BAB II

TINJAUAN PUSTAKA

2.1 Gambaran umum Apotek Dua Saudara

2.1.1 Sejarah Apotek Dua Saudara

Apotek Dua Saudara Bandung merupakan salah satu dari sarana kesehatan, yang bertujuan untuk menciptakan daerah yang cinta akan kesehatan dan mewujudkan derajat kesehatan yang optimal bagi masyarakat bandung. Apotek Dua Saudara pertama kali didirikan oleh Ir.Tenny Widjaja pada tanggal 25 Agustus 2002,yang berlokasi di Jl. Cihampelas No. 260 Bandung 40132. Melihat perkembangan yang telah dialaminya, apotek dua saudara berkembang dengan cepat dan memiliki pelanggan yang cukup banyak, sehingga apotek berinisiatif untuk pindah tempat dan membangun instansi sendiri, dengan membeli tanah dan mendirikan bangunan tersebut, setelah selesai dibangun Apotek Dua Saudara pindah pada tanggal 5 Agustus 2007, di Jl. Cihampelas No. 256 Bandung 40132, hingga sampai sekarang apotek itu berjalan.

2.1.2 Visi dan Misi Apotek Dua Saudara

2.1.2.1 Visi

Menjadikan Apotek yang berkualitas dibidang kesehatan dan terjangkau oleh masyarakat kota Bandung.

2.1.2.2 Misi

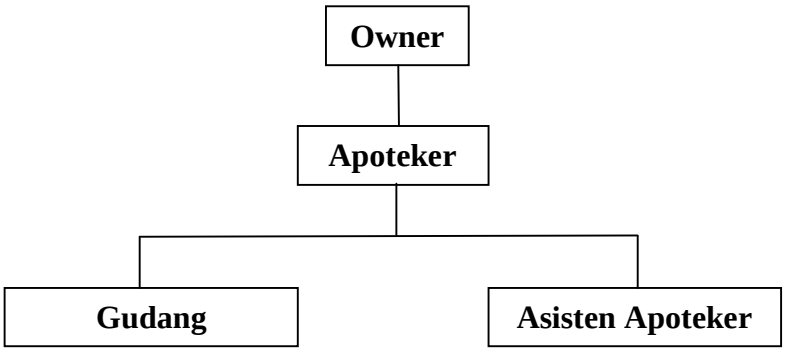
1. Menyelenggarakan pelayanan kesehatan yang baik pada pelanggan
2. Meningkatkan kualitas dan kuantitas sumber daya manusia
3. Mengupayakan perlindungan hokum bagi sumber daya manusia
4. Menciftakan dan mengembangkan lingkunagn yang sehat
5. Meningkatkan program-program yang menunjang Apotek

2.1.3 Tempat dan Kedudukan Apotek Dua Saudara

Jl. Cihampelas No. 256 Telp. (022) 22032810 Bandung 40132

2.1.4 Struktur Organisasi

Struktur organisasi merupakan susunan yang terdiri dari fungsi-fungsi dan hubungan-hubungan yang menyatakan keseluruhan kegiatan untuk mencapai suatu sasaran. Struktur keorganisasian pada Apotek Dua Saudara Bandung dapat dilihat pada Gambar 2.1 Struktur organisasi Apotek Dua Saudara Bandung.



Gambar 2.1 Struktur organisasi Apotek Dua Saudara Bandung

Keterangan :

1. *Owner*

   Orang yang memiliki perusahaan tersebut, *owner* bertugas mengontrol staf/Asisten apoteker dalam bekerja dan sekaligus memegang keuangan didalam penggajian karyawan. *Owner* juga bertugas menjadi admin untuk mengetahui semua data obat, mengolah data user,mengolah transaksi baik penjualan atau pembelian obat dan pembuatan laporan.

2. *Apoterker*

   Apoteker adalah orang yang bertanggung jawab diapotek itu, yang mengontrol penjualan obat, dan pengecekan obat.

3. *Asisten Apoteker (AA)*

   Asisten apoteker (AA) atau staf yang bertugas melayani pembelian obat, meracik obat, melakukan transaksi dan membuat laporan.

4. *Gudang*

   Gudang adalah suatu ruangan yang hanya untuk penyimpanan obat saja.

2.1.5 Sistem Informasi

2.2.1 Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan. Sistem informasi dalam suatu

organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja diperlukan. Sistem ini menyimpan, mengambil, mengubah, mengolah dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya.

2.2.2 Elemen Sistem Informasi

Sistem informasi terdiri dari elemenelemen yang terdiri dari orang, prosedur, perangkat keras, perangkat lunak, basis data, jaringan komputer dan komunikasi data. Semua elemen ini merupakan komponen fisik.

1. *Orang*

Orang atau personil yang di maksudkan yaitu operator komputer, analis sistem, programmer, personil data entry, dan manajer sistem informasi/EDP

2. *Prosedur*

Prosedur merupakan elemen fisik. Hal ini di sebabkan karena prosedur disediakan dalam bentuk fisik seperti buku panduan dan instruksi. Ada 3 jenis prosedur yang dibutuhkan, yaitu instruksi untuk pemakai, instruksi untuk penyiapan masukan, instruksi pengoperasian untuk karyawan pusat komputer.

3. *Perangkat keras*

Perangkat keras bagi suatu sistem informasi terdiri atas komputer (pusat pengolah, unit masukan/keluaran), peralatan penyiapan data, dan terminal masukan/keluaran.

4. *Perangkat lunak*

Perangkat lunak dapat dibagi dalam 3 jenis utama :

a. Sistem perangkat lunak umum, seperti sistem pengoperasian dan sistem manajemen data yang memungkinkan pengoperasian sistem komputer.
b. Aplikasi perangkat lunak umum, seperti model analisis dan keputusan.
c. Aplikasi pernagkat lunak yang terdiri atas program yang secara spesifik dibuat untuk setiap aplikasi.

5. Basis data

File yang berisi program dan data dibuktikan dengan adanya media penyimpanan secara fisik seperti *diskatte*, *harddisk*, *magnetic tape*, dan sebagainya. File juga meliputi keluaran tercetak dan catatan lain diatas kertas, mikro film, an lain sebagainya.

6. Jaringan komputer

Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung dalam satu kesatuan. Informasi dan data bergerak melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data.

7. Komunikasi data

Komunikasi data adalah merupakan bagian dari telekomunikasi yang secara khusus berkenaan dengan transmisi atau pemindahan data dan informasi diantara komputerkomputer dan pirantipiranti yang lain dalam bentuk digital yang dikirimkan melalui media komunikasi data. Data berarti informasi yang disajikan oleh isyarat digital. Komunikasi data merupakan bagian vital dari suatu sistem

informasi karena sistem ini menyediakan infrastruktur yang memungkinkan komputerkomputer dapat berkomunikasi satu sama lain.

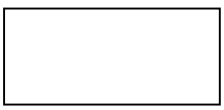
2.3 *Tools* Pemodelan

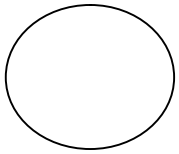
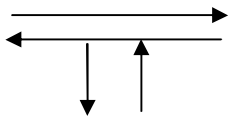
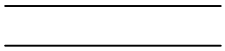
2.3.1 *Data Flow Diagram* (DFD)

Data Flow Diagram (DFD) adalah alat pembuatan model yang profesional sistem untuk menggambarkan system sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi.

DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem. DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

Tabel 2.1 Simbol Diagram Arus Data (*Data Flow Diagram*)

NO	SIMBOL	NAMA	KETERANGAN
1		*External Entity* (kesatuan luar)	Menggambarkan kegiatan proses dari operasi program komputer

2		*Process* (proses)	Menggambarkan kegiatan atau kerja yang dilakukan oleh manusia atau komputer
3		*Data Flow* (arus data)	Menunjukan arus data yang mengalir antara proses, simpanan data dan kesatuan luar dari ke suatu arus
4		*Data Store* (simpanan luar)	Menggambarkan suatu tempat penyimpanan data

Beberapa simbol yang digunakan dalam *Data Flow Diagram* (DFD) antara lain:

1. *External Entity* (kesatuan luar) atau *boundary* (batas sistem)

Setiap sistem pasti mempunyai batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima input dan menghasilkan output kepada lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem.

2. *Process* (proses)

Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang , mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. Untuk *physical data flow diagram* (PDFD), proses dapat dilakukan oleh orang, mesin atau komputer,

sedangkan untuk *logical data flow diagram* (LDFD), suatu proses hanya menunjukkan proses dari komputer. Setiap proses harus diberi penjelasan yang lengkap meliputu identifikasi proses, nama proses dam pemroses.

3. *Data Flow* (arus data)

Arus data (*data flow*) di DFD diberi simbol suatu panah. Arus data ini mengalir diantara proses (*process*), simpanan data (*data strore*) dan kesatuan luar (*external entity*). Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

4. *Data Store* (simpanan luar)

Simpanan data (*data store*) merupakan simpanan dari data yang dapat berupa, yaitu suatu file atau *database* di sistem komputer, suatu arsip atau catatan manual, suatu kotak tempat data di meja seseorang, suatu tabel acuan manual, dan suatu agenda atau buku.

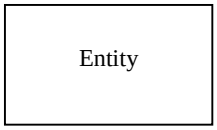
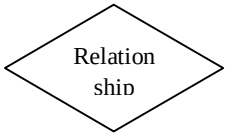
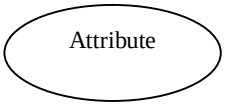
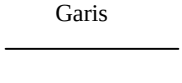
2.3.2 Diagram E-R (*Entity Relationship Diagram*)

Menurut Budi Sutedjo, ERD adalah peralatan yang berfungsi untuk menggambarkan relasi atau hubungan dari dua file atau dua tabel yang dapat digolongkan dalam tiga macam bentuk relasi, yaitu satu – satu (1 – 1),
satu – banyak (1 – N), banyak – banyak (N – N). Model entity relationship yang berisi komponen – komponen himpunan entitas dan himpunan relasi yang masing–masing dilengkapi dengan atribut–atribut yang merepresentasikan seluruh fakta yang kita tinjau digambarkan lebih sistematis dengan menggunakan Diagram E-R. Pembuatan Diagram E-R meliputi tahap-tahap berikut:

a. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlihat.
b. Menentukan atribut – atribut *key* dari masing – masing himpunan entitas.
c. Mengidentifikasi dan menetapkan seluruh himpunan relasi di antara himpunan entitas–himpunan entitas yang ada beserta *foreign key* yang terdapat pada relasi tersebut.
d. Menentukan derajat atau kardinalitas relasi untuk setiap himpunan relasi.

Melengkapi himpunan entitas dan himpunan relasi dengan atribut – atribut deskriptif (non *key*).

Tabel 2.2 Simbol *Entity Relationship Diagram* (ERD)

NO	SIMBOL	KETERANGAN
1		Menunjukan suatu objek berupa benda, konsep atau entitas, Entity adalah file yang berjenis master .
2		Menunjukan hubungan *logical* antara satu *file* dengan *file* lainnya. *Relationship* adalah file yang berjenis transaksi, digunakan untuk menyimpan data transaksi yang terjadi
3		Menunjukan atribut suatu entitas, Atribut adalah bagian-bagian spesifik dari file (entitas)
4		Digunakan sebagai penghubung antara relasi dengan entitas, entitas dengan atribut dan relasi dengan atribut

1. Entitas (Entity)

   Entitas adalah suatu yang nyata atau abstrak dimana kita akan menyimpan data. Contoh entitas pegawai, entitas kampus, entitas buku dsb.

2. Relasi (*Relationship*)

   Relasi adalah hubungan alamiah yang terjadi antara satu atau lebih entitas, misal proses pembayaran pegawai. Kardinalitas menentukan kejadian suatu entitas untuk satu kejadian pada entitas yang berhubungan. Misal, mahasiswa bisa mengambil banyak mata kuliah

3. Atribut (*Attribute*)

   Atribut adalah ciri umum semua atau sebagian besar instansi pada entitas tertentu. Sebutan lain atribut adalah properti, elemen data dan field. Misal, nama, alamat, nomor pegawai, gaji dsb.

2.3.3 Kamus Data (*Data Dictionary*)

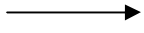
Kamus data adalah peralatan yang ikut berperan dalam perancangan dan pembangunan sistem informasi karena berfungsi untuk menjelaskan arti aliran data dan penyimpanan dalam penggambaran pada data *flow diagram*, mendeskripsikan komposisi paket data yang bergerak melalui aliran, dan menjelaskan spesifikasi nilai dan satuan yang relevan terhadap data yang mengalir dalam sistem tersebut.

2.3.4 Diagram Konteks

Diagram konteks adalah suatu diagram alir yang tingkat tinggi yang menggambarkan seluruh jaringan, *input* dan *output*. sistem yang dimaksud adalah untuk menggambarkan sistem yang sedang berjalan. mengidentifikasikan awal dan akhir data awal dan akhir yang masuk dan keluaran *system*, menggambarkan hubungan antara sistem dengan entitas luarnya. Diagram konteks berfungsi sebagai transformasi dari satu proses yang melakukan transformasi data *input* menjadi data *output*. Entitas yang dimaksud adalah entitas yang mempunyai hubungan langsung dengan sistem.Diagram konteks selalu mengandung satu dan hanya satu proses saja. Proses ini mewakili proses dari seluruh system.

Dibawah ini adalah symbol-simbol yang akan Digunakan di Diagram Konteks

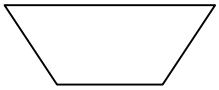
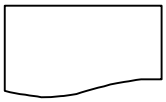
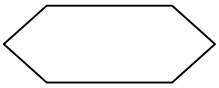
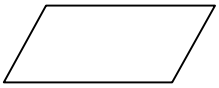
Tabel 2.3 Simbol Diagram Konteks

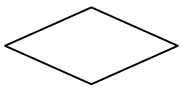
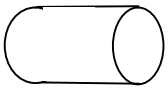
NO	SIMBOL	KETERANGAN
1	Terminator	Pihak-pihak yang berada di luar sistem, tetapi secara langsung berhubungan dengan sistem dalam hal memberi data atau menerima informasi
2	Process	Di dalam diagram konteks, berisi mengenai sistem yang akan dibuat.
3	Data Flow 	Berisi data atau informasi yang mengalir dari satu pihak ke sistem dan sebaliknya.

2.3.5 Flowmap

Flowmap adalah campuran peta dan *flow chart*, yang menunjukkan pergerakan benda dari satu lokasi ke lokasi lain, seperti jumlah orang dalam migrasi, jumlah barang yang diperdagangkan, atau jumlah paket dalam jaringan. Flowmap menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

Tabel 2.4 Simbol Flowmap

SIMBOL	NAMA	FUNGSI
	GARIS ALIR (FLOW LINE)	Arah aliran program
	MANUAL OPERATION	Operasi Manual
	DOKUMEN	I/O dalam format yang dicetak
	PREPARATION	Proses inisialisasi/pemberian harga awal
	PROSES	Proses perhitungan/proses pengolahan data
	INPUT/OUTPUT DATA	Proses input/output data, parameter, informasi
	PREDEFINED PROCESS (SUB PROGRAM)	Permulaan sub program/proses menjalankan sub program

	KEPUTUSAN	Keputusan dalam program
	PENGHUBUNG	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	PENYIMPANAN DATA	Penyimpanan yang dapat diakses langsung

Secara umum, struktur suatu program terdiri dari beberapa bagian yaitu :

1. *Input*

Bagian ini merupakan proses untuk memasukkan data ke komputer melalui device yang ada misalnya *keyboard, mouse*, *scanner* dll. Program melakukan proses membaca data yang akan diolah dari *device* tersebut.

2. *Output*

Bagian ini merupakan proses untuk menampilkan data yang telah diolah, melaporkan hasil pegolahan data melalui device seperti monitor, *printer* dll. Program melakukan proses mencetak data ke *device* tersebut.

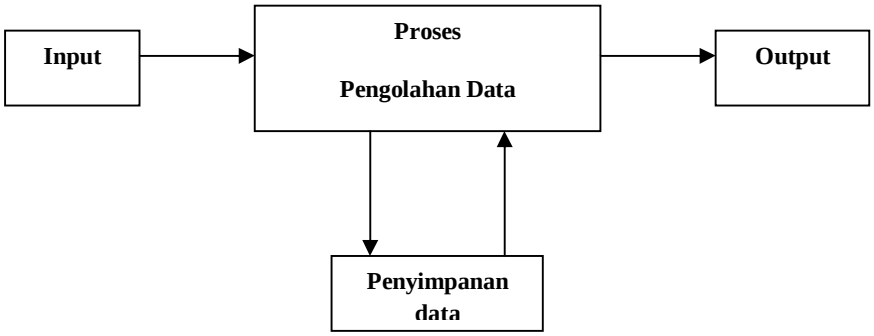
3. *Proses Pengolahan Data*

Bagian ini merupakan proses mengolah data yang diinputkan dengan nenerapkan metode-metode, teknik-teknik, algoritma-algoritma yang ada. Proses ini menghasilkan data output yang akan dioutputkan kepada pengguna program.

4. *Penyimpanan Data*

Bagian ini merupakan proses menyimpan data dalam memori atau device penyimpanan data seperti harddisk, CD dll.

Berikut ini adalah blok diagram struktur dari suatu program secara umum.



Ganbar 2.2 Blok Diagram

2.4 Tool Yang Digunakan Untuk Membangun Aplikasi

2.4.1 Borland Delphi 7.0

Borland Delphi adalah suatu bahasa pemograman (*development language*) yang digunakan untk merancang suatu aplikasi program. *Borland Delphi 7.0* merupakan bahasa pemrograman berbasis windows. Yang menjadi pilihan bagi sebagian *programmer* untuk membuat aplikasi, hai ini disebabkan kelebihan yang ada pada Borland Delphi 7.0 tersebut. Borland delphi merupakan aplikasi *database* yang berbasis objek pascal dari borland. Delphi juga memberikan fasilitas pembuatan aplikasi visual. *Delphi* menjadi pilihan untuk membuat aplikasi visual karena produktifitasnya tinggi. *Delphi* 7.0 memberikan fasilitas untuk dua *platform windows* dan *linux*.

Berikut ini sebagian kecil dari kelebihan *Borland Delphi 7.0* diantaranya :

1. Berbasiskan OOP (*Object Oriented Programming*). Setiap bagian yang ada pada program dipandang sebagai suatu object yang mempunyai sifat-sifat yang dapat diubah dan diatur.
2. IDE yang berkualitas. Delphi memiliki lingkungan pengembangan yang lengkap.Terdapat menu-menu ysng memudahkan anda mengatur proyek pengembangan *software*.
3. Proses Kompilasi yang cepat. *Delphi* memiliki kecepatan kompilasi yang tidak perlu diragukan. Saat aplikasi yang anda buat dijalankan dilingkungan Delphi , aplikasi tersebut otomatis di-*compile* secara terpisah. Kecepatan kompilasi *Delphi* lebih baik dibanding dengan *Visual Basic* dan C++ *Builder*, namun C++ *Builder* memiliki ukuran file hasil kompilasi yang lebih kecil dibanding *Visual Basic* dan *Delphi*.
4. Mudah digunakan. *Delphi* menggunakan bahasa *object pascal* yang telah mendunia. Menggunakan *Delphi*, anda dapat membangun aplikasi apa saja, bahkan yang kompleks sekalipun, akses ke *hardware* misalnya. Anda mungkin sedikit lebih lelah jika menggunakan *Delphi* untuk mengakses *hardware* tetapi itu lebih mudah dibanding bahasa lain dan tidak ada artinya dibanding kompleksitas *pemrograman hardware*.
5. Aplikasi yang dapat dihasilkan. *Delphi* bersifat *multi-purpose*, dapat digunakan untuk berbagai keputusan pengembangan aplikasi mulai perhitungan sederhana sampai aplikasi multimedia bahkan yang terkoneksi ke *internet*. Tetapi harus dipahami bahwa kehebatan *Delphi* paling utama adalah

cara termudah untuk mengakses Database menggunakan komponen-komponen yang disediakan.

6. Satu *file Exe*. Setelah anda merancang program dala IDE *Delphi*, *Delphi* akan mengkompilasinya menjadi sebuah *file executeable* tunggal. Program yang anda buat langsung didistribusikan dan dijalankan pada *computer* lain tanpa perlu menyertakan file DLL dari luar.

2.4.2 MySQL

2.4.2.1 Pengertian MySQL

MySQL adalah sebuah server database SQL multiuser dan multi-threaded. SQL sendiri adalah salah satu bahasa database yang paling populer di dunia. Implementasi program server database ini adalah program daemon 'mysqld' dan beberapa program lain serta beberapa pustaka. MySQL dibuat oleh TcX dan telah dipercaya mengelola sistem dengan 40 buah database berisi 10,000 tabel dan 500 di antaranya memiliki 7 juta baris (kira-kira 100 gigabyte data). *Database* ini dibuat untuk keperluan sistem database yang cepat, handal dan mudah digunakan. Walaupun memiliki kemampuan yang cukup baik, MySQL untuk sistem operasi Unix bersifat *freeware*, dan terdapat versi *shareware* untuk *system* operasi *windows*.

2.4.2.2 Sistem Server Database MySQL

Sistem *database* MySQL memiliki sistem sekuritas dengan tiga *verifikasi* yaitu user(name), password dan host. *Verifikasi* host memungkinkan untuk membuka sekuriti di *'localhost'*, tetapi tertutup bagi host lain (bekerja di lokal komputer). Sistem sekuriti ini ada di dalam *database* MySql dan pada tabel user. Proteksi juga dapat dilakukan terhadap database, tabel, hingga kolom secara terpisah.

2.4.2.3 Fungsi-fungsi MySQL

MySQL memiliki fungsi-fungsi standar SQL dan beberapa kemampuan tambahan. Secara lengkap dokumentasi ini terdapat pada manual MySQL. Namun demikian pada bagian ini perlu disajikan beberapa fungsi SQL yang sering digunakan dalam halaman web.

- SELECT <column,...> FROM <table_name,...> WHERE <where_definition> GROUP BY <column,...> ORDER BY <column, ...> [ASC|DESC]
- INSERT INTO <table_name> VALUES(<list_of_data>)
- REPLACE INTO <table_name> VALUES(<list_of_data>)
- UPDATE <table_name> SET column=<expression> WHERE <where_definition>
- DELETE FROM <table_name> WHERE <where_definition>

2.5 Tools Perangkat keras (*Hardware*)

2.5.1 *Barcode Scaner*

Barcode adalah sebuah bentuk *artificial identifier*. *Barcode* merupakan sebuah kode mesin yang dapat dibaca. *Barcode* terdiri dari sebuah bentuk bar dan spasi (hitam dan putih) dalam rasio yang didefinisikan yang mempresentasikan karakter *alphanumerik*. Kode baris digambarkan dalam bentuk baris hitam tebal dan tipis yang disusun berderet sejajar *horisontal*. Untuk membantu pembacaan secara manual dicantumkan juga angka-angka dibawah kode baris tersebut. Angka-angka tersebut tidak mendasari pola kode baris yang tercantum. Ukuran dari kode baris tersebut dapat diperbesar maupun diperkecil dari ukuran nominalnya tanpa tergantung dari mesin yang membaca.Alat yang digunakan untuk membaca *barcode* adalah *barcode scanner*.

Penggunaan *barcode scanner* sangat mudah sehingga pengguna (*operator*) hanya memerlukan sedikit latihan. *Barcode scanner* dapat membaca informasi / data dengan kecepatan yang jauh lebih tinggi dan memiliki tingkat ketelitian yang lebih tinggi. Bentuk *Barcode* ada dua jenis, yaitu :

1. *Barcode* satu dimensi (1D)
2. *Barcode* dua dimensi (2D)

1. *Barcode* Satu Dimensi

Barcode satu dimensi biasanya dinamakan *linear bar codes* (kode berbentuk baris). Contoh *barcode* satu dimensi adalah sebagai berikut :

- ***Code* 39 (code 3 of 9)**

Type Barcode ini Adalah sebuah *barcode alphanumerik* (full ASCII) yang memiliki panjang baris yang bervariasi. Aplikasi *barcode* jenis code 39 adalah untuk *inventory*, *asset tracking* dan digunakan pada tanda pengenal identitas.



Gambar 2.3. *Barcode* jenis Code 39

- ***Code* 128**

Type Barcode ini Adalah suatu *barcode* alphanumerik (full ASCII) yang memiliki kerapatan (density) yang sangat tinggi dan panjang baris yang bervariasi. *Barcode code* 128 ideal untuk aplikasi seperti *shipping and warehouse management* (pangaturan maskapai pelayaran dan pengelolaan gudang).



Gambar 2.4 *Barcode* jenis Code 128

- ***Interleaved* 2 of 5**

Interleaved 2 of 5 adalah sebuah *barcode* yang berbentuk numerik dan memiliki panjang baris yang bervariasi. *Barcode interleaved* 2 of 5 dapat dipergunakan untuk aplikasi industri dan laboratorium.



Gambar 2.5 *Barcode* jenis *Interleaved* 2 of 5

- **UPC (*Universal Product Code*)**

UPC (*Universal Product Code*) adalah sebuah *barcode* yang berbentuk numerik dan memiliki panjang baris yang tetap (*fixed*). UPC digunakan untuk pelabelan pada produk-produk kecil/eceran (*retail product labeling*). Simbol ini dibuat untuk kemudahan pemeriksaan keaslian suatu produk. Bilangan-bilangan UPC harus diregistrasikan atau terdaftar di Uniform Code Council.



Gambar 2.6 *Barcode* jenis UPC

2. *Barcode* Dua Dimensi

Barcode Dua Dimensi adalah *barcode* yang dikembangkan lebih dari sepuluh tahun lalu, tetapi baru sekarang ini mulai semakin populer. *Barcode* dua dimensi ini memiliki beberapa keuntungan dibandingkan *linear bar codes* (*barcode* satu dimensi) yaitu, dengan menggunakan *barcode* dua dimensi, informasi atau data yang besar dapat disimpan di dalam suatu ruang (*space*) yang lebih kecil.

Contoh *barcode* dua dimensi adalah "symbology PDF417" yang dapat menyimpan lebih dari 2000 karakter di dalam sebuah ruang (*space*) yang berukuran 4 inch persegi (in^2).



Gambar 2.7 *Barcode* jenis PDF417