah user memilih

salah satu record. Hal ini dilakukan untuk melakukan pencarian data user pemakai sistem. Adapun *form* tambah data obat untuk pencarian data yang menampilkan obyeklistview tampak pada Gambar 4.6 berikut :

TAMBAH DATA OBAT	
Kode Obat	KO-00000147
Nama Obat	
Kode/Nama User	
Kode/Efek Terapi	
Kode/Nama Kemasan	
Kode/Nama Satuan	
Kode/Nama Golongan	

ID User	Nama User
A	A
B	B
C	C

Gambar 4.6 Layar Utama Aplikasi

Untuk pengisian textbox untuk Efek Terapi, Nama Kemasan, Satuan dan Golongan sama seperti pada pengisian untuk Kode>Nama User.

1.1.b. Tombol Edit

Form ini pada Gambar 4.5 digunakan juga untuk Edit data obat dimana tombol simpan akan berubah menjadi Edit, caption dari *form* berubah menjadi “Edit Data Obat” tinggal pilih data yang akan di-edit. Tombol Simpan akan berubah menjadi “EDIT” Kemudian click Edit untuk meng-edit data dan menyimpan di simpanan fisik.

1.1.c. Tombol Tambah Stok

Form juga Pada Gambar 4.5 ini digunakan juga untuk tambah data stok obat dimana tombol simpan akan berubah menjadi Tambah Stok. Seperti pada *form* Edit, *form* Tambah Stok akan men-disable semua textbox dan akan muncul textbox untuk tambah stok dan label Stok Obat akan berubah menjadi Tambah Stok.

1.1.d. Tombol Obat Terlaris

Tombol Obat terlaris akan memanggil *form* frmObatMin. Fungsi dari *form* ini adalah untuk mengetahui rangking obat dari yang paling banyak terjual. Adapun bentuk dari *form* Obat Terlaris dapat dilihat pada Gambar 4.7 berikut :

The screenshot shows a software window titled "RANGKING PENJUALAN OBAT". Inside, there is a table with two columns: "Kode Obat" and "Nama Obat". The table lists 13 drugs. Below the table, a status bar indicates "Terdapat 13 jenis obat yang dirangking".

Kode Obat	Nama Obat
KO-00000002	Alopurinol tablet 100 mg
KO-00000005	Aminofilin tablet 200 mg
KO-00000003	Amilorida tablet 5 mg (HCl)
KO-00000009	Amoksisilin sirup kering 125 mg/
KO-00000146	Vitamin B Kompleks tablet
KO-00000007	Amoksisilin Anhidrat tablet 500
KO-00000028	Deksametason injeksi i.v. 5 mg
KO-00000001	Albendazol tablet 400 mg
KO-00000059	Gentamisin injeksi 80 mg/ml (sebagai
KO-00000101	Natrium Klorida larutan infus 0.9%
KO-00000042	Ekstrak Belladonna tablet 10 mg
KO-00000050	Ekstrak Belladonna tablet 20 mg

Terdapat 13 jenis obat yang dirangking

Gambar 4.7 Rangkin Penjualan Obat

1.1.e. Tombol Stok Obat Minimum

Tombol Stok Obat Minimum akan memanggil *form* frmObatMin. Fungsi dari *form* ini adalah untuk mengetahui stok obat minimum habis 0 atau sisa 1 yang nantinya akan digunakan untuk melakukan penggandaan barang Adapun bentuk dari *form* Obat Terlaris dapat dilihat pada Gambar 4.8 berikut :

DATA OBAT MINIMUM	
Cetak Keluar	
Kode Obat	Nama Obat
KO-00000046	Eritromisin sirup 200 mg/ 5 r
KO-00000049	Fenilbutason tablet 200 mg
KO-00000051	Fenobarbital injeksi i.m./i.v. 50
KO-00000050	Fenobarbital tablet 30 mg
KO-00000053	Fenoksimetil Penisilin tablet 250
KO-00000052	Fenoksimetil Penisilin tablet 500
KO-00000054	Fitomenadion (Vitamin K) injeksi 10
KO-00000055	Fitomenadion (Vitamin K) tablet 500
KO-00000056	Flour tablet 0,5 mg
KO-00000057	Furosemid injeksi i.v./i.m. 10 mg
KO-00000058	Furosemid tablet 40 mg
KO-00000059	Centamin injeksi 80 mg/ml (ceh)
Terdapat 94 jenis obat yang stoknya minimum	

Gambar 4.8 Data Obat Minimum

1.2. Form Menu Detail Obat

Form Menu Detail Obat memiliki sub-sub menu yaitu : Detail Obat sub menu Efek Terapi Obat, Detail Obat sub menu golongan Obat, Detail Obat sub menu Kemasan Obat dan Detail Obat sub menu Satuan Obat.

Sebelum memasukkan data obat di apotek, terlebih dahulu mengisi data- data yang terdapat pada Detail Obat sub menu Efek Terapi Obat, Detail Obat sub menu golongan Obat, Detail Obat sub menu Kemasan Obat dan Detail Obat sub menu Satuan Obat yang nantinya akan digunakan untuk pengisian data obat secara keseluruhan.

Sub-sub menu yang terdapat di menu Detail Obat hanya menggunakan satu form yaitu frmIdObat. Form ini hanya menggunakan dua tombol navigasi yaitu simpan dan keluar.

Adapun bentuk form dari Detail Obat sub menu Efek Terapi Obat adalah sebagai berikut :

Kd Efek Trp	Nama Efek Terapi
T00001	Anti Biotik
T00002	Anti Radang
T00003	Anti Hestamin
T00004	Anti Disentri
T00005	Anti Jantung

Gambar 4.9 Data Efek Terapi

Tampilan yang akan membedakan dari frmIdObat ini adalah caption, label dan DTBGrid. Untuk sub menu dari Golongan Obat : caption akan berubah menjadi

“DATA GOLONGAN OBAT” , label Kode Efek Terapi dan Nama Efek Terapi akan berubah menjadi Kode Golongan dan Nama Golongan, DTBGrid Kd_Efek_Trp dan Nama Efek Terapi akan berubah menjadi Kode Golongan dan Nama Golongan. Perubahan ini akan sama untuk sub menu Kemasan dan sub menu Satuan.

4.3.3.3. Form Pada Menu Bar Transaksi

1. Karakteristik *Form* pada Menu Bar Informasi Obat

Pada Menu Bar Transaksi Obat terdapat tiga menu yaitu : Menu Cari Obat, Menu Penjualan Resep dan Menu Penjualan Non Resep. *Form-form* yang ada dalam menu Bar Transaksi memiliki fungsi untuk melakukan transaksi penjualan baik penjualan resep maupun penjualan non resep.

2. *Form* Cari Obat.

Form cari obat adalah *form* yang digunakan untuk melakukan pencarian Obat. Dalam melakukan pencarian obat didasarkan pada kategori tertentu antara lain kategori berdasarkan Nama Obat, Kemasan Obat, satuan Obat, Efek Terapi dan Golongan. Pada *form* ini terdapat tombol untuk mencari data obat yang berguna untuk mengetahui informasi obat. Adapun *form* cari obat dapat dilihat pada Gambar 4.10 berikut :

Nama Obat	Kode Obat	Stok Barang
Amoksisilin Anhidrat	KO-00000008	100
Amoksisilin Anhidrat	KO-00000007	100
Amoksisilin sirup kerir	KO-00000009	1100
Amoksisilin sirup kerir	KO-00000012	3455
Parasetamol sirup 12	KO-00000108	1
Parasetamol tablet 1	KO-00000110	1
Parasetamol tablet 5	KO-00000109	1

Gambar 4.10 Cari Obat

Untuk pencarian obat user memilih kategori pencarian obat, kemudian memasukkan angka atau huruf pada obyek textbox. Dengan menekan tombol

cari atau tekan tombol Alt + c maka informasi obat akan ditampilkan. Adapun listing

pencarian obat adalah sebagai berikut :

```
Private Sub CmdCari_Click()
Dim TblCari As ADODB.RecordsetIf
Me.TxtCari.Text = "" Then
MsgBox "Masukkan kata kunci yang akan dicari!!!", vbOKOnly + vbInformation, "Konfirmasi"
Me.TxtCari.SetFocus
Exit SubEnd If

If OptNamaObat.Value = True Then
PerintahSQL.CommandText = "SELECT nama_obat as [Nama Obat], kode_obat as [Kode Obat], stk_brg as [Stok Barang], harga_jual as [Harga Jual]
" & _
"FROM Obat where nama_obat LIKE '%" & Me.TxtCari.Text & "%' order by nama_obat asc"Set TblCari =
PerintahSQL.Execute
Set Me.TDBGrid1.DataSource = TblCari Me.TDBGrid1.Columns(3).NumberFormat =
"Currency"
End If

If OptKemasan.Value = True Then
PerintahSQL.CommandText = "SELECT Kemasan.nama_kemasan as [Nama Kemasan], Kemasan.kode_kemasan as [Kode Kemasan], " & _ "nama_obat
as [Nama Obat], kode_obat as [Kode Obat], stk_brg as [Stok], harga_jual as [Harga Jual] " & _
"FROM Obat INNER JOIN Kemasan ON Obat.kode_kemasan = Kemasan.kode_kemasan where kemasan.nama_kemasan LIKE '%" &
Me.TxtCari.Text & "%' " & _
"order by kemasan.nama_kemasan asc"Set TblCari =
PerintahSQL.Execute
Set Me.TDBGrid1.DataSource = TblCari Me.TDBGrid1.Columns(5).NumberFormat =
"Currency"
End If

If OptSatuan.Value = True Then
PerintahSQL.CommandText = "SELECT Satuan.nama_satuan as [Nama Satuan], Satuan.kode_satuan as [Kode Satuan], " & _ "Obat.nama_obat as
[Nama Obat], Obat.kode_obat as [Kode Obat], Obat.stk_brg as [Stok Barang], Obat.harga_jual as [Harga Jual] " & _ "FROM Obat INNER JOIN
Satuan ON Obat.kode_satuan = Satuan.kode_satuan " & _
"where satuan.nama_satuan LIKE '%" & Me.TxtCari.Text & "%' order by satuan.nama_satuan asc"Set TblCari =
PerintahSQL.Execute
Set Me.TDBGrid1.DataSource = TblCari Me.TDBGrid1.Columns(5).NumberFormat =
"Currency"

End If

If OptEfek.Value = True Then
PerintahSQL.CommandText = "SELECT Efek_Trp.efek_terapi as [Nama Efek], Efek_Trp.kode_efek_t as [Kode Efek], " & _ "Obat.nama_obat
as [Nama Obat], Obat.kode_obat as [Kode Obat], Obat.stk_brg as [Stok], Obat.harga_jual as [Harga Jual] " & _ "FROM Obat INNER JOIN
Efek_Trp ON Obat.kode_efek_t = Efek_Trp.kode_efek_t " & _
"where Efek_Trp.efek_terapi LIKE '%" & Me.TxtCari.Text & "%' order by efek_trp.efek_terapi asc"Set TblCari =
PerintahSQL.Execute
Set Me.TDBGrid1.DataSource = TblCari Me.TDBGrid1.Columns(5).NumberFormat =
"Currency"

End If

If OptGolongan.Value = True Then
PerintahSQL.CommandText = "SELECT Golongan.nama_gol as [Nama Golongan], Golongan.kode_gol as [Kode Golongan],
Obat.nama_obat as [Nama Obat], " & _
"Obat.kode_obat as [Kode Obat], Obat.stk_brg as [Stok], Obat.harga_jual as [Harga Jual] FROM Obat INNER JOIN " & _ "Golongan
ON Obat.kode_gol = Golongan.kode_gol " & _
"where Golongan.nama_gol LIKE '%" & Me.TxtCari.Text & "%' order by golongan.nama_gol asc"Set TblCari =
PerintahSQL.Execute
Set Me.TDBGrid1.DataSource = TblCari Me.TDBGrid1.Columns(5).NumberFormat =
"Currency"
End If
```

3. Form Penjualan Resep

Form Penjualan Resep adalah *form* yang digunakan untuk melakukan transaksi penjualan dengan resep. Pada form ini mempunyai tombol Input Data Dokter, Input Data Pasien, Hapus dari Daftar, Masukkan Daftar, Cetak Dan Keluar. Input Data Dokter digunakan untuk menambah data baru dokter, Input Data Pasien digunakan untuk menambah data baru pasien, Hapus dari daftar digunakan untuk menghapus dari daftar pembelian obat, Masukkan Daftar digunakan untuk menambah daftar pembelian obat. Cetak digunakan untuk menyimpan data pembelian obat dan mencetak pembelian obat pada printer, Keluar untuk keluar dari *form*. Adapun bentuk dari *form* penjualan resep dapat dilihat pada Gambar 4.10 berikut :

NOTA RESEP

Tanggal/ No. Nota: 06-08-2006 NR-060806-001

ID / Nama User:

No. Resep:

Kode / Nama Dokter:

Kode / Nama Pasien:

INPUT DATA DOKTER

INPUT DATA PASIEN

Kode Obat:

Nama Obat:

Hapus dari Daftar:

Masukkan Daftar:

Harga Satuan / Stok:

Jumlah/Total:

Biaya Racik:

Grand Total:

Bayar:

Kembali:

CETAK:

KELUAR:

Double-Click Mouse Anda Untuk Data Lebih Detail

Gambar 4.10 Penjualan Resep

Obyek penting lain pada *form* ini adalah tanggal dan no nota secara otomatis akan diberikan. Adapun listing transaksi baru program untuk tanggal dan no nota otomatis adalah sebagai berikut :

```
Sub TransBaru() Dim
Nota As String

Me.TxtTanggal = Format(Date, "dd-mm-yyyy")

PerintahSQL.CommandText = "select * from Penj where tgl_trans = ? and no_nota like 'N%' order by no_nota desc"
PerintahSQL.Parameters(0).Value = Date

Set Tabel = PerintahSQL.Execute

If Tabel.RecordCount = 0 Then

Nota = "NR-" & Trim(Format(Date, "ddmmyy")) & "-001"Else
```

Untuk pengisian kode dan nama user, meng-click double pada textbox Kode>Nama User kemudian akan tampil obyek listview yang berisi data user pemakai sistem. Adapun tampilan seperti pada gambar 4.11 setelah obyek listviewtertampil kemudian user meng-click obyek listview dengan memilih salah satu record atau dengan menekan tombol enter dari keyboard setelah user memilih salah satu record. Hal ini dilakukan untuk melakukan pencarian data user pemakaisistem. Adapun listing program pencarian data nama user yang berdasarkan kode dan nama user adalah sebagai berikut :

```

Sub tekantombol3() Me.LVObatDokterPasien.Visible =
True

Me.LVObatDokterPasien.Top = Me.TxtIdUser.Top + (Me.TxtIdUser.Height / 2)
Me.LVObatDokterPasien.Left = Me.TxtIdUser.Left

With Me.LVObatDokterPasien

.ListItems.Clear

.ColumnHeaders.Clear

.ColumnHeaders.Add , , "Id User", 1250

.ColumnHeaders.Add , , "Nama Lengkap", 1750

POSITION = "USER"

PerintahSQL.CommandText = "SELECT id_user, nama_lkp FROM Login order by nama_lkp asc"Set Tabel =
PerintahSQL.Execute

If Tabel.RecordCount <> 0 Then

```

Adapun *form* Nota Resep untuk pencarian data yang menampilkan obyeklistview tampak pada Gambar 4.11 berikut :

Id User	Nama Lengkap
A	A
B	B
C	C

Gambar 4.11 Listview Nota Resep

Obyek listview juga digunakan untuk pengisian dokter, pasien dan obat yang akan dibeli. Listing program listview untuk pengisian dokter sebagai berikut :


```

Sub tekantombol()
Me.LVObatDokterPasien.Visible = True

Me.LVObatDokterPasien.Top = Me.TxtKdDoc.Top + (Me.TxtKdDoc.Height / 2)
Me.LVObatDokterPasien.Left = Me.TxtKdDoc.Left

With Me.LVObatDokterPasien

.ListItems.Clear

.ColumnHeaders.Clear

.ColumnHeaders.Add , , "Kode Dokter", 1500

.ColumnHeaders.Add , , "Nama Dokter", 2500

        POSISI = "DOKTER"

PerintahSQL.CommandText = "Select kode_doc, nama_doc from dokter order by nama_doc asc"Set
Tabel = PerintahSQL.Execute

```

Listing program listview untuk pengisian pasien sebagai berikut :

```

Sub tekantombol1()

Dim pasien As ADODB.Recordset Me.LVObatDokterPasien.Visible =
True

Me.LVObatDokterPasien.Top = Me.TxtKdPas.Top + (Me.TxtKdPas.Height / 2)
Me.LVObatDokterPasien.Left = Me.TxtKdPas.Left

With Me.LVObatDokterPasien

.ListItems.Clear

.ColumnHeaders.Clear

.ColumnHeaders.Add , , "Kode Pasien", 1500

.ColumnHeaders.Add , , "Nama Pasien", 2500

        POSISI = "PASIEN"

PerintahSQL.CommandText = "Select kode_pas, nama_pas from pasien order by nama_pas asc"Set
Tabel = PerintahSQL.Execute

```

Listing program listview untuk pengisian Obat sebagai berikut :

```
Sub tekantombol2()

Dim obat As ADODB.Recordset
Me.LVObatDokterPasien.Visible = True

Me.LVObatDokterPasien.Top = Me.TxtKodeObat.Top + (Me.TxtKodeObat.Height / 2)
Me.LVObatDokterPasien.Left = Me.TxtKodeObat.Left

With Me.LVObatDokterPasien

.ListItems.Clear

.ColumnHeaders.Clear

.ColumnHeaders.Add , , "Kode Obat", 1250

.ColumnHeaders.Add , , "Nama Obat", 1750

.ColumnHeaders.Add , , Format("Harga Satuan", "currency"), 1000

.ColumnHeaders.Add , , "Stok Obat", 1000

POSISI = "OBAT"

PerintahSQL.CommandText = "SELECT kode_obat, nama_obat, stk_brg, harga_jual FROM Obat order by nama_obat asc"Set Tabel =
PerintahSQL.Execute

PerintahSQL.CommandText = "Select * from obat where (nama_obat like '%" & TxtNamaObat.Text & "%') or (kode_obat like '%" &
TxtNamaObat.Text & "%') order by nama_obat asc"
```

3.a *Form* Input Data Dokter

Form Input Data Dokter adalah *form* yang digunakan untuk modifikasi pada tabel dokter.

Pada *form* ini terdapat nfigator yaitu : Data Baru, Edit, Hapus dan Keluar. Fungsi dari Data Baru adalah untuk menambah data baru dokter, Edit untuk meng-edit data dokter, Hapus untuk menghapus data dokter dan Keluar untuk keluar dari *form* yang sedang ditampilkan. Bentuk dari *form* Input Data Dokter dapat dilihat pada Gambar 4.12 berikut

:

The screenshot shows a window titled "DATA DOKTER". Inside, there is a table with the following data:

	Kode Dokter	Nama Dokter	Alamat
▶	DOK-061127-001	Lukas Trim, Dr. Spd	jln beo no.105 yog
	DOK-061127-002	Kenz Dr. Spd	jln Nangka no 208 p

Below the table is a large blue rectangular area. At the bottom of the window are four buttons: "DATA BARU", "EDIT", "HAPUS", and "KELUAR".

Gambar 4.12 Data Dokter

Pada *form* Data Dokter tombol Data Baru akan menuju *form* Tambah DataDokter. *Form* Tambah data dokter dapat dilihat pada Gambar 4.13 berikut.

The screenshot shows a window titled "TAMBAH DATA DOKTER". It contains four input fields with labels: "Kode Dokter", "Nama Dokter", "Alamat", and "Telp". Below these fields are two buttons: "SIMPAN" and "KELUAR".

Gambar 4.13 Input Data Dokter

Form Tambah Data Dokter ini juga digunakan untuk melakukan Edit Data Dokter. *Form* ini juga digunakan untuk menambah data pasien dan juga untuk edit data pasien. Jika akan meng-edit data dokter maupun pasien maka tombol Simpan akan berubah menjadi Edit.

Listing program untuk Simpan atau Edit Data Dokter Maupun Pasien adalah sebagai berikut :

```
Private Sub CmdSimpan_Click()

If Me.Caption = "TAMBAH DATA PASIEN" Then

PerintahSQL.CommandText = "INSERT INTO PASIEN (kode_pas, nama_pas, umur, jenis_kelamin, almt_pas,
telepon_pas) " & _

" VALUES (?, ?, ?, ?, ?, ?)"

PerintahSQL.Parameters(0).Value = Me.TxtKdPasien.Text
PerintahSQL.Parameters(1).Value = Me.TxtNama.Text
PerintahSQL.Parameters(2).Value = Me.TxtUsia.Text
PerintahSQL.Parameters(3).Value = Me.TxtSex.Text
PerintahSQL.Parameters(4).Value = Me.TxtAlamat.Text
PerintahSQL.Parameters(5).Value = Me.TxtTelp.Text
PerintahSQL.Execute

Unload Me

FrmMenuUtama.StatusBar1.Panels(1) = "DATA PASIEN"End If

If Me.Caption = "EDIT DATA PASIEN" Then

PerintahSQL.CommandText = "UPDATE Pasien SET nama_pas = ?, umur = ?, jenis_kelamin = ?, almt_pas = ?,
telepon_pas = ? " & _

"where kode_pas = ?" PerintahSQL.Parameters(0).Value =
Me.TxtNama.Text PerintahSQL.Parameters(1).Value =
Me.TxtUsia.Text PerintahSQL.Parameters(2).Value =
Me.TxtSex.Text PerintahSQL.Parameters(3).Value =
Me.TxtAlamat.Text PerintahSQL.Parameters(4).Value =
Me.TxtTelp.Text PerintahSQL.Parameters(5).Value =
Me.TxtKdPasien.TextPerintahSQL.Execute

Unload Me

FrmMenuUtama.StatusBar1.Panels(1) = "DATA PASIEN"End If

If Me.Caption = "TAMBAH DATA DOKTER" Then

PerintahSQL.CommandText = "INSERT INTO dokter (kode_doc, nama_doc, almt_doc, telepon_doc) " & _
VALUES (?, ?, ?, ?)"

PerintahSQL.Parameters(0).Value = Me.TxtKdPasien.Text
PerintahSQL.Parameters(1).Value = Me.TxtNama.Text
PerintahSQL.Parameters(2).Value = Me.TxtAlamat.Text
PerintahSQL.Parameters(3).Value = Me.TxtTelp.Text
```

Listing program untuk hapus data dokter dan hapus data pasien adalah sebagai berikut :

```
Private Sub CmdHapus_Click()
Dim vHapus As String

On Error GoTo Pesan

If Me.Caption = "DATA PASIEN" Then
Me.TDBGrid1.Col = 1

If MsgBox("Apakah yakin akan menghapus data pasien ini?", vbYesNo + vbQuestion, "PERHATIAN") = vbYes Then
Me.TDBGrid1.Col = 0

vHapus = Me.TDBGrid1.Text

PerintahSQL.CommandText = "DELETE FROM pasien WHERE kode_pas = ?"
PerintahSQL.Parameters(0).Value = vHapus

PerintahSQL.ExecuteCall
Form_Activate

End IfElse

Me.TDBGrid1.Col = 1

If MsgBox("Apakah yakin akan menghapus data dokter ini?", vbYesNo + vbQuestion, "PERHATIAN") = vbYes Then
Me.TDBGrid1.Col = 0
```

3.b. *Form* Input Data Pasien

form Input Data Pasien adalah *form* yang digunakan untuk modifikasi pada tabel Pasien. *Form* ini sama dengan *form* yang digunakan oleh *form* data dokter. Pada *form* ini terdapat nafigator yaitu : Data Baru, Edit, Hapus dan Keluar. Fungsi dari Data Baru adalah untuk menambah data baru Pasien, Edit untuk meng-edit data pasien, Hapus untuk menghapus data pasien dan Keluar untuk keluar dari *form* yang sedang ditampilkan. Bentuk dari *form* Input Data Dokter dapat dilihat pada Gambar 4.14 berikut :

	Kode Pasien	Nama Pasien	Sex	Umur	Alar
▶	PAS-061127-001	Agus Tri W	L	21	jln. r
	PAS-061127-002	Lena W	P	23	jln. B

DATA BARU EDIT HAPUS KELUAR

Gambar 4.14 Data Dokter

Form Tambah data dokter sebenarnya sama dengan *form* yang digunakan oleh *form* tambah dokter. Pada *form* ini kode pasien dibuat secara otomatis. Pada saat user menekan tombol Data Baru pada *form* Data Pasien, akan muncul *form* Tambah Data Pasien dan kode pasien secara otomatis akan didapat. Adapun bentuk dari *form* dapat dilihat pada Gambar 4.15 berikut :

TAMBAH DATA PASIEN

Kode Pasien PAS-061203-001

Nama Pasien

Alamat

Telp

Jenis Kelamin/Usia

SIMPAN KELUAR

Gambar 4.15. Tambah Data pasien

Adapun listing program untuk kode pasien secara otomatis adalah sebagai berikut:

```
Sub databaru()

If Me.Caption = "TAMBAH DATA PASIEN" ThenDim
pasNow As String

pasNow = "PAS-" & Trim(Format(Date, "yymmdd")) & "%"

PerintahSQL.CommandText = "select * from Pasien where kode_pas like ? order by kode_pas desc"
PerintahSQL.Parameters(0).Value = pasNow

Set Tabel = PerintahSQL.Execute

If Tabel.RecordCount = 0 Then

kdepc = "PAS-" & Trim(Format(Date, "yymmdd")) & "-" & "001"
```

Untuk *form* tambah data pasien sebenarnya hanya satu *form* dengan input datadokter seperti yang dijelaskan sebelumnya pada Input Data Dokter.

3.c. Tombol Masukkan Daftar

Tombol Masukkan Daftar digunakan untuk memasukkan daftar obat resepyang dibeli oleh pembeli kedalam tabel penj_resep, Penj dan Detail_Penj. Obat yag dibeli akan ditampilkan pada listview. Listing program adalah sebagai berikut:

```

Private Sub CmdDaftar_Click()On
Error Resume Next

Dim kontrol As Control

Dim penyelesaian As Integer

'-----mengecek keseluruhan textbox dan harus diisi -----

For Each kontrol In Me.Controls

If TypeOf kontrol Is TextBox ThenIf
kontrol.Text = "" Then

MsgBox "Data belum lengkap", vbOKOnly + vbInformation, "PERHATIAN!!!"
kontrol.SetFocus

Exit SubEnd If

End If Next
kontrol

If Val(Me.TxtJumlah.Text) > Val(TxtStok.Text) Then

MsgBox "Jumlah permintaan lebih besar dari stok. Masukkan permintaan di bawah stok!", vbOKOnly + vbInformation, "Konfirmasi"
Me.TxtJumlah.SetFocus

Exit SubEnd
If

PerintahSQL.CommandText = "select * from Penj_Resep where no_nota = ?"
PerintahSQL.Parameters(0).Value = Me.TxtNoNota.Text

Set Tabel = PerintahSQL.Executelf
Tabel.EOF = True Then

PerintahSQL.CommandText = "INSERT INTO Penj ( no_nota, id_user, tgl_trans) VALUES (?, ?, ?)"
PerintahSQL.Parameters(0).Value = Me.TxtNoNota.Text

```



```
Sub tampiltransaksi()  
With Me.LVTransaksi  
  
.ListItems.Clear  
.ColumnHeaders.Clear  
.ColumnHeaders.Add , , "Kode Obat", 1250  
.ColumnHeaders.Add , , "Nama Obat", 1750  
.ColumnHeaders.Add , , "Harga Satuan", 1500  
.ColumnHeaders.Add , , "Jumlah", 1000  
.ColumnHeaders.Add , , "Total", 1500  
  
PerintahSQL.CommandText = "SELECT Obat.kode_obat, Obat.nama_obat, " & _  
"obat.harga_jual, Detail_Penj.jumlah_brg, Detail_Penj.total " & _  
"FROM Obat INNER JOIN Detail_Penj ON Obat.kode_obat = Detail_Penj.kode_obat " & _  
"WHERE Detail_Penj.no_nota = ?"  
  
PerintahSQL.Parameters(0) = Me.TxtNoNota.TextSet  
Tabel = PerintahSQL.Execute  
  
If Tabel.RecordCount <> 0 Then  
Tabel.MoveFirst  
  
total = 0  
  
Do Until Tabel.EOF
```

3.d. Tombol Hapus Daftar

Tombol Hapus Daftar ini berfungsi untuk menghapus daftar pembelian obat pada obyek listview tampil transaksi dan akan menghapus obat dari tabel Detail_penj, adapun listing program dari Hapus Daftar adalah sebagai berikut :

```
Private Sub CmdHapus_Click()On
Error Resume Next

Dim KdObat As String
Dim beli As String

KdObat = Me.LVTransaksi.SelectedItem.Text

beli = Me.LVTransaksi.SelectedItem.ListSubItems(3)

PerintahSQL.CommandText = "update obat set stk_brg = stk_brg + " & beli & " where kode_obat = " & KdObat & ""
PerintahSQL.Execute
```

3.e. Tombol Cetak

Tombol Cetak digunakan untuk mencetak transaksi penjualan resep. Tombol Cetak akan menyimpan penjualan resep pada database Penj_resep, dan akan meng-update tabel Penj dan tabel Detail_penj dan akan memanggil ActiveRiport RptCetakResep yang kemudian dicetak. Adapun listing program sebagai berikut :

```

Private Sub CmdCetak_Click()

PerintahSQL.CommandText = "select * from Penj where no_Nota = ?"
PerintahSQL.Parameters(0).Value = Me.TxtNoNota.Text

Set Tabel = PerintahSQL.Executelf
Tabel.EOF = False Then

PerintahSQL.CommandText = "UPDATE Penj Set GrandTotal = ? where no_nota = ?"
PerintahSQL.Parameters(0).Value = Me.txtGrandTotal.Text
'PerintahSQL.Parameters(1).Value = Me.TxtRacik.Text PerintahSQL.Parameters(1).Value =
Me.TxtNoNota.Text

PerintahSQL.Execute

PerintahSQL.CommandText = "INSERT INTO Penj_Resep (no_nota, kode_resep, kode_pas, kode_doc) VALUES(?,?,?,?)"

PerintahSQL.Parameters(0).Value = Me.TxtNoNota.Text
PerintahSQL.Parameters(1).Value = Me.TxtNoResep.Text
PerintahSQL.Parameters(2).Value = Me.TxtKdPas.Text
PerintahSQL.Parameters(3).Value = Me.TxtKdDoc.Text
PerintahSQL.Execute

MsgBox "TRANSAKSI SUDAH TERSIMPAN", vbOKOnly + vbInformation, "INFORMASI"End If


With RptCetakResep

.DataControlCetakResep.ConnectionString = K24 PHARMACY.ConnectionString

.DataControlCetakResep.Source = "SELECT Obat.kode_obat, Obat.nama_obat, Obat.harga_jual," & _
"Detail_Penj.jumlah_brg, Detail_Penj.total, Penj.no_nota, Penj_Resep.kode_resep," & _
"Penj_Resep.no_nota , Detail_Penj.no_nota FROM Detail_Penj INNER JOIN " & _

"Obat ON Detail_Penj.kode_obat = Obat.kode_obat INNER JOIN " & _ "Penj ON
Detail_Penj.no_nota = Penj.no_nota INNER JOIN " & _ "Penj_Resep ON
Penj.no_nota = Penj_Resep.no_nota " & _

"WHERE penj.no_nota = '" & Me.TxtNoNota.Text & "'"

.txtGrandTotal.Text = Me.txtGrandTotal.Text
'.TxtBiayaRacik.Text = Me.TxtRacik.Text

```

APOTIK "SIAGA"
Jln. S. Parman No. 36 TEMANGGUNG

Tanggal : 04-12-2006
No. Nota : NR-041206-002
No. Resep : 23
Nama Pasien : Lena W
Nama Dokter : Kenz Dr. Spd

Kode Obat	Nama Obat	Jumlah	Total
KO-0000003	Amloride tablet 5 mg (PMS)	2	Rp38.220
KO-0000004	Aminofilin injeksi 24 mg/ml	1	Rp11.507
KO-00000145	Verapamil tablet 80 mg (PMS)	5	Rp176.000

GRAND TOTAL : Rp296.527
Bayar : Rp300.000
Kembali : Rp13.473

Kasir

Gambar 4.16 Nota Penjualan Resep

4. Form Penjualan Non Resep

Form Penjualan Non Resep adalah *form* yang digunakan untuk melakukan transaksi penjualan tanpa resep. Pada *form* ini mempunyai tombol hampir sama dengan penjualan Resep namun tidak ada tombol untuk Input Data Dokter dan tombol Input Data, *form* ini mempunyai tombol Hapus dari Daftar, Masukkan Daftar, Cetak Dan Keluar. Masing-masing fungsi dari tombol sama dengan yang ada dalam penjualan Resep. Adapun bentuk dari *form* penjualan non resep dapat dilihat pada Gambar 4.17 berikut :

Gambar 4.17 Penjualan Non Resep

Obyek penting lain pada *form* ini adalah tanggal dan no nota secara otomatis akan diberikan. Adapun listing transaksi baru program untuk tanggal dan no nota otomatis adalah sebagai berikut :

```

Sub TransBaru() Dim
Nota As String

Dim kontrol As Control

PerintahSQL.CommandText = "select * from Penj where tgl_trans = ? and no_nota like 'T%' order by no_nota desc"
PerintahSQL.Parameters(0).Value = Date

Set Tabel = PerintahSQL.Execute

If Tabel.RecordCount = 0 Then

```

Untuk pengisian kode dan nama user, meng-click double pada textbox Kode/Nama User kemudian akan tampil obyek listview yang berisi data user pemakai sistem. Adapun tampilan seperti pada gambar 4.18 setelah obyek listview tertampil kemudian user meng-click obyek listview dengan memilih salah satu record atau dengan menekan tombol enter dari keyboard setelah user memilih salah satu record. Hal ini dilakukan untuk melakukan pencarian data user pemakai sistem. Adapun listing program pencarian data nama user yang berdasarkan kode dan nama user sama seperti pada transaksi Resep. Adapun *form* Transaksi Non Resep untuk pencarian data yang menampilkan obyek listview tampak pada Gambar 4.18 berikut :

The screenshot shows a software interface titled "TRANSAKSI NON RESEP". It contains several input fields and a listview. The input fields are: Tanggal (04-12-2006), No. Nota (TNR-041206-002), Id / Nama User (with a dropdown menu), Kode Obat, Nama Obat, Harga Satuan / %nk, Jumlah, and Total (Rp0). The listview displays a table with two columns: "Id User" and "Nama User". The data rows are: A, B, and C. The first row (A, A) is highlighted. To the right of the listview are two buttons: "MASUKKAN DAFTAR" and "HAPUS DARI DAFTAR".

Tanggal	04-12-2006
No. Nota	TNR-041206-002
Id / Nama User	[Dropdown Menu]
Kode Obat	
Nama Obat	
Harga Satuan / %nk	
Jumlah	
Total	Rp0

Id User	Nama User
A	A
B	B
C	C

MASUKKAN DAFTAR

HAPUS DARI DAFTAR

Gambar 4.18 Listview transaksi Non Resep

Obyek listview juga digunakan untuk pengisian obat yang akan dibeli. Listing program listview untuk pengisian data Obat sama seperti pada listing program untuk Transaksi Resep.

4.a. Tombol Masukkan Daftar

Tombol Masukkan Daftar digunakan untuk memasukkan daftar obat non resep yang dibeli oleh pembeli kedalam tabel Penj dan Detail_Penj. Obat yang dibeli akan ditampilkan pada obyek listview. Listing program menampilkan transaksi Penjualan Non Resep pada obyek listview sama seperti listing program untuk menampilkan pada obyek listview pada Transaksi Non Resep.

4.b. Tombol Hapus Dari Daftar

Tombol Hapus Daftar ini berfungsi untuk menghapus daftar pembelian obat pada obyek listview tampil transaksi dan akan menghapus obat dari tabel Detail_penj, adapun listing program dari Hapus Daftar adalah sama dengan listing dari Transaksi Non Resep.

4.c. Tombol Cetak

Tombol Cetak digunakan untuk mencetak transaksi Penjualan Non Resep. Tombol Cetak akan menyimpan penjualan Non Resep pada database Penj, dan akan meng-update tabel Detail_penj dan akan memanggil modul MdLaporan. LaporanTransNonResep yang kemudian dicetak. Adapun listing program sebagai berikut :

```

Private Sub CmdCetak_Click()

PerintahSQL.CommandText = "select * from Penj where no_Nota = ?"
PerintahSQL.Parameters(0).Value = Me.TxtNoNota.Text

Set Tabel = PerintahSQL.Executelf
Tabel.EOF = False Then

PerintahSQL.CommandText = "UPDATE Penj Set GrandTotal = ? where no_nota = ?"
PerintahSQL.Parameters(0).Value = Me.txtGrandTotal.Text
PerintahSQL.Parameters(1).Value = Me.TxtRacik.Text PerintahSQL.Parameters(1).Value =
Me.TxtNoNota.Text

PerintahSQL.Execute

MsgBox "TRANSAKSI SUDAH TERSIMPAN", vbOKOnly + vbInformation, "INFORMASI"End If

```

```

'.....MdLaporan.....
Sub LaporanTransNonResep()
With RptNoRes

.DataControlNonResep.ConnectionString = K24 PHARMACY.ConnectionString

.DataControlNonResep.Source = "SELECT Obat.kode_obat, Obat.nama_obat, Obat.harga_jual, " & _
"Detail_Penj.jumlah_brg , Detail_Penj.total FROM Detail_Penj INNER JOIN Obat ON " & _
"Detail_Penj.kode_obat = Obat.kode_obat INNER JOIN " & _
"Penj ON Detail_Penj.no_nota = Penj.no_nota " & _
"WHERE penj.no_nota = " & FrmNotaNonResep.TxtNoNota.Text & ""

.txtGrandTotal.Text = FrmNotaNonResep.txtGrandTotal.Text

.TxtBayar.Text = FrmNotaNonResep.TxtBayar.Text

```

Adapun report untuk mencetak penjualan resep tampak pada gambar berikut:

4.19

APOTIK "SIAGA"
Jl. S. Paman No. 36 TEMANGGUNG

Tanggal: 04-12-2006
No. Nota: TNR-041206-001

Kode Obat	Nama Obat	Jumlah	Total
KD-0000005	Amoxifin tablet 200 mg	1	Rp11.400
KD-0000007	Amoksisilin Arbidol tablet 500 mg	1	Rp47.317
KD-0000010	Daktarisonatun sirup 10 mg/5 ml (100)	1	Rp136.400
KD-0000054	Flour tablet 0,5 mg	2	Rp33.000

GRAND TOTAL: Rp227.200
Bayar: Rp300.000
Kembali: Rp71.800

Kasir: C
JEMAH24 CERMA7-SEMERUP

Gambar 4.19 Nota Penjualan Non Resep

4.3.3.4. *Form* Pada Menu Bar Laporan

1. Karakteristik *Form* pada Menu Bar Laporan

Form yang ada pada menu bar ini berfungsi sebagai antarmuka bagi user untuk berdialog dengan sistem untuk meng-generate laporan. Laporan yang akan dicetak akan ditentukan oleh beberapa parameter yang diberikan oleh user. Parameter itu adalah :

a. Periode Laporan

Periode laporan yang ada dalam sistem ini yaitu semua laporan menggunakan periode tanggal. Jika user memasukkan tanggal tertentu maka laporan yang akan di-generate adalah laporan yang sampai tanggal tertentu.

b. Kriteria

Kriteria pada laporan merupakan pilihan user untuk mengelompokkan laporan berdasarkan kriteria tertentu

c. Tipe Output

Tipe output pada setiap laporan adalah bertipe *Print Preview* yaitu laporan akan dicetak oleh sistem kelayar atau screen dan juga bisa dicetak pada printout.

2. *Form* Laporan

Adapun bentuk dari *form* yang digunakan sebagai antarmuka untuk mencetak laporan-laporan tampak pada gambar 4.20.

Gambar 4.20 Menu Cetak

Terdapat enam kriteria, dua komponen tanggal, satu textbox dan dua tombol yang terdapat dalam *form* Menu Cetak, ke-enam kriteria tersebut antara lain: Transaksi Resep untuk mengetahui transaksi resep, Transaksi Non Resep untuk mengetahui transaksi non resep, Obat Keras untuk mengetahui transaksi obat keras yang terjual dan stok obat keras, Obat Psikotropika dan Narkotika untuk mengetahui transaksi obat psikotropika dan narkotika yang terjual dan stok obat psikotropika dan narkotika, obat bebas dan bebas terbatas untuk mengetahui transaksi obat bebas dan bebas terbatas yang terjual dan stok obatnya, obat stok minimal habis untuk mengetahui obat-obat yang habis 0 atau sisa 1. komponen tanggal pertama digunakan untuk menentukan tanggal awal dan komponen tanggal kedua digunakan untuk menentukan tanggal akhir. Tombol cetak untuk mencetak laporan pada kertas dan tombol keluar untuk menutup *form*. Sedangkan textbox diisi nama user pemakai sistem. Adapun hasil Laporan output dari masing-masing kriteria sebagai berikut :

a. Kriteria Transaksi Resep

ActiveReport1

DATA TRANSAKSI RESEP

Mulai Tanggal 01/11/2006 Sampai 31/12/2006
 Dicetak pada : Senin, 04-12-2006
 Dicetak oleh : Ophie'x

No. Nota	Tgl Transaksi	Kode Resep	Nama Pasien	Nama Dokter	Grand Total
NR-011206-001	tgl_brans	x	Lena W	Lukas Trins, Dr. Spd	Rp41.275
NR-011206-002	tgl_brans	x	Agus Tri W	Lukas Trins, Dr. Spd	Rp54.925
NR-011206-003	tgl_brans	x	Agus Tri W	Lukas Trins, Dr. Spd	Rp22.165
NR-011206-004	tgl_brans	x	Lena W	Lukas Trins, Dr. Spd	Rp49.790
NR-041206-002	tgl_brans	x	Lena W	Kenz Dr. Spd	Rp286.527
NR-291106-001	tgl_brans	x	Agus Tri W	Kenz Dr. Spd	Rp36.155
NR-291106-002	tgl_brans	x	Lena W	Lukas Trins, Dr. Spd	Rp103.870
NR-291106-003	tgl_brans	x	Agus Tri W	Kenz Dr. Spd	Rp22.165
NR-301106-001	tgl_brans	x	Lena W	Lukas Trins, Dr. Spd	Rp44.921
NR-301106-002	tgl_brans	x	Agus Tri W	Kenz Dr. Spd	Rp80.617
NR-301106-004	tgl_brans	x	Lena W	Kenz Dr. Spd	Rp54.283
NR-301106-005	tgl_brans	x	Lena W	Lukas Trins, Dr. Spd	Rp177.320
NR-301106-006	tgl_brans	x	Lena W	Lukas Trins, Dr. Spd	Rp19.110

Jumlah Nota : 13

Gambar 4.21 Report Transaksi Resep

b. Kriteria Transaksi Non Resep

ActiveReport1

DATA TRANSAKSI NON RESEP

Mulai Tanggal 01/11/2006 Sampai 31/12/2006
 Dicetak pada : Senin, 04-12-2006
 Dicetak oleh : ophie'x

Tgl Transaksi	No. Nota	Nama Operator	Grand Total
01/12/2006	TNR-011206-001	B	Rp63.487
01/12/2006	TNR-011206-002	C	Rp106.116
01/12/2006	TNR-011206-003	C	Rp54.670
01/12/2006	TNR-011206-004	C	Rp47.317
04/12/2006	TNR-041206-001	C	Rp228.200
29/11/2006	TNR-291106-001	C	Rp46.090
29/11/2006	TNR-291106-002	C	Rp32.670
29/11/2006	TNR-291106-003	C	Rp47.317
29/11/2006	TNR-291106-004	C	Rp45.932
29/11/2006	TNR-291106-005	C	Rp27.653
30/11/2006	TNR-301106-001	C	Rp102.996

Jumlah Nota : 11

Gambar 4.22 Report Transaksi Non Resep

c. Kriteria Obat Keras

ActiveReport1

Print... 100% 1/1 Back Forward

DATA OBAT KERAS YANG TERJUAL

Mulai Tanggal 01/11/2006 Sampai 31/12/2006
 Dicetak pada : Senin, 04-12-2006
 Dicetak oleh : oBLo

Kode Obat	Nama Obat	Terjual	Jenis Obat	Stok
KO-0000001	Albendazol tablet 400 aa	2	KERAS	51
KO-00000059	Gentamisin injeksi 80 mg/ml (sebagai Sulfat)	2	KERAS	998

Jumlah 2 jenis obat

Gambar 4.23 Report Obat Keras

d. Kriteria Obat Psikotropika dan Narkotika

ActiveReport1

Print... 100% 1/1 Back Forward

DATA OBAT PSIKOTROPIKA DAN NARKOTIKA TERJUAL

Mulai Tanggal 01/11/2006 Sampai 31/12/2006
 Dicetak pada : Senin, 04-12-2006
 Dicetak oleh : oBLo

Kode Obat	Nama Obat	Terjual	Jenis Obat	Stok
KO-00000004	Aminofilin injeksi 24 mg/ml	2	NARKOTIKA	971
KO-00000007	Amoksisilin Anhidrat tablet 500 mg	6	PSIKOTROPIKA	136
KO-00000009	Amoksisilin sirup kering 125 mg/ 5 ml	8	PSIKOTROPIKA	1072
KO-00000042	Ekstrak Belladonna tablet 10 mg	1	PSIKOTROPIKA	999
KO-00000145	Verapamil tablet 80 mg (HCl)	5	PSIKOTROPIKA	995

Jumlah 5 jenis obat

Gambar 4.24. Report Transaksi obat Psikotropika dan Narkotika

e. Kriteria Obat Bebas dan Bebas Terbatas

ActiveReport1

Print... 100% 1/1 Back Forward

DATA OBAT BEBAS & BEBAS TERBATAS YANG TERJUAL

Mulai Tanggal 01/11/2006 Sampai 31/12/2006
 Dicetak pada : Senin, 04-12-2006
 Dicetak oleh : emma

Kode Obat	Nama Obat	Terjual	Jenis Obat	Stok
KO-0000003	Amilorida tablet 5 mg (HCl)	12	BEBAS	819
KO-0000005	Aminofilin tablet 200 mg	19	BEBAS	116
KO-0000028	Deksametason injeksi i.v. 5 mg/ml	2	BEBAS	163
KO-0000030	Dekstrometorfan sirup 10 mg/5 ml (HBr)	1	BEBAS	29
KO-0000050	Fenobarbital tablet 30 mg	1	BEBAS	999
KO-0000056	Flour tablet 0,5 mg	2	BEBAS	985
KO-0000101	Natrium Klorida larutan infus 0,9 %	1	BEBAS	17
KO-0000146	Vitamin B Kompleks tablet	7	BEBAS	993
KO-0000002	Alopurinol tablet 100 mg	22	BEBAS TERBATAS	915

Jumlah 9 jenis obat

Gambar 4.25 Report Transaksi Bebas dan Bebas Terbatas

f. Kriteria Stok Obat Minimal/Habis

ActiveReport1

Print... 100% 1/1 Back Forward

LAPORAN DATA OBAT HABIS/MINIMUM

Dicetak oleh : andrie'x Dicetak pada: Senin, 04/12/2006

Kode Obat	Nama Obat	Stok
KO-0000010	Ampisilin serbuk injeksi i.m./i.v. 1000 mg/ml	1
KO-0000111	Perfenazin tablet 4 mg (HCl)	1

Gambar 4.21 Report Obat Minimum/Habis

4.3.3.4. Form Pada Menu Bar Ubah Hak Akses

1. Karakteristik *form* pada Menu Bar Ubah Hak Akses

Form yang ada pada menu ini hanya dapat dibuka oleh yang mempunyai group admin saja. Tampilan yang akan dijumpai pengguna untuk masuk ke menu Ubah Hak Akses, adalah tampilan *form* untuk proses login.

2. *form* Ubah Hak Akses

Pada tampilan *form* ubah Hak Akses ini caption dari *form* Login akan berubah menjadi Ubah Hak Akses, *form* akan meminta user untuk memasukkan IdUser, Nama User, Status, *Password* dan akan me-masking *password* dengan karakter '*'.



Gambar 4.22 Login Ubah Hak Akses

Obyek penting lainnya pada *form* ini adalah obyek combo status yang juga merupakan navigator. Obyek combo memuat status dari login atau group dari login. *Form* menu Login Ubah Hak Akses berperan penting salah satunya untuk mengetahui data keseluruhan dari tabel login, jika user lupa akan *password* dapat dilihat pada *form* Data Login. User harus memasukkannya Id User, Nama User, Status pilih pada obyek combo dan password jika user ingin ke *form* Data Login.

Setelah dapat masuk ke menu Ubah Hak Akses, *form* yang akan tampil sepertipada gambar 4.22 berikut :

DATA LOGIN

Id User

Nama Lengkap User

Nama User

Status

Password

SIMPAN **HAPUS** **KELUAR**

Id User	Nama Lengkap	Nama User
A	A	A
B	B	B
C	C	C

Gambar 4.22 Data Login

Form ini memiliki tiga tombol simpan hapus dan keluar. Tombol simpan untuk menyimpan data login baru. Tombol hapus akan muncul ketika meng-klik double pada obyek listview, sedangkan tombol Simpan berubah menjadi tombol Edit. Tombol Keluar berfungsi untuk keluar dari *form* Listing program untuk simpan yang juga dipakai untuk tombol Edit adalah sebagai berikut :

4.3.3.6. Menu Bar Help

Menu Bar Help berfungsi untuk memberikan informasi batuan penggunaan program aplikasi Sistem yang dibuat. Untuk mendapatkan bantuan cara menjalankan program user dapat menekan tombol F1 pada keyboard.

4.3.3.7. Menu Bar Keluar

Menu Bar Keluar berfungsi untuk keluar dari program. Program akan memutuskan koneksi *server* dengan menutup database *server*. Listig program untuk keluar program adalah sebagai berikut :

```
Private Sub Keluar_Click()
    MdlKoneksi.TutupDatabase      ' ----- memanggil modul koneksi Tutup Database
    Unload Me

    Sub TutupDatabase()

        If K24 PHARMACY.State = 1 Then K24
        PHARMACY.CloseSet K24 PHARMACY =
        Nothing

        'Perintah untuk menghapus otomatis
        tmpOn Error Resume Next
```

4.4. Menerapkan Sistem Baru

Pada bagian ini akan dijelaskan mengenai persiapan-persiapan yang harus dilakukan untuk menerapkan sistem yang telah dibuat.

4.4.1 Persiapan Personel

Persiapan personil dimaksudkan adalah melatih user yang nantinya akan menggunakan sistem ini, sehingga dapat berjalan dengan baik. Disamping itu juga dituntut tanggung jawab dan kerja sama dari setiap bagian untuk dapat mensukseskan sistem yang dibuat.

4.4.2 Persiapan Perangkat Komputer

Persiapan perangkat komputer harus dilakukan agar perangkat lunak yang telah dibuat dapat dioperasikan. Aplikasi sistem yang telah dibuat adalah aplikasi

Client-server oleh karena itu secara fisik komputer *client* harus terkoneksi ke *server*.

Pengenai spesifikasi perangkat komputer telah dijelaskan pada bagian Desain Teknologi untuk *Client-Server* Bab III. Persiapan komputer adalah terbagi atas dua yakni :

A. Persiapan sisi *Client*

Persiapan yang dilakukan pada sisi *Client* adalah

1. Meng-install sistem operasi Microsoft Windows Xp SP1 atau versi di atasnya
2. Meng-install Aplikasi Sistem Informasi Penjualan Berbasis *Client-Server*
3. Memastikan seting bahasa indonesia pada windows agar aplikasi untuk uang dengan Rp.

B. Persiapan sisi *Server*

Hal-hal yang harus dipersiapkan pada sisi *Server* adalah :

1. Meng-install Sistem Operasi Microsoft Windows Xp SP 2
2. Meng-install Microsoft SQL *Server* 2005
3. Meng-install Aplikasi Sistem Informasi Penjualan Berbasis *Client-Server*

BAB V

ANALISA HASIL

5.1 Analisa Hasil

Bahwa metode yang dipakai dalam pengembangan sistem yang dibangun menggunakan metodologi pendekatan terstruktur yaitu analisis dan desain terstruktur. Alasan menggunakan metode ini untuk sistem informasi Apotek adalah :

- metodologi analisis dan desain terstruktur mudah dipelajari
- membantu pada tahap implementasi sistem
- hasil dari desain sistem akan didapatkan dokumentasi yang baik

5.2 Analisa Tools

Implementasi Sistem Informasi Penjualan Obat di Apotek yang berbasis Client-Server ini menggunakan :

1. *Microsoft Visual Basic 6.0 (VB)*

Microsoft Visual Basic adalah bahasa pemrograman yang digunakan untuk membuat aplikasi Windows yang berbasis grafis (GUI – *Graphical User Interface*). VB merupakan bahasa pemrograman yang mudah dipelajari. Peran VB sebagai *front-end* terhadap *database*, yang artinya VB menyediakan antarmuka *user* dengan

database. VB sangat mendukung dalam penyediaan *tools* dan pengembangan aplikasi *client-server* pada jaringan.

2. *Microsoft SQL Server 2005*

Microsoft SQL Server 2005 merupakan *database server* karena *database* cukup dilakukan di satu tempat yang disebut *server*, untuk pemanfaatan data dilakukan di komputer lain yang disebut dengan *client* yang terhubung dengan jaringan. *Microsoft SQL Server 2005* mempunyai kemampuan *mem-backup* seluruh *database*, *security* tinggi, kecepatan proses *query*, serta kompatibilitas *platform* dan menjaga integritas data sehingga konsistensi dan pengontrolan terpusat.

5.3 Analisa Hasil Program

Sistem yang telah selesai dibangun adalah sistem Informasi Penjualan Obat di Apotek yang berbasis *Client-Server* dimana sistem ini menangani transaksi penjualan obat baik resep maupun non resep. Untuk Sistem Informasi penjualan Obat baik dengan resep maupun tidak dengan resep, sistem ini dapat menangani beberapa hal antara lain :

- a. Di dalam transaksi penjualan non resep, obat yang harus dengan resep dokter tidak bisa dikeluarkan atau dijual.
- b. Selain dalam perekaman data sistem ini juga menangani dalam pembuatan laporan.

- c. Sistem ini juga dapat memberikan informasi data yang dibutuhkan seperti harga obat, jumlah stok obat maupun informasi lain tentang obat.

Sistem ini masih mempunyai beberapa kekurangan :

- a. Untuk penggandaan obat belum ada distributor penyetok obat
- b. Untuk penjualan resep biaya racik tidak disertakan.

5.4 Kendala yang Dihadapi

Kendala-kendala yang dihadapi dalam pengembangan sistem ini adalah :

- a. Pembuatan antarmuka yang lebih user friendly dan sederhana
- b. Data-data yang diperlukan untuk uji coba program dalam memasukkan data.
- c. Dalam penulisan program, sulit untuk memberi nama untuk *sub function, procedur* dan *variable*.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan dari hasil analisis dan implementasi Sistem Informasi Penjualan Obat di Apotek Berbasis *Client-Server*, maka dapat diambil kesimpulanyaitu :

1. Metode pendekatan terstruktur masih cukup efektif digunakan untuk merancang sistem informasi penjualan obat di apotek berbasis *client-server*
2. Sistem informasi yang telah dibangun diharapkan dapat mendukung, menamngani dan membantu pengolahan data bagi pengguna.
3. Sistem informasi penjualan ini dapat membantu untuk langsung dapat mengetahui stok obat yang ada tidak harus mencari, memastikan obat masih atau habis.
4. Sistem informasi ini dapat membantu dalam pembuatan laporan-laporan.
5. Sistem informasi ini dapat membantu mencegah penjualan obat-obatan terlarang tanpa dengan resep dokter.
6. Teknologi komputer dapat membantu menghasillkan informasi yang lebih teliti, cepat, akurat dan relevan..
7. Sistem informasi ini membantu bagian penjualan (kasir) tidak lagi mencatat obat terjual dalam pencatatan obat terjual.

6.2 Saran

Sistem informasi yang telah dibangun masih memiliki banyak kekurangan sehingga diperlukan pengembangan untuk menyempurnakan aplikasi ini. ada beberapa saran untuk pengembangan selanjutnya, semoga bisa digunakan sebagai masukan, antara lain:

1. Sistem informasi ini hanya membantu dalam hal menangani masalah yang dihadapi selama ini, untuk itu perlu dikembangkan agar lebih lengkap dan sempurna.
2. Sistem yang baru ini dapat dikembangkan lagi ke arah yang lebih luas, yaitu untuk pengembangan dibagian pembelian obat yang berhubungan langsung dengan supplier.

DAFTAR PUSTAKA

- Jogiyanto H.M. Analisis dan Desain , Sistem Informasi : Pendekatan terstruktur Teori dan Praktek Aplikasi Bisnis, Yogyakarta.,Andi Offset,1989,.
- jogiyantoHM, Sistem Teknologi Informasi: Pendekatan Terintegrasi Konsep Dasar, Teknologi, Aplikasi, Pengembangan dan Pengelola, Edisi Pertama, Yogyakarta: Andi,2003.
- Harpiandi, Pemrograman Database dengan ADO Menggunakan Visual Basic 6.0, Kakarta, PT Elex Media Komputindo, 2003.
- Martina Inge Ir, 36 Jam Belajar Komputer Microsoft SQL Server 2000, Jakarta, PT Elex Media Komputindo, 2003.
- Pandia Henry, Visual Basic 6 Tingkat Lanjut, Yogyakarta, Andi Offset,2004.
- Agus J. M. Alam, Manajemen Database dengan Microsoft Visual Basic Versi 6.0, Jakarta, PT Elex Media Komputindo, 2000.

Sumber-sumber lain berupa artikel :

- <http://www.visualbasic.com>
- <http://www.tek-tips.com>
- Microsoft SQL Server 2005 Express Edition Book Online Tutorial.

Json

```
Private Sub CmdLogin_Click() If
    Me.Caption = "LOGIN" Then

    PerintahSQL.CommandText = "Select * from login WHERE id_user = ? and STATUS = ? "
    PerintahSQL.Parameters(0).Value = Me.TxtIdUser.Text

    PerintahSQL.Parameters(1).Value = Me.TxtStatus.TextSet
    Tabel = PerintahSQL.Execute

    If Tabel.EOF = True Then

        MsgBox "MAAF HAK AKSES TIDAK DIBERIKAN PADA ANDA!", vbOKOnly + vbExclamation, "HAK AKSES DITOLAK"

    Exit SubEnd If

    If Tabel.RecordCount <> 0 Then

        If Me.TxtPassword.Text = Tabel.Fields("password").Value ThenIf
            Me.TxtStatus.Text = "MNGR" Then

            FrmMenuUtama.Transaksi.Visible = False
            FrmMenuUtama.InfObat.Visible = False
            FrmMenuUtama.UbahHA.Visible = False

            End If

            If Me.TxtStatus.Text = "KASIR" Then
                FrmMenuUtama.InfObat.Visible = False
                FrmMenuUtama.lapor.Visible = False
                FrmMenuUtama.UbahHA.Visible = False

            End If Unload Me

            Load FrmMenuUtama
            FrmMenuUtama.Show

        Else

            MsgBox "Password tidak dikenal!", vbOKOnly + vbExclamation, "HAK AKSES DITOLAK"
            Me.TxtPassword.Text = ""

            Me.TxtPassword.SetFocusLoad Me

        End IfElse

        MsgBox "Password tidak dikenal!", vbOKOnly + vbExclamation, "HAK AKSES DITOLAK"Load Me

    End If
    Else

        PerintahSQL.CommandText = "Select * from login WHERE id_user = ? and STATUS = ? "
        PerintahSQL.Parameters(0).Value = Me.TxtIdUser.Text
```

```
PerintahSQL.Parameters(1).Value = Me.TxtStatus.TextSet
Tabel = PerintahSQL.Execute
```

```
    If Tabel.EOF = True Then
```

```
MsgBox "MAAF HAK AKSES TIDAK DIBERIKAN PADA ANDA!", vbOKOnly + vbExclamation, "HAK AKSES DITOLAK"
```

```
Unload MeEnd If If Tabel.RecordCount <> 0 Then
```

```
If Me.TxtPassword.Text = Tabel.Fields("password").Value ThenIf
Me.TxtStatus.Text = "ADMIN" Then
```

```
Load FrmDataLogin
```

```
FrmDataLogin.Show vbModal, FrmMenuUtamaUnload
Me
```

```
End IfElse
```

```
MsgBox "Password tidak dikenal!", vbOKOnly + vbExclamation, "HAK AKSES DITOLAK"
Me.TxtPassword.Text = ""
```

```
Me.TxtPassword.SetFocusUnload Me
```

```
End IfEnd If
```

```
End If End Sub Public K24 PHARMACY As New ADODB.Connection
```

```
Sub BukaDatabase()
```

```
App.HelpFile = App.Path & "\Help.HLP"zIf K24
PHARMACY.State = 1 Then K24 PHARMACY.Close
```

```
K24 PHARMACY.CursorLocation = adUseClient
```

```
K24
PHARM
ACY.Ope
n
"Provide
r=SQLNC
LI.1;Inte
grated
```

```
K24
PHARMACY.Ope
n
"Provider=SQLN
CLI.1;Integrated
Security=SSPI;P
ersist Security
Info=False;Initia
l Catalog=K24
PHARMACY;Dat
a
Source=andrie\
sqlexpress"
```

```
PerintahSQL.Acti  
veConnection =  
K24 PHARMACY  
End Sub
```

```
Sub  
TutupDatabase(  
)
```

```
If K24  
PHARMACY.Stat  
e = 1 Then K24  
PHARMACY.Clos  
e Set K24  
PHARMACY =  
Nothing
```

```
'Perintah untuk  
menghapus  
otomatis tmp  
On Error  
Resume Next
```

```
If Dir(App.Path  
& "\*.tmp") <>  
"" Then  
FileSystem.Kill  
App.Path &  
"\*.tmp"
```

```
End If
```

```
End End Sub
```

```
Private Sub  
TxtIdUser_DbICl  
ick()  
Me.LVKode.Visi  
ble = True
```

```
Call  
tekantombol5  
End Sub
```

```
Sub  
tekantombol5()
```

```
Dim Login As  
ADODB.Records  
et  
Me.LVKode.Visi  
ble = True
```

```
Me.LVKode.Top  
=  
Me.TxtIdUser.Top +  
(Me.TxtIdUser.  
Height / 2)  
Me.LVKode.Left  
=  
Me.TxtIdUser.L  
eft
```

```
With  
Me.LVKode
```

```
.ListItems.Clear
```

```
.ColumnHeader  
s.Clear
```

```
.ColumnHeader  
s.Add , , "ID  
User", 1250
```

```
.ColumnHeader  
s.Add , , "Nama  
User", 1750
```

```
POSISI = "USER"
```

```
PerintahSQL.Co  
mmandText =  
"Select id_user,  
nama_lkp from  
login" Set Login  
=  
PerintahSQL.Exe  
cute
```

```
If  
Login.RecordCo  
unt <> 0 Then  
Login.MoveFirst
```

```
Do Until  
Login.EOF
```

```
.ListItems.Add ,  
, Login!id_user
```

```
.ListItems(.ListIt  
ems.Count).List  
SubItems.Add , ,  
Login!nama_lkp
```

```
Login.MoveNex  
t Loop
```

End If

.SetFocus End
With End Sub

Private Sub
CmdSimpan_Click()
End Sub

If Me.Caption =
"TAMBAH DATA
OBAT" Then

PerintahSQL.Co
mmandText =
"INSERT INTO
Obat
(kode_obat,
id_user,
nama_obat,
kode_efek_t,
kode_gol, " & _
"kode_kemasan
, kode_satuan,
stk_brg,
harga_beli,
harga_jual,
satatus) VALUES
(?, ?, ?, ?, ?, ?, ?, ?,
, ?, ?)"
PerintahSQL.Par
ameters(0).Valu
e =
Me.TxtKodeOba
t.Text

PerintahSQL.Par
ameters(1).Valu
e =
Me.TxtIdUser.T
ext
PerintahSQL.Par
ameters(2).Valu
e =
Me.TxtNamaOb
at.Text
PerintahSQL.Par
ameters(3).Valu
e =
Me.TxtKdTerapi
.Text
PerintahSQL.Par
ameters(4).Valu
e =
Me.TxtKdGol.Te

```

xt
PerintahSQL.Parameters(5).Value =
Me.TxtKdKemas
.Text
PerintahSQL.Parameters(6).Value =
Me.TxtKdSat.Te
xt
PerintahSQL.Parameters(7).Value =
Me.TxtStok.Text
PerintahSQL.Parameters(8).Value =
Format(Me.Txt
HrgBel.Text,
"currency")
PerintahSQL.Parameters(9).Value =
Format(Me.Txt
HrgJual.Text,
"currency")
PerintahSQL.Parameters(10).Value =
Me.TxtStatus.Te
xt
PerintahSQL.Execute

Else

If Me.Caption =
"TAMBAH STOK
OBAT" Then
Dim x As Integer

Me.txtTambah.
SetFocus x =
Me.TxtStok

If
Me.txtTambah
= "" Then
txtTambah = 0

Me.TxtStok =
Me.txtTambah
+ x Else

```

```
Me.TxtStok =  
Me.txtTambah  
+ x End If
```

```
PerintahSQL.Co  
mmandText =  
"UPDATE Obat  
SET id_user = ?,  
nama_obat = ?,  
kode_efek_t =  
?,  
kode_kemasan  
=?,  
kode_satuan  
=?, kode_gol =  
?, " & _
```

```
"stk_brg = ?,  
harga_beli = ?,  
harga_jual = ?,  
satatus = ? " & _  
"where  
kode_obat =?"
```

```
PerintahSQL.Par  
ameters(0).Valu  
e =  
Me.TxtIdUser.T  
ext  
PerintahSQL.Par  
ameters(1).Valu  
e =  
Me.TxtNamaOb  
at.Text  
PerintahSQL.Par  
ameters(2).Valu  
e =  
Me.TxtKdTerapi  
.Text  
PerintahSQL.Par  
ameters(3).Valu  
e =  
Me.TxtKdKemas  
.Text  
PerintahSQL.Par  
ameters(4).Valu  
e =  
Me.TxtKdSat.Te  
xt  
PerintahSQL.Par  
ameters(5).Valu
```



```
e =  
Me.TxtKdGol.Text  
PerintahSQL.Parameters(6).Value  
e =  
Me.TxtStok.Text  
PerintahSQL.Parameters(7).Value  
e =  
Format(Me.Txt  
HrgBel.Text,  
"currency")  
PerintahSQL.Parameters(8).Value  
e =  
Format(Me.Txt  
HrgJual.Text,  
"currency")  
PerintahSQL.Parameters(9).Value  
e =  
Me.TxtStatus.Text  
PerintahSQL.Parameters(10).Value  
Me.TxtKodeObat.Text  
PerintahSQL.Execute
```

Else

```
If Me.Caption =  
"EDIT DATA  
OBAT" Then
```

```
PerintahSQL.CommandText =  
"UPDATE Obat  
SET id_user = ?,  
nama_obat = ?,  
kode_efek_t =  
?,  
kode_kemasan  
=?,  
kode_satuan  
=?, kode_gol =  
?, " & _
```

```
"stk_brg = ?,  
harga_beli = ?,  
harga_jual = ?,  
satatus = ? " & _
```

"where
kode_obat =?"

PerintahSQL.Parameters(0).Value =
Me.TxtIdUser.Text
PerintahSQL.Parameters(1).Value =
Me.TxtNamaObat.Text
PerintahSQL.Parameters(2).Value =
Me.TxtKdTerapi.Text
PerintahSQL.Parameters(3).Value =
Me.TxtKdKemas.Text
PerintahSQL.Parameters(4).Value =
Me.TxtKdSat.Text
PerintahSQL.Parameters(5).Value =
Me.TxtKdGol.Text
PerintahSQL.Parameters(6).Value =
Me.TxtStok.Text
PerintahSQL.Parameters(7).Value =
Format(Me.TxtHrgBel.Text,
"currency")
PerintahSQL.Parameters(8).Value =
Format(Me.TxtHrgJual.Text,
"currency")
PerintahSQL.Parameters(9).Value =
Me.TxtStatus.Text
PerintahSQL.Par

```
ameters(10).Value =  
Me.TxtKodeObat.Text  
PerintahSQL.Execute
```

```
End If  
End If  
End If
```

```
Unload Me
```

```
FrmMenuUtama.StatusBar1.Panels(1) = "Data  
Obat" End Sub
```

```
Private Sub  
Form_Activate()
```

```
If Me.Caption =  
"DATA OBAT  
MINIMUM"  
Then
```

```
PerintahSQL.CommandText =  
"SELECT  
kode_obat as  
[Kode Obat],  
nama_obat as  
[Nama Obat],  
stk_brg as [Stok  
Barang] FROM  
Obat where  
stk_brg <= 1  
order by  
nama_obat asc"
```

```
Set Tabel =  
PerintahSQL.Execute
```

```
Set  
Me.TDBGrid1.DataSource =  
Tabel
```

```
Dim i As Integer
```

```
For i = 0 To  
Me.TDBGrid1.Columns.Count -  
1
```

```
Me.TDBGrid1.Columns(i).AutoSize  
Me.TDBGrid1.Columns(i).Alignment =  
dbCenter  
Me.TDBGrid1.Columns(i).Header  
Alignment =  
dbCenter
```

```
Next i
```

```
Me.LblJumlah.Caption =  
Tabel.RecordCount
```

```
If Me.Caption =  
"DATA OBAT  
MINIMUM"  
Then  
Me.HelpContextID = 22 Else
```

```
PerintahSQL.CommandText =  
"SELECT  
DISTINCT  
Detail_Penj.kode_obat AS  
[Kode Obat],  
Obat.nama_obat as [Nama  
Obat], " & _
```

```
"SUM(Detail_Penj.jumlah_brg)  
As [Total  
Penjualan]  
FROM  
Detail_Penj  
INNER JOIN  
Obat ON " & _  
"Detail_Penj.kode_obat =  
Obat.kode_obat  
INNER JOIN  
Penj ON " & _
```

```
"Detail_Penj.no_notas =  
Penj.no_notas  
GROUP BY
```

```
Detail_Penj.kod  
e_obat,  
Obat.nama_obat  
ORDER BY  
[Total  
Penjualan]  
DESC"
```

```
Set Tabel =  
PerintahSQL.Exe  
cute
```

```
Set  
Me.TDBGrid1.D  
ataSource =  
Tabel
```

```
For i = 0 To  
Me.TDBGrid1.C  
olumns.Count -  
1  
Me.TDBGrid1.C  
olumns(i).AutoS  
ize  
Me.TDBGrid1.C  
olumns(i).Align  
ment =  
dbgCenter  
Me.TDBGrid1.C  
olumns(i).Head  
Alignment =  
dbgCenter
```

```
Next i
```

```
Me.LblJumlah.C  
aption =  
Tabel.RecordCo  
unt  
Me.Label3.Capti  
on = "jenis obat  
yang  
dirangking"
```

```
If Me.Caption =  
"RANGKING  
PENJUALAN  
OBAT" Then  
Me.HelpContex  
tID = 25 End If
```

```
Private Sub  
EfTerap_Click()  
Load FrmIdObat
```

```
FrmIdObat.Caption = "DATA  
EFEK TERAPI"  
FrmIdObat.Label1.Caption =  
"Kode Efek  
Terapi"  
FrmIdObat.Label2.Caption =  
"Nama Efek  
Terapi"  
Me.StatusBar1.  
Panels(1).Text =  
"Data Efek  
Terapi"
```

```
PerintahSQL.CommandText =  
"Select  
kode_efek_t as  
[Kd Efek Trp],  
efek_terapi as  
[Nama Efek  
Terapi] from  
Efek_Trp" Set  
Tabel =  
PerintahSQL.Execute
```

```
Set  
FrmIdObat.TDB  
Grid1.DataSource = Tabel
```

```
Dim i As Integer
```

```
For i = 0 To  
FrmIdObat.TDB  
Grid1.Columns.  
Count - 1  
FrmIdObat.TDB  
Grid1.Columns(i  
)AutoSize  
FrmIdObat.TDB  
Grid1.Columns(i  
)Alignment =  
dbgCenter  
FrmIdObat.TDB  
Grid1.Columns(i  
)HeadAlignmen  
t = dbgCenter
```

Next i

```
FrmlIdObat.TDB
Grid1.Columns(
1).Alignment =
dbgLeft
FrmlIdObat.Sho
w vbModal,
FrmMenuUtam
a
```

End Sub

```
Private Sub
GolObat_Click()
Load FrmlIdObat
```

```
FrmlIdObat.Capt
ion = "DATA
GOLONGAN
OBAT"
FrmlIdObat.Labe
l1.Caption =
"Kode Gol.
Obat"
FrmlIdObat.Labe
l2.Caption =
>Nama Gol
Obat"
Me.StatusBar1.
Panels(1).Text =
>Data Golongan
Obat"
```

```
PerintahSQL.Co
mmandText =
"Select
kode_gol as
[Kode
Golongan],
nama_gol as
>Nama
Golongan] from
Golongan" Set
Tabel =
PerintahSQL.Exe
cute
```

```
Set
FrmlIdObat.TDB
Grid1.DataSour
ce = Tabel
```

Dim i As Integer

```
For i = 0 To  
FrmIdObat.TDB  
Grid1.Columns.  
Count - 1  
FrmIdObat.TDB  
Grid1.Columns(i  
)AutoSize  
FrmIdObat.TDB  
Grid1.Columns(i  
)Alignment =  
dbgCenter  
FrmIdObat.TDB  
Grid1.Columns(i  
)HeadAlignmen  
t = dbgCenter
```

```
Next i
```

```
FrmIdObat.TDB  
Grid1.Columns(  
1).Alignment =  
dbgLeft  
FrmIdObat.Sho  
w vbModal,  
FrmMenuUtam  
a
```

```
End Sub
```

```
Private Sub  
GoObat_Click()  
Load FrmIdObat
```

```
FrmIdObat.Capt  
ion = "DATA  
GOLONGAN  
OBAT"  
FrmIdObat.Labe  
l1.Caption =  
"Kode Gol.  
Obat"  
FrmIdObat.Labe  
l2.Caption =  
"Nama Gol  
Obat"  
Me.StatusBar1.  
Panels(1).Text =  
"Data Golongan  
Obat"
```

```
PerintahSQL.Co  
mmandText =  
"Select  
kode_gol as
```



```
[Kode  
Golongan],  
nama_gol as  
[Nama  
Golongan] from  
Golongan" Set  
Tabel =  
PerintahSQL.Exe  
cute
```

```
Set  
FrmIdObat.TDB  
Grid1.DataSour  
ce = Tabel
```

```
Dim i As Integer
```

```
For i = 0 To  
FrmIdObat.TDB  
Grid1.Columns.  
Count - 1  
FrmIdObat.TDB  
Grid1.Columns(i  
)AutoSize  
FrmIdObat.TDB  
Grid1.Columns(i  
)Alignment =  
dbgCenter  
FrmIdObat.TDB  
Grid1.Columns(i  
)HeadAlignmen  
t = dbgCenter
```

```
Next i
```

```
FrmIdObat.TDB  
Grid1.Columns(  
1).Alignment =  
dbgLeft  
FrmIdObat.Sho  
w vbModal,  
FrmMenuUtam  
a
```

```
End Sub
```

```
Private Sub GoObat_Click() Load FrmIdObat
```

```
FrmIdObat.Caption = "DATA GOLONGAN OBAT" FrmIdObat.Label1.Caption = "Kode Gol. Obat" FrmIdObat.Label2.Caption = "Nama Gol  
Obat" Me.StatusBar1.Panels(1).Text = "Data Golongan Obat"
```

```
PerintahSQL.CommandText = "Select kode_gol as [Kode Golongan], nama_gol as [Nama Golongan] from Golongan" Set Tabel =  
PerintahSQL.Execute
```

```

Set FrmIdObat.TDBGrid1.DataSource = Tabel

Dim i As Integer

For i = 0 To FrmIdObat.TDBGrid1.Columns.Count - 1 FrmIdObat.TDBGrid1.Columns(i).AutoSize FrmIdObat.TDBGrid1.Columns(i).Alignment
= dbgCenter FrmIdObat.TDBGrid1.Columns(i).HeadAlignment = dbgCenter

Next i

FrmIdObat.TDBGrid1.Columns(1).Alignment = dbgLeft FrmIdObat.Show vbModal, FrmMenuUtama

End Sub

Private Sub KemObat_Click() Load FrmIdObat

FrmIdObat.Caption = "DATA KEMASAN OBAT" FrmIdObat.Label1.Caption = "Kode Kemasan" FrmIdObat.Label2.Caption = "Nama Kemasan"
Me.StatusBar1.Panels(1).Text = "Data Kemasan Obat"

PerintahSQL.CommandText = "Select kode_kemasan as [Kode Kemasan], nama_kemasan as [Nama Kemasan] from Kemasan"

Set Tabel = PerintahSQL.Execute

Set FrmIdObat.TDBGrid1.DataSource = Tabel

Dim i As Integer

For i = 0 To FrmIdObat.TDBGrid1.Columns.Count - 1 FrmIdObat.TDBGrid1.Columns(i).AutoSize FrmIdObat.TDBGrid1.Columns(i).Alignment
= dbgCenter FrmIdObat.TDBGrid1.Columns(i).HeadAlignment = dbgCenter

Next i

FrmIdObat.TDBGrid1.Columns(1).Alignment = dbgLeft FrmIdObat.Show vbModal, FrmMenuUtama

Private Sub SatObat_Click() Load FrmIdObat

FrmIdObat.Caption = "DATA SATUAN OBAT" FrmIdObat.Label1.Caption = "Kode Satuan" FrmIdObat.Label2.Caption = "Nama Satuan"
Me.StatusBar1.Panels(1).Text = "Data Satuan Obat"

PerintahSQL.CommandText = "Select kode_satuan as [Kode Satuan], nama_satuan as [Nama Satuan] from Satuan" Set Tabel =
PerintahSQL.Execute

Set FrmIdObat.TDBGrid1.DataSource = Tabel

Dim i As Integer

For i = 0 To FrmIdObat.TDBGrid1.Columns.Count - 1 FrmIdObat.TDBGrid1.Columns(i).AutoSize FrmIdObat.TDBGrid1.Columns(i).Alignment
= dbgCenter FrmIdObat.TDBGrid1.Columns(i).HeadAlignment = dbgCenter

Next i

FrmIdObat.TDBGrid1.Columns(1).Alignment = dbgLeft FrmIdObat.Show vbModal, FrmMenuUtama

End SubEnd Sub

Private Sub CmdSimpan_Click()

If Me.Caption = "DATA EFEK TERAPI" Then

PerintahSQL.CommandText = "INSERT INTO Efek_Trp (kode_efek_t, efek_terapi) VALUES (?,?)" PerintahSQL.Parameters(0).Value =
Me.Text1.Text

PerintahSQL.Parameters(1).Value = Me.Text2.Text PerintahSQL.Execute

MsgBox "Data Sudah Tersimpan", vbOKOnly + vbInformation, "Informasi"

PerintahSQL.CommandText = "Select kode_efek_t as [Kd Efek Trp], efek_terapi as [Nama Efek Terapi] from Efek_Trp" Set Tabel =
PerintahSQL.Execute

Set FrmIdObat.TDBGrid1.DataSource = Tabel Dim i As Integer

```

```
For i = 0 To FrmIdObat.TDBGrid1.Columns.Count - 1 FrmIdObat.TDBGrid1.Columns(i).AutoSize FrmIdObat.TDBGrid1.Columns(i).Alignment = dbgCenter FrmIdObat.TDBGrid1.Columns(i).HeadAlignment = dbgCenter
```

```
Next i
```

```
FrmIdObat.TDBGrid1.Columns(1).Alignment = dbgLeft Call kosongkan
```

```
End If
```

```
If Me.Caption = "DATA GOLONGAN OBAT" Then
```

```
PerintahSQL.CommandText = "INSERT INTO Golongan (kode_gol, nama_gol) VALUES (?,?)" PerintahSQL.Parameters(0).Value = Me.Text1.Text
```

```
PerintahSQL.Parameters(1).Value = Me.Text2.Text PerintahSQL.Execute
```

```
MsgBox "Data Sudah Tersimpan", vbOKOnly + vbInformation, "Informasi"
```

```
PerintahSQL.CommandText = "Select kode_gol as [Kode Golongan], nama_gol as [Nama Golongan] from Golongan" Set Tabel = PerintahSQL.Execute
```

```
Set FrmIdObat.TDBGrid1.DataSource = Tabel
```

```
For i = 0 To FrmIdObat.TDBGrid1.Columns.Count - 1 FrmIdObat.TDBGrid1.Columns(i).AutoSize FrmIdObat.TDBGrid1.Columns(i).Alignment = dbgCenter FrmIdObat.TDBGrid1.Columns(i).HeadAlignment = dbgCenter
```

```
Next i
```

```
FrmIdObat.TDBGrid1.Columns(1).Alignment = dbgLeft Call kosongkan
```

```
End If
```

```
If Me.Caption = "DATA KEMASAN OBAT" Then
```

```
PerintahSQL.CommandText = "INSERT INTO Kemasan (kode_Kemasan, nama_kemasan) VALUES (?,?)" PerintahSQL.Parameters(0).Value = Me.Text1.Text
```

```
PerintahSQL.Parameters(1).Value = Me.Text2.Text PerintahSQL.Execute
```

```
MsgBox "Data Sudah Tersimpan", vbOKOnly + vbInformation, "Informasi"
```

```
PerintahSQL.CommandText = "Select kode_kemasan as [Kode Kemasan], nama_Kemasan as [Nama Kemasan] from Kemasan"
```

```
Set Tabel = PerintahSQL.Execute
```

```
Set FrmIdObat.TDBGrid1.DataSource = Tabel
```

```
For i = 0 To FrmIdObat.TDBGrid1.Columns.Count - 1 FrmIdObat.TDBGrid1.Columns(i).AutoSize FrmIdObat.TDBGrid1.Columns(i).Alignment = dbgCenter FrmIdObat.TDBGrid1.Columns(i).HeadAlignment = dbgCenter
```

```
Next i
```

```
FrmIdObat.TDBGrid1.Columns(1).Alignment = dbgLeft Call kosongkan
```

```
End If
```

```
If Me.Caption = "DATA SATUAN OBAT" Then
```

```
PerintahSQL.CommandText = "INSERT INTO Satuan (kode_Satuan, nama_satuan) VALUES (?,?)" PerintahSQL.Parameters(0).Value = Me.Text1.Text
```

```
PerintahSQL.Parameters(1).Value = Me.Text2.Text PerintahSQL.Execute
```

MsgBox "Data Sudah Tersimpan", vbOKOnly + vbInformation, "Informasi"

PerintahSQL.CommandText = "Select kode_satuan as [Kode Satuan], nama_satuan as [Nama Satuan] from Satuan" Set Tabel = PerintahSQL.Execute

Set FrmIdObat.TDBGrid1.DataSource = Tabel

For i = 0 To FrmIdObat.TDBGrid1.Columns.Count - 1 FrmIdObat.TDBGrid1.Columns(i).AutoSize FrmIdObat.TDBGrid1.Columns(i).Alignment = dbgCenter FrmIdObat.TDBGrid1.Columns(i).HeadAlignment = dbgCenter

Next i

FrmIdObat.TDBGrid1.Columns(1).Alignment = dbgLeft Call kosongkan

End If End Sub

Private Sub CmdCari_Click()

Dim TblCari As ADODB.Recordset If Me.TxtCari.Text = "" Then

MsgBox "Masukkan kata kunci yang akan dicari!!!", vbOKOnly + vbInformation, "Konfirmasi" Me.TxtCari.SetFocus

Exit Sub End If

If OptNamaObat.Value = True Then

PerintahSQL.CommandText = "SELECT nama_obat as [Nama Obat], kode_obat as [Kode Obat], stk_brg as [Stok Barang], harga_jual as [Harga Jual] " & _

"FROM Obat where nama_obat LIKE '%" & Me.TxtCari.Text & "%' order by nama_obat asc" Set TblCari = PerintahSQL.Execute

Set Me.TDBGrid1.DataSource = TblCari Me.TDBGrid1.Columns(3).NumberFormat = "Currency"

End If

If OptKemasan.Value = True Then

PerintahSQL.CommandText = "SELECT Kemasan.nama_kemasan as [Nama Kemasan], Kemasan.kode_kemasan as [Kode Kemasan], " & _
"nama_obat as [Nama Obat], kode_obat as [Kode Obat], stk_brg as [Stok], harga_jual as [Harga Jual] " & _

"FROM Obat INNER JOIN Kemasan ON Obat.kode_kemasan = Kemasan.kode_kemasan where kemasan.nama_kemasan LIKE '%" & Me.TxtCari.Text & "%' " & _

"order by kemasan.nama_kemasan asc" Set TblCari = PerintahSQL.Execute

Set Me.TDBGrid1.DataSource = TblCari Me.TDBGrid1.Columns(5).NumberFormat = "Currency"

End If

If OptSatuan.Value = True Then

PerintahSQL.CommandText = "SELECT Satuan.nama_satuan as [Nama Satuan], Satuan.kode_satuan as [Kode Satuan], " & _

"Obat.nama_obat as [Nama Obat], Obat.kode_obat as [Kode Obat], Obat.stk_brg as [Stok Barang], Obat.harga_jual as [Harga Jual] " & _
"FROM Obat INNER JOIN Satuan ON Obat.kode_satuan = Satuan.kode_satuan " & _

"where satuan.nama_satuan LIKE '%" & Me.TxtCari.Text & "%' order by satuan.nama_satuan asc" Set TblCari = PerintahSQL.Execute

Set Me.TDBGrid1.DataSource = TblCari Me.TDBGrid1.Columns(5).NumberFormat = "Currency"

End If

If OptEfek.Value = True Then

```
PerintahSQL.CommandText = "SELECT Efek_Trp.efek_terapi as [Nama Efek], Efek_Trp.kode_efek_t as [Kode Efek], " & _ "Obat.nama_obat  
as [Nama Obat], Obat.kode_obat as [Kode Obat], Obat.stk_brg as [Stok], Obat.harga_jual as [Harga Jual] " & _ "FROM Obat INNER JOIN  
Efek_Trp ON Obat.kode_efek_t = Efek_Trp.kode_efek_t " & _
```

"where Efek_Trp.efek_terapi LIKE '%" & Me.TxtCari.Text & "%' order by efek_trp.efek_terapi asc" Set TblCari = PerintahSQL.Execute

Set Me.TDBGrid1.DataSource = TblCari Me.TDBGrid1.Columns(5).NumberFormat = "Currency"

End If

If OptGolongan.Value = True Then

```
PerintahSQL.CommandText = "SELECT Golongan.nama_gol as [Nama Golongan], Golongan.kode_gol as [Kode Golongan],
```

```
Obat.nama_obat as [Nama Obat], " & _
```

```
"Obat.kode_obat as [Kode Obat] , Obat.stk_brg as [Stok], Obat.harga_jual as [Harga Jual] FROM Obat INNER JOIN " & _ "Golongan ON  
Obat.kode_gol = Golongan.kode_gol " & _
```

"where Golongan.nama_gol LIKE '%" & Me.TxtCari.Text & "%' order by golongan.nama_gol asc" Set TblCari = PerintahSQL.Execute

Set Me.TDBGrid1.DataSource = TblCari Me.TDBGrid1.Columns(5).NumberFormat = "Currency"

End If

Private Sub CmdSimpan_Click()

If Me.CmdSimpan.Caption = "&SIMPAN" Then

```
PerintahSQL.CommandText = "INSERT INTO LOGIN (id_user, nama_lkp, nama_user, status, password) " & _ "VALUES (?, ?, ?, ?, ?)"
```

```
PerintahSQL.Parameters(0).Value = Me.TxtIdUser.Text
```

If Me.TxtIdUser.Text = Me.ListView1.ListItems.Item(1).Text Then

MsgBox "DATA SUDAH TERSIMPAN", vbOKOnly + vbInformation, "INFORMASI" Else

```
PerintahSQL.Parameters(1).Value = Me.TxtNamaLengkap.Text PerintahSQL.Parameters(2).Value = Me.TxtNamaUser.Text
```

```
PerintahSQL.Parameters(3).Value = Me.TxtStatus.Text PerintahSQL.Parameters(4).Value = Me.TxtPassword.Text PerintahSQL.Execute
```

MsgBox "DATA SUDAH TERSIMPAN", vbOKOnly + vbInformation, "INFORMASI" End If

Call Tampil Call kosongkan

Else

```
PerintahSQL.CommandText = "UPDATE LOGIN SET nama_lkp = ?, nama_user = ?, status = ?, password = ? " & _ "where id_user = ?"
```

```
PerintahSQL.Parameters(0).Value = Me.TxtNamaLengkap.Text PerintahSQL.Parameters(1).Value = Me.TxtNamaUser.Text
```

```
PerintahSQL.Parameters(2).Value = Me.TxtStatus.Text PerintahSQL.Parameters(3).Value = Me.TxtPassword.Text
```

```
PerintahSQL.Parameters(4).Value = Me.TxtIdUser.Text PerintahSQL.Execute
```

MsgBox "DATA SUDAH TER-UPDATE", vbOKOnly + vbInformation, "INFORMASI" Call Tampil

Unload Me End If

End Sub

Private Sub CmdHapus_Click() On Error GoTo Pesan

Dim vHapus As String

If MsgBox("Apakah yakin akan menghapus data LOGIN ini?", vbYesNo + vbQuestion, "PERHATIAN") = vbYes Then vHapus =

Me.ListView1.SelectedItem.Text

```
PerintahSQL.CommandText = "delete from login where id_user = ?" PerintahSQL.Parameters(0).Value = vHapus
```

```
PerintahSQL.Execute End If
```

Call kosongkan Call Tampil Exit