BAB II

LANDASAN TEORI

2.1 Konsep Dasar Prototipe

Prototipe merupakan bentuk awal (contoh) atau standar ukuran dari sebuah entitas. Dalam desain, sebuah prototipe dibuat sebelum dikembangkan atau justru dibuat khusus untuk pengembangan sebelum dibuat dalam skala sebenarnya atau sebelum diproduksi secara massal.

Dalam menyelesaikan proyek *software* akan terdapat satu pendapat bahwa masalah pertama adalah memperoleh kebutuhan dari *user*. Permasalahan kedua adalah berdasarkan persetujuan spesifikasi fungsional (FS). Spesifikasi fungsional mencoba untuk menggambarkan sistem yang berbasis grafik dan narasi, tetapi gambar dan penjelasan tidak dapat menerangkan cara sistem tersebut berjalan, berlaku, dan mempengaruhi bisnis *user*. Sebagai tambahan, spesifikasi fungsional biasanya menimbulkan kesalah pahaman.

Kesalah pahaman antara *user* dan analis mengakibatkan perubahan yang berarti atau sistem tidak akan pernah sempurna dalam pelaksanaannya atau sekaligus ditolak. Prototipe dapat memecahkan masalah ini untuk tipe-tipe tertentu dalam sistem.

Seperti halnya ketika akan menilai sebuah mobil tanpa mencobanya, *user* juga tidak dapat menilai dari spesifikasi fungsional, bagaimana sistem akan berlaku dan berjalan. Tetapi jika *user* dapat melihat, menyentuh dan

menggunakan ‘model’ atau prototipe dari tujuan sistem dapat langsung menilai kegunaan sistem. Jika perubahan diperlukan prototipe dapat dimodifikasi, memungkinkan dimodifikasi beberapa kali sampai keadaaan yang ditetapkan *user*.

2.1.1 Keuntungan dari prototipe

1. *User* dapat mempertimbangkan sedikit perubahan selama masih bentuk prototipe.
2. Memberikan hasil yang lebih akurat dari pada perkiraan sebelumnya, karena fungsi yang diinginkan dan kerumitannya sudah dapat diketahui dengan baik.
3. *User* merasa puas. Pertama, *user* dapat mengenal melalui komputer. Dengan melakukan prototipe (dengan analisis yang sudah ada), *user* belajar mengenai komputer dan aplikasi yang akan dibuatkan untuknya. Kedua, *user* terlibat langsung dari awal dan memotivasi semangat untuk mendukung analisis selama proyek berlangsung.

2.2 Konsep Dasar Aplikasi

2.2.1 Definisi Aplikasi

Aplikasi adalah sebuah program komputer yang dibuat khusus untuk menjalankan fungsi-fungsi tertentu sesuai dengan kebutuhan *user* yang digunakan untuk mempercepat suatu pekerjaan. Dengan menggunakan sistem komputerisasi, diharapkan pekerjaan dapat dilakukan dengan cepat. Aplikasi dapat dibagi menjadi dua bagian, yaitu aplikasi yang digunakan dan terhubung pada jaringan

komputer (*client server*) dan aplikasi yang berdiri sendiri tidak terhubung dengan jaringan komputer (*stand alone*).

2.2.2 Arsitektur Aplikasi *Client-Server*

Arsitektur aplikasi *client-server* didasarkan pada hal yang sederhana yaitu komputer yang berbeda melakukan tugas yang berbeda, dan setiap komputer dapat dioptimalkan untuk tugas tertentu sehingga dapat memisahkan *Database Management System* (DBMS) dari aplikasi *client*.

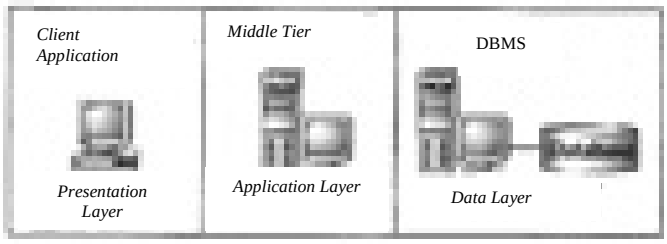
Pada arsitektur aplikasi *client-server*, aplikasi dipecah-pecah ke dalam dua komponen utama yang bekerja sama untuk mencapai satu tujuan bersama. Komponen-komponen ini disebut dengan *tier* (tingkat), dan setiap tingkat mengimplementasikan fungsi yang berbeda-beda.

Pada arsitektur dua tingakat atau *client-server*, *client* terhubung langsung ke *database server*. Setiap aplikasi yang terhubung ke SQL *Server* atau *MySQL* dan mengambil informasi, misalnya nama pemilih atau nama calon/kandidat, adalah aplikasi *client-server* (*two-tier*). Peran dari *database* adalah mengakses dan meng-*update* data. Dengan kata lain, *client* bertanggung jawab menampilkan data kepada *user*, mengirimkan *input* dari *user*, mempersiapkan permintaan yang sesuai untuk *database server*, dan akhirnya mengimplementasikan suatu hal yang disebut *Business Rules*. *Business Rules* adalah prosedur yang khusus digunakan oleh suatu instansi atau perusahaan. Sebagai contoh Komisi Pemilihan Umum (KPU) memiliki aturan yang membatasi usia minimum seorang pemilih adalah 17 tahun atau sudah menikah. Aturan tersebut harus diterjemahkan ke dalam kode

program *tool developer*, yang akan dijalankan pada *client*. Pada sisi *server* dapat ditulis kode program yang akan dijalankan, tetapi tidak dapat memindahkan semua pemrosesan kembali ke *server*.

Hal ini terdapat dalam sebuah tingkat ketiga yang disebut dengan *middle tier*. Tingkat menengah ini merupakan sebuah objek yang berada di antara aplikasi *client* dan *server*. Objek ini berisi fungsi-fungsi *business rules* (kode program) yang telah didefinisikan sebelumnya, dan akhirnya aplikasi *client* tidak terlalu dibebankan dengan *business rule* yang awalnya dijalankan di sisi *client*.

Keuntungan utama dari *middle tier* ini adalah dapat mengisolasi *client* dari *server*. *Client* tidak lagi mengakses *database*, tetapi memanggil fungsi-fungsi yang dimiliki oleh objek pada *middle tier*. Model aplikasi tiga tingkat ini memecah komponen-komponen aplikasi ke dalam tiga kategori atau lapisan seperti yang terlihat pada gambar di bawah ini.



Gambar II.1 Tiga kategori model aplikasi tiga tingkat

2.3 Konsep Dasar *E-Election*

E-Election merupakan terobosan pendayagunaan media elektronik dalam kaitannya dengan pemilu sebagai *influence* (pengaruh) dari kehadiran teknologi

informasi dan komunikasi saat ini. Tahun 1982, Indonesia sudah memanfaatkan teknologi informasi (TIK) dan sistem komputasi yang sudah terintegrasi dalam pemilu pada saat itu, dalam hal ini mungkin kecepatan perhitungan dan pengiriman informasi data belum secepat sistem yang digunakan sekarang. Pemilu berbantukan komputer (*E-Election*) merupakan terobosan dalam proses pemilihan untuk melengkapi kekurangan pada pemilu konvensional. Pemanfaatan TIK dalam *E-Election* ini dapat memberikan kemudahan bagi aparatur penyelenggara pemilu maupun peserta pemilu dalam melaksanakan tugasnya.

2.4 Konsep Perancangan Terstruktur

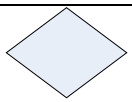
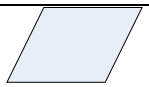
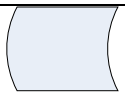
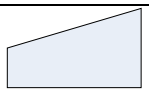
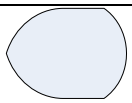
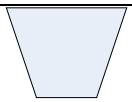
Pendekatan perancangan terstruktur dimulai dari awal 1970. Pendekatan terstruktur dilengkapi dengan alat-alat (*tools*) dan teknik-teknik (*techniques*) yang dibutuhkan dalam pengembangan sistem, sehingga hasil akhir dari sistem yang dikembangkan akan diperoleh sistem yang strukturnya didefinisikan dengan baik dan jelas. Melalui pendekatan terstruktur, permasalahan yang komplek di organisasi dapat dipecahkan dan hasil dari sistem akam mudah untuk dipelihara, fleksibel, lebih memuaskan pemakainya, mempunyai dokumentasi yang baik, tepat waktu, sesuai dengan anggaran biaya pengembangan, dapat meningkatkan produktivitas dan kualitasnya akan lebih baik (bebas kesalahan).

2.4.1 *Flowmap*

Flowmap adalah campuran peta dan *flowchart*, 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. Simbol-simbol *flowmap*:

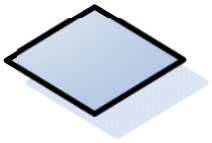
Tabel II.1. Simbol-simbol *flowmap*

Simbol	Keterangan
	Proses : mempresentasikan operasi
	Keputusan : keputusan dalam program
	Dokumen : *input*/*output* dalam format yang dicetak
	*Input*/*output* : mempresentasikan *input* data atau *output* data yang diproses atau informasi
	*Online storage* : *input*/*output* yang menggunakan penyimpanan akses langsung
	Manual *input* : *input* yang dimasukkan secara manual dari *keyboard* atau *mouse*
	*Display* : *output* yang ditampilkan pada terminal
	Manual *operation* : operasi manusia
	*Punched card* : *input*/*output* yang menggunakan kartu.
	Anak panah : mempresentasikan alur kerja

2.4.2 *Entity Relationship Diagram* (ERD)

ERD adalah suatu pemodelan dari basisdata relasional yang didasarkan atas persepsi di dalam dunia nyata, dunia ini senantiasa terdiri dari sekumpulan objek yang saling berhubungan antara satu dengan yang lainnya. Suatu objek disebut *entity* dan hubungan yang dimilikinya disebut *relationship*. Suatu *entity* bersifat unik dan memiliki atribut sebagai pembeda dengan *entity* lainnya. Contoh : *entity* pemilih, mempunyai atribut nip (nomor induk pemilih) yang bersifat unik, nama, alamat dan atribut yang lainnya. ERD terdiri dari:

Tabel II.2. Simbol-simbol ERD

Simbol	Keterangan
	Menggambarkan himpunan entitas. Entitas merupakan objek utama yang informasi akan disimpan, biasanya berupa kata benda, seperti; Mahasiswa, Dosen, Nasabah, Mata Kuliah, Ruangan, dan lain-lain. Objek dapat berupa benda nyata maupun abstrak.
	Menggambarkan atribut-atribut entitas. Atribut merupakan deskripsi dari objek yang bersangkutan
	Menggambarkan hubungan antara himpunan entitas.
	Relationship merupakan suatu hubungan yang terjalin antara dua entitas yang ada.

ERD sebagai suatu pemodelan setidaknya memiliki beberapa karakteristik dan manfaat sebagai berikut:

1. Memudahkan untuk dilakukannya analisis dan perubahan sistem sejak dini, bersifat murah dan cepat
2. Memberikan gambaran umum akan sistem yang akan di buat sehingga memudahkan *developer*.
3. Menghasilkan dokumentasi yang baik untuk client sebagai bahan diskusi dengan bentuk E-R.
4. Memudahkan untuk dilakukannya analisis dan perubahan sistem sejak dini, bersifat Diagram itu sendiri.
5. Kamus data bagi bagi para pengembang *database*.

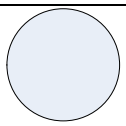
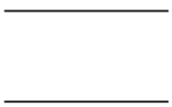
2.4.3 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. 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. Komponen DFD menurut Yourdan dan De Marco:

Tabel II.3. Komponen DFD menurut Yourdan dan De Marco

Nama	Keterangan
	Terminator Sumber (*source*) : terminator yang menjadi sumber.
	Terminator Tujuan (*sink*) : terminator yang menjadi tujuan data/informasi sistem.
	Proses : menggambarkan bagian dari sistem yang mentransformasikan *input* menjadi *output*.
	Data *store* : biasanya berkaitan dengan penyimpanan-penyimpanan, seperti *file* atau *database* yang berkaitan dengan penyimpanan secara komputerisasi, misalnya *file* disket, *file* harddisk, *file* pita magnetik. Data *store* juga berkaitan dengan penyimpanan secara manual seperti buku alamat, *file folder*, dan agenda.
	Alur data ini digunakan untuk menerangkan perpindahan data atau paket data/informasi dari satu bagian sistem ke bagian lainnya.

2.5 Pengenalan PHP (*Personal Home Page*)

PHP (*Personal Home Page*) merupakan *script* untuk pemrograman *web server-side*, *script* yang membuat dokumen HTML secara *on the fly*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan teks *editor* atau *editor* HTML. Dengan menggunakan PHP, maintanance suatu *situs web* menjadi lebih mudah. Proses *update* data dapat

dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan *script* PHP. PHP/FI merupakan nama awal dari PHP, dibuat pertama kali oleh Rasmus Lerdoff. PHP awalnya merupakan program CGI yang dikhususkan untuk menerima *input* melalui *form* yang ditampilkan dalam *browser web*. *Software* ini disebarkan dan dilisensikan sebagai perangkat lunak *Open Source*.

Adapun keunggulan yang dimiliki oleh PHP adalah:

1. *Life Cycle* yang sangat singkat, sehingga PHP selalu *up to date* mengikuti perkembangan teknologi internet.
2. *Cross Platform*, yakni PHP dapat dipakai di hampir semua *webserver* yang ada di pasaran (terutama Apache dan Microsoft IIS) dan dijalankan pada berbagai sistem operasi (Linux, Windows, FreeBSD).
3. PHP mendukung koneksi ke banyak *database* baik yang gratis maupun komersil, seperti *MySQL, mSQL, Oracle, Microsoft SQL Server, Interbase*, dan banyak lagi.
4. PHP bersifat *open source* dan gratis. Kemudahan dalam mendapatkan dokumentasi. di Internet, kita tidak akan sulit untuk mencari baik itu referensi, kode-kode PHP yang sudah jadi dan juga mengajukan pertanyaan pada grup-grup diskusi yang di dalamnya banyak sekali para master PHP.

2.6 *MySQL*

MySQL merupakan sebuah *software* yang berguna sebagai suatu *database server* yang cukup terkenal. Kepopulerannya seiring dengan *user*an *script* PHP

untuk *web programming*. *Database server* itu sendiri merupakan suatu *software* yang bertugas untuk melayani permintaan (*request*) *query* dari *client*.

MySQL sebagai suatu *database server* mempunyai beberapa kemampuan, salah satunya harus menyediakan suatu sistem manajemen *database* yang dapat mengatur bagaimana menyimpan, menambah, mengakses data dan transaksi-transaksi *database* lainnya. *MySQL* cepat sekali berkembang, karena *MySQL* merupakan suatu *software* yang *Open Source*.

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat komersial. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh *user* maupun program-program aplikasinya. Sebagai *database server*, *MySQL* dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query MySQL* bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase.

MySQL memiliki beberapa keistimewaan, antara lain :

1. *Portabilitas*. *MySQL* dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga*, dan masih banyak lagi.
2. *Open Source*. *MySQL* didistribusikan secara *open source*, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.
3. *Multiuse*. *MySQL* dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*. *MySQL* memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Jenis Kolom. *MySQL* memiliki tipe kolom yang sangat kompleks, seperti *signed/unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. Perintah dan Fungsi. *MySQL* memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
7. Keamanan. *MySQL* memiliki beberapa lapisan sekuritas seperti *level subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. *MySQL* mampu menangani basis data dalam skala besar, dengan j*UML*ah rekaman (*records)* lebih dari 50 juta dan 60

ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas. *MySQL* dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, *Unix soket* (UNIX), atau *Named Pipes* (NT).
10. Lokalisasi. *MySQL* dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. *MySQL* memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. Klien dan Peralatan. *MySQL* dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basisdata, dan pada setiap peralatan yang ada disertakan petunjuk *online*.
13. Struktur tabel. *MySQL* memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basisdata lainnya semacam PostgreSQL ataupun Oracle.

2.7 Javascript

Javascript merupakan *cross-platform* yang dikembangkan oleh Netscape dan pertama kali digunakan dalam *browser* Netscape. Javascript dibuat agar mudah diintegrasikan kedalam program dan aplikasi lain, misalnya *browser*. Sebagian besar *browser* saat ini sudah mendukung javascript. Oleh karena itu,

script dari javascript biasanya dimasukkan kedalam suatu HTML dan dieksekusi di*browser* lain.

2.8 HTML

HTML merupakan singkatan dari *hypertext markup language*, yang merupakan program penulisan informasi pada sebuah *hompage*. Penulisan HTML dapat dilakukan menggunakan alat bantu seperti *Notepad* yang terdapat pada *windows* atau *simple text machintosh*. Selain itu juga dapat digunakan editor HTML seperti *Macromedia Dreamweaver* dan *Microsoft Frontpage* yang dapat memudahkan dalam menulis HTML dan memungkinkan dokumen HTML yang dibuat dapat diakses oleh berbagai jenis *Browser*. HTML berupa kode-kode tag yang memberikan instruksi pada *Web Browser* untuk memberikan tampilan sesuai yang diinginkan.

2.9 Macromedia Dreamweaver

Macromedia Dreamweaver adalah *tools* HTML *editor* professional, untuk mendesain, melakukan *coding*, dan pengembangan dari *website*, *web page*, dan *web application. Tools* ini memiliki fitur *visual editing* yang memungkinkan *user* membuat halaman *web* tanpa harus menulis barisan kode HTML. Dreamweaver membantu membangun *dynamic database-backed web applications* dengan menggunakan bahasa *server* seperti ASP, ASP.NET, *ColdFusion Markup Language*(CFML), JSP dan PHP. (Rickyanto, 2002, p3).

2.10 *Web Server*

Web server menurut Minoli(1998,p33) berpendapat bahwa sebuah *web server* adalah seuatu program untuk menawarkan pelayanan yang bisa diperoleh seluruh jaringan. *Web server* merupakan suatu tipe *server* khusus yang dapat berkomunikasi langsung dengan *client* menggunakan HTTP, *web server*

2.11 *Web browser*

Menurut Ellsworth, *web browser* adalah *software* yang dijalankan pada komputer pemakai yang meminta informasi dari *web server* dengan menampilkannya sesuai dengan nama *file* itu sendiri.

2.12 *Client-Server*

Client-Server adalah arsitektur jaringan yang memisahkan *client* (biasanya aplikasi yang menggunakan GUI) dengan *server*. Masing-masing *clien*t dapat meminta data atau informasi dari *server*. Perbedaan karakteristik sistem *client* dan *server*:

1. *Service* (layanan)
    a. Hubungan antara proses yang berjalan pada mesin yang berbeda.
    b. Pemisahan fungsi berdasarkan ide layanannya.
    c. *Server* sebagai *provider* dan *client* sebagai konsumen.

2. *Sharing Resources* (sumber daya)

Server bisa melayani beberapa *client* pada waktu yang sama dan meregulasi akses bersama untuk *share* sumber daya dalam menjamin konsistensinya

3. *Asymmetrical Protocol* (protokol yang tidak simetris)

Many-to-one relationship antara *client* dan *server*. *Client* selalu menginisiasikan dialog melalui layanan permintaan dan *server* menunggu secara pasif *request* dari *client*.

4. Pengaksesan yang mudah

Proses yang dilakukan *server* boleh terletak pada mesin yang sama atau pada mesin yang berbeda melalui jaringan. Lokasi *server* harus mudah diakses dari *client*.

5. Pesan berbasiskan komunikasi

Interaksi *server* dan *client* melalui pengiriman pesan yang menyertakan permintaan dan jawaban.

6. Pemisahan *interface* dan implementasi

Server bisa di-*upgrade* tanpa mempengaruhi *client* selama *interface* pesan yang diterbitkan tidak berubah.

2.13 Bar*code*

Bar*code* pada dasarnya adalah susunan garis vertikal hitam dan putih dengan ketebalan yang berbeda, sangat sederhana tetapi sangat berguna, dengan kegunaan untuk menyimpan data-data spesifik misalnya kode produksi, tanggal kadaluwarsa, nomor identitas dengan mudah dan murah, walaupun teknologi

semacam itu terus berkembang dengan ditemukannya media *magnetic*, *RFID*, *electronics tags*, serial eeprom (seperti pada *smart card*), bar*code* terus bertahan dan masih memiliki kelebihan-kelebihan tertentu yaitu murah dan mudah, sebab media yang digunakan adalah kertas dan tinta, sedangkan untuk membaca bar*code* ada begitu banyak pilihan di pasaran dengan harga yang relatif murah mulai dari yang berbentuk pena (wand), *slot*, *scanner*, sampai ke *CCD* dan bahkan kita dapat membuatnya sendiri.

Penggunaan bar*code* sering dijumpai pada produk makanan, minuman, obat-obatan, tiket pesawat, dan produk-produk lainnya. Bar*code* pada dasarnya adalah susunan garis vertikal hitam dan putih dengan ketebalan yang berbeda, sangat sederhana tapi sangat berguna. Kegunaannya adalah untuk meyimpan data-data spesifik misalnya kode produksi, tanggal kadaluwarsa, nomor identitas produk dan lain sebagainya. Bar*code* ada dua bentuk yaitu bar*code* satu dimensi (1D) dan bar*code* dua dimensi (2D).

Bar*code* yang digunakan adalah bar*code* 128. Bar*code* 128 adalah suatu bar*code alphanumerik* (full ASCII) yang memiliki kerapatan (*density*) yang sangat tinggi dan panjang baris yang bervariasi. Bar*code code* 128 ideal untuk aplikasi seperti pengaturan pengelolaan data gudang. Setiap karakter pada *code* 128 dikodekan oleh 3 bar dan 3 spasi (atau 6 elemen) dengan ketebalan masing-masing elemen 1 sampai 4 kali ketebalan minimum (*module*), jika dihitung dngan satuan *module* maka tiap karakter *code* 128 terdiri dari 11 *module* kecuali untuk stop *character* yang terdiri dari 4 bar 3 spasi (13 *module*). Jumlah total *module*

untuk bar selalu genap sedangkan untuk spasi selalu ganjil, selain itu *code* 128 memiliki 3 start *character* yang berbeda sehingga *code* 128 memiliki 3 sub set karakter yang bersesuaian dengan start *character*nya.

Tabel II.4. Tabel karakter *code* 128

Karakter ASCII	Digit Biner									Nilai Karakter
	B	S	B	S	B	S	B	S	B	
0	0	0	0	1	1	0	1	0	0	0
1	1	0	0	1	0	0	0	0	1	1
2	0	0	1	1	0	0	0	0	1	2
3	1	0	1	1	0	0	0	0	0	3
4	0	0	0	1	1	0	0	0	1	4
5	1	0	0	1	1	0	0	0	0	5
6	0	0	1	1	1	0	0	0	0	6
7	1	0	1	1	0	0	1	0	1	7
8	0	0	0	1	0	0	1	0	0	8
9	1	0	0	1	0	0	1	0	0	9
A	1	0	1	0	0	1	0	0	1	10
B	0	0	1	0	0	1	0	0	1	11
C	1	0	0	0	0	1	0	0	0	12
D	0	0	0	0	1	1	0	0	1	13
E	1	0	1	0	1	1	0	0	0	14
F	0	0	1	0	1	1	0	0	0	15
G	0	0	0	0	0	1	1	0	1	16
H	1	0	0	0	0	1	1	0	0	17
I	0	0	1	0	0	1	1	0	0	18
J	0	0	1	0	1	1	1	0	0	19
K	1	0	0	0	0	0	0	1	1	20
L	0	0	0	0	0	0	0	1	1	21
M	1	0	1	0	0	0	0	1	0	22
N	0	0	1	0	1	0	0	1	1	23
O	1	0	0	0	1	0	0	1	1	24
P	0	0	0	0	1	0	0	1	0	25
Q	0	0	1	0	0	0	1	1	1	26
R	1	0	1	0	0	0	1	1	0	27
S	0	0	0	0	0	0	1	1	0	28
T	0	0	0	0	1	0	1	1	0	29

U	1	1	1	0	0	0	1	0	1	30
V	0	1	1	0	0	0	0	0	1	31
W	1	1	1	0	0	0	0	0	0	32
Z	0	1	0	0	1	0	0	0	1	33
Y	1	1	0	0	1	0	0	0	0	34
Z	0	1	1	0	1	0	0	0	0	35
-	0	1	0	0	0	0	1	0	1	36
.	1	1	0	0	0	0	1	0	0	37
SPACE	0	1	1	0	0	0	1	0	0	38
*	0	1	0	0	1	0	1	0	0	
$	0	1	0	1	0	1	0	1	0	39
/	0	1	0	1	0	0	0	1	0	40
TAMBAH	0	1	0	0	0	1	0	1	0	41
%	0	0	0	1	0	1	0	1	0	42

Perhitungan check *character code* 128 adalah sebagai berikut:

Message : *CODE* 128

Karakter : Start A C O D E 1 2 8

Nilai karakter : 103 35 47 36 37 0 17 18 24

Posisi : 1 2 3 4 5 6 7 8 5

Perhitungan Total : 103 + (35 x 1) + (47 x 2) + (36 x 3) + (37 x 4) +

(0 x 5) + (17 x 6) + (18 x 7) + (24 x 8) = 908

908 / 103 = 8 remainder 84

84 = DC4

Message akhir : (Start A) *CODE* 128 (DC4) (STOP)