

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Konsep Dasar Program**

Konsep pemrograman terstruktur memegang peranan penting dalam merancang, menyusun, memelihara dan mengembangkan suatu program, khususnya program aplikasi yang cukup besar dan kompleks. Proses pemrograman komputer bukan hanya sekedar menulis suatu urutan instruksi yang harus dikerjakan oleh komputer, akan tetapi bertujuan untuk memecahkan suatu masalah serta membuat mudah suatu pekerjaan yang diinginkan oleh pemakai (*user*). Yang menjadi alasan utama mengapa kita belajar pemrograman komputer adalah karena kita ingin memanfaatkan komputer sebagai alat bantu untuk menyelesaikan masalah.

##### **2.1.1. Program**

Menurut Kadir (2012:2), program adalah “Kumpulan instruksi yang digunakan untuk mengatur komputer agar melakukan suatu tindakan tertentu”. Tanpa program, komputer sesungguhnya tidak dapat berbuat apa-apa. Itulah sebabnya, sering dikatakan bahwa komputer mencakup tiga aspek penting, berupa perangkat keras (*hardware*), perangkat lunak (*software*), yang dalam hal ini berupa program; dan perangkat akal (*brainware*) atau orang yang berperan terhadap operasi komputer maupun pengembangan perangkat lunak. Dengan kata lain, program merupakan salah satu bagian penting pada komputer, yang

mengatur komputer agar melakukan tindakan yang sesuai dengan yang dikehendaki oleh pembuatnya.

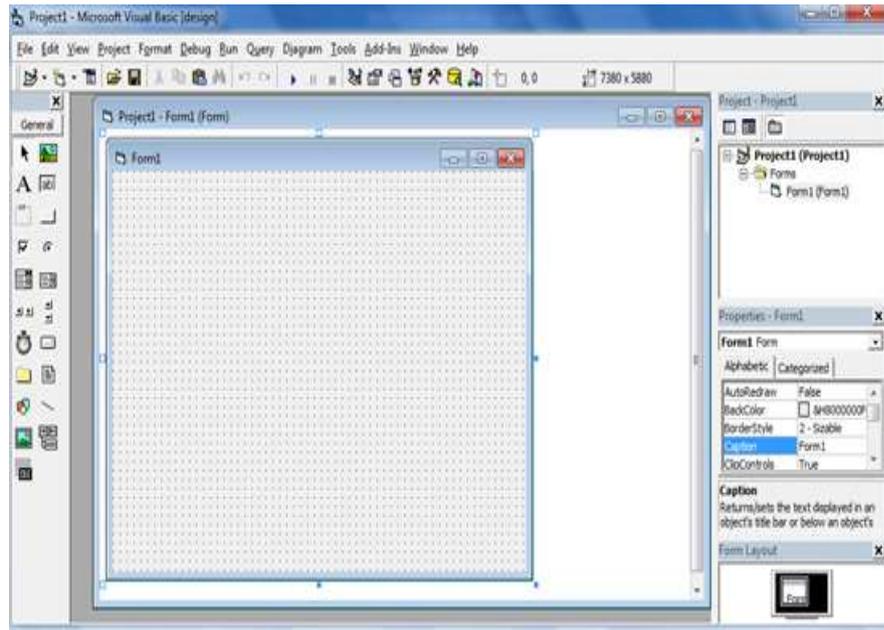
### **2.1.2. Bahasa Pemrograman**

Menurut Munir (2011:13) mengemukakan bahwa “Bahasa pemrograman adalah bahasa *computer* yang digunakan dalam menulis program”.

Bahasa pemrograman yang digunakan dalam penyusunan tugas akhir ini yaitu:

1. *Microsoft Visual Basic 6.0*

Menurut Madcoms (2008:1), *Microsoft Visual Basic 6.0* adalah “Salah satu aplikasi pemrograman *visual* yang memiliki bahasa pemrograman yang cukup populer dan mudah untuk dipelajari”. Basis bahasa pemrograman yang digunakan dalam *visual basic* adalah bahasa BASIC (*Beginners All-Purpose Symbolic Instruction Code*) yang merupakan salah satu bahasa pemrograman tingkat tinggi yang sederhana dan mudah dipelajari. Dengan *Visual Basic*, kita bisa membuat program dengan aplikasi GUI (*Graphical User Interface*) atau program yang memungkinkan pengguna komputer berkomunikasi dengan komputer tersebut menggunakan grafik atau gambar. *Microsoft Visual Basic 6.0* menyediakan berbagai perangkat kontrol yang dapat digunakan untuk membuat program aplikasi dalam sebuah *form* baik aplikasi kecil, sederhana hingga ke aplikasi pengolahan *database*.



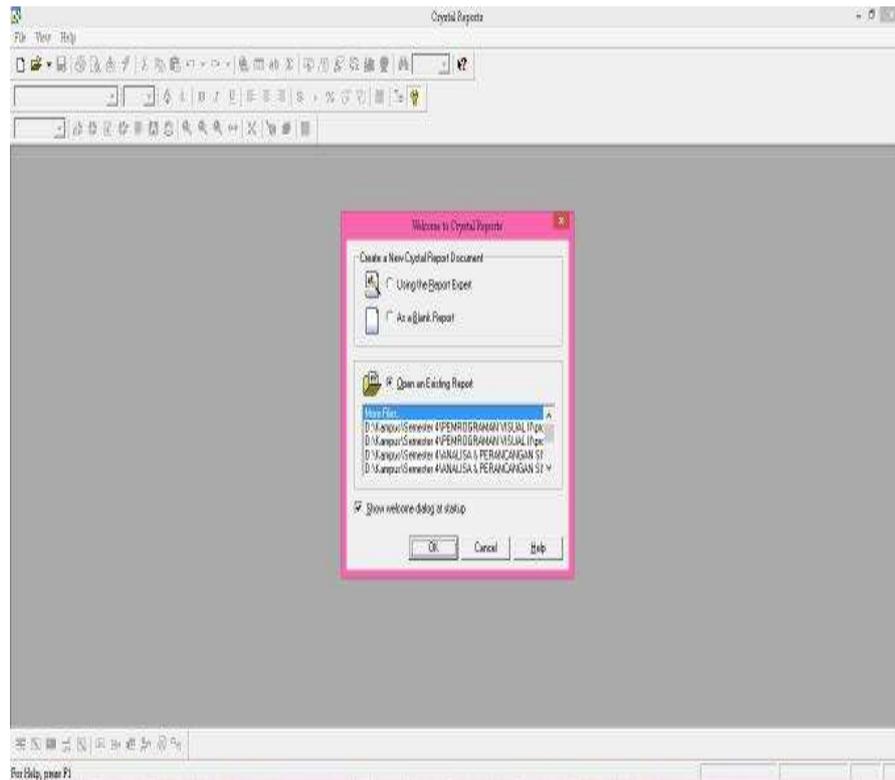
Sumber: Madcoms (2008:1)

Gambar II.1. Tampilan *Microsoft Visual Basic 6.0*

## 2. *Crystal Report*

Menurut Wahana Komputer (2012:61), *Crystal Report* merupakan “Sarana untuk mencetak data yang ada pada tabel”.

Sedangkan menurut Atmoko (2013:3), *Crystal Report 8.5* adalah “Program yang akan kita gunakan untuk membuat *report* program, agar dapat dipahami oleh pengguna, yang *report* tersebut diambil dari kumpulan data yang tersimpan di dalam *database PHPmyAdmin* yang akan kita buat”. Saat ini *Crystal Report* dimiliki oleh SAP, yaitu salah satu perusahaan software ERP terbesar di dunia. Untuk penggunaan *Crystal Report* ini sangat mudah, hanya tinggal *click* dan *drag-drop* saja untuk membuat sebuah laporan yang akan kita gunakan sebagai *report* program yang kita bangun dengan *Microsoft Visual Basic 6.0*, yang hasil dari laporan disesuaikan dengan kebutuhan pengguna.



Sumber: Atmoko (2013:3)

Gambar II.2. Tampilan *Crystal Report*

### 2.1.3. Basisdata

Menurut Silberschatz, dkk dalam Simarmata dan Paryudi (2010:1) mengemukakan bahwa :

Basisdata adalah kumpulan data berisi informasi yang sesuai untuk sebuah perusahaan. Sistem manajemen basisdata (DBMS) adalah kumpulan data yang saling berhubungan dan kumpulan program untuk mengakses data. Tujuan utama sistem manajemen basisdata adalah menyediakan cara menyimpan dan mengambil informasi basis data secara mudah dan efisien.

*Database* menurut Anhar (2010:45) didefinisikan sebagai, “Sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari *field* atau kolom”.

Struktur *file* yang menyusun sebuah *database* adalah *Data Record* dan *Field*.

*Software* yang digunakan dalam pembuatan basis data pada tugas akhir ini antara lain:

## 1. *MySQL*

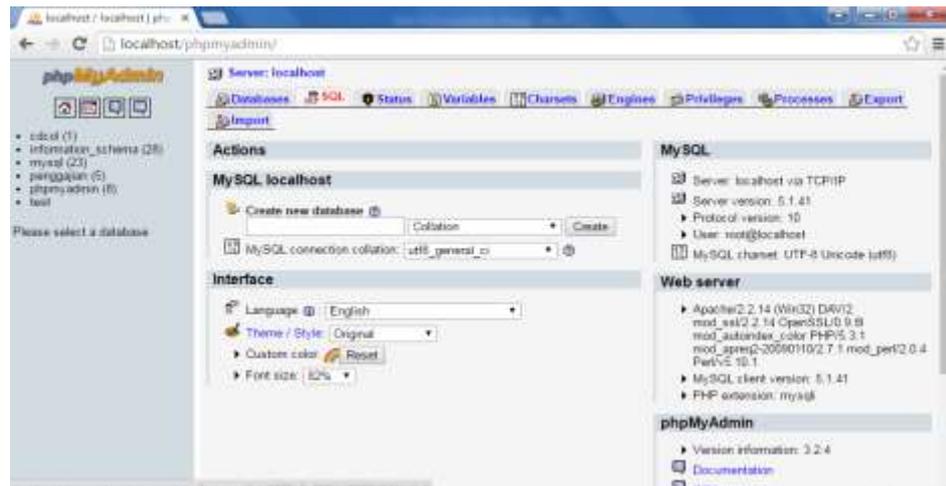
*MySQL (My Structure Query Language)* menurut Anhar (2010:45) adalah, “Salah satu *Database Management System (DBMS)* dari sekian banyak DBMS seperti *Oracle, MS SQL, Postagre SQL*, dan lainnya. *MySQL* berfungsi untuk mengolah *database* menggunakan bahasa *SQL*. *MySQL* bersifat *open source* sehingga kita bisa menggunakannya secara gratis”. Pemrograman *PHP* juga sangat mendukung/support dengan *database MySQL*.

*MySQL* menurut Raharjo (2015:16) merupakan, “*Software RDBMS (atau server database)* yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user (multi user)*, dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*)”.

## 2. *PHPMyAdmin*

*PhpMyAdmin* menurut Riyanto (2011:17) merupakan “Aplikasi *Web* berbasis *PHP* yang telah banyak digunakan untuk administrasi *database MySQL*”. *PhpMyAdmin* mendukung berbagai aplikasi *MySQL*, diantaranya (mengelola basis data, tabel-tabel bidang, relasi, indeks, pengguna, perizinan, dll).

Menurut Nugroho (2009a:13) “*PhpMyadmin* adalah aplikasi berbasis web yang dibuat dari pemrograman *PHP* dan diramu dengan *JavaScript*. *PhpMyadmin* juga dapat disebut sebagai tools yang berguna untuk mengakses *database MySQL Server* dalam bentuk tampilan web”.



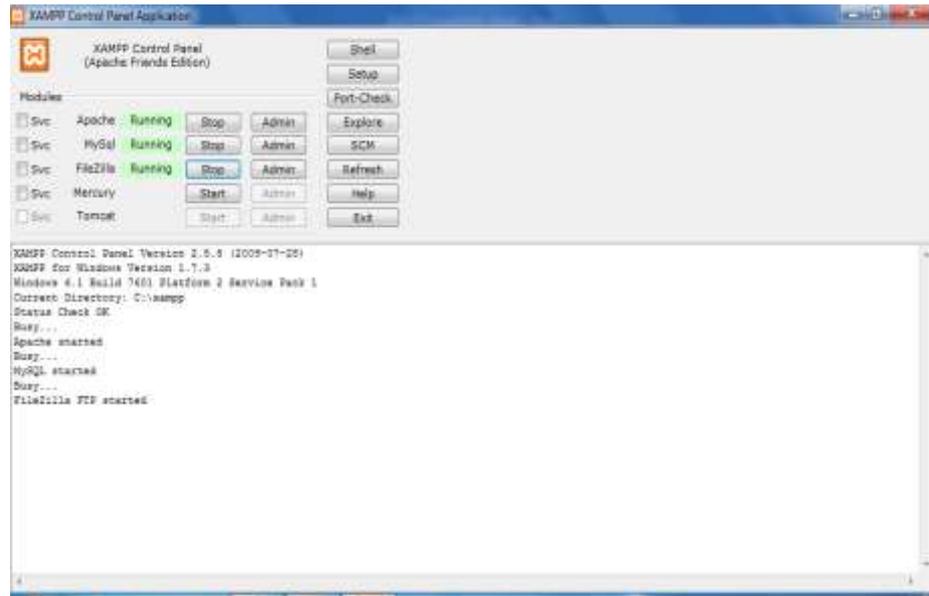
Sumber: Riyanto (2011:17)

Gambar II.3. Tampilan *PHPMyAdmin*

### 3. *XAMPP*

*XAMPP* menurut Murya (2016:14) merupakan: “Sebuah perangkat lunak gratis sehingga bebas digunakan. *XAMPP* berfungsi sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri dari *Apache*, *HTTP Server*, *MySQL database* dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*.”

Menurut Nugroho (2009b:74) *XAMPP* merupakan “Paket *PHP* berbasis *Open Source* yang dikembangkan oleh sebuah komunitas *Open Source*”. Paket yang tersedia pada *XAMPP* versi 1.4.11 diantaranya *Apache HTTPD 2.0.54*, *MySQL 4.1.12*, *PHP 5.0.3*, *FilZilla FTP Server 0.9 Beta*, *PhpMyadmin 2.6.1-pl3* dan lain-lain.



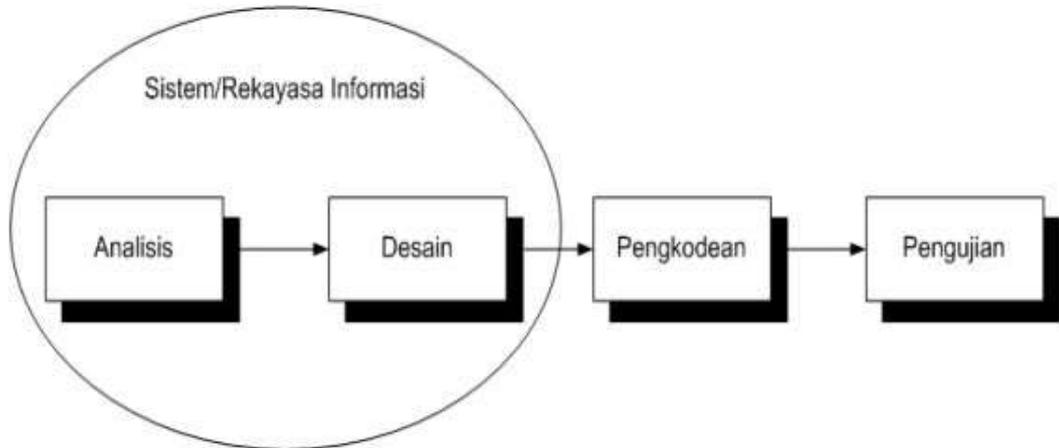
Sumber: Nugroho (2009b:75)

Gambar II.4. Tampilan XAMPP

#### 2.1.4. Model Pengembangan Perangkat Lunak

Menurut Sukamto dan Shalahuddin (2016:28), model air terjun (*waterfall*) adalah “Model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*)”.

Berikut adalah gambar model air terjun menurut Sukanto dan Shalahuddin:



Sumber: Sukanto dan Shalahuddin (2016:29)

Gambar II.5. Ilustrasi Model *Waterfall*

#### 1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

#### 2. Desain

Desain perangkat lunak adalah proses multi langkah yang *focus* pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada

tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

### 3. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

### 4. Pengujian

Pengujian fokus pada perangkat lunak dari segi *logic* dan fungsional untuk memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

### 5. Pendukung (*Support*) atau Pemeliharaan (*Maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak yang harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tetapi tidak untuk membuat perangkat lunak yang baru.

## 2.2. Teori Pendukung

Suatu perancangan program yang baik haruslah didukung dengan suatu peralatan pendukung guna menunjang keberhasilan program dan untuk menjelaskan kepada pengguna bagaimana fungsi dari sistem informasi dapat bekerja dengan suatu logika yang digambarkan dengan simbol-simbol. *Tools* program yang penulis gunakan diantaranya:

### 2.2.1. *Enterprise Relationship Diagram* (ERD)

#### 1. Pengertian *Enterprise Relationship Diagram* (ERD)

Menurut Simarmata dan Paryudi (2010:67), ERD adalah, “Alat pemodelan data utama dan akan membantu mengorganisasi data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan antar entitas. Proses memungkinkan analisis menghasilkan struktur basisdata yang baik sehingga data dapat disimpan dan diambil secara efisien.”

- a. Entitas (*Entity*) adalah sesuatu yang nyata atau abstrak di mana kita akan menyimpan data.
- b. Relasi (*Relationship*) adalah hubungan alamiah yang terjadi antara satu atau lebih entitas, misalnya proses pembayaran pegawai. Kardinalitas menentukan kejadian suatu entitas untuk satu kejadian pada entitas yang berhubungan. Misalnya, mahasiswa bisa mengambil banyak mata kuliah.
- c. Atribut (*Attribute*) adalah ciri umum semua atau sebagian besar instansi pada entitas tertentu. Sebutan lain atribut adalah properti, elemen data dan *field*. Misalnya nama, alamat, nomor pegawai dan gaji adalah atribut entitas pegawai. Sebuah atribut atau kombinasi atribut yang

mengidentifikasi satu dan hanya satu instansi suatu entitas disebut kunci utama atau pengenal. Misalnya nomor pegawai adalah kunci utama untuk pegawai.

## 2. Pemetaan Kardinalitas

Menurut Simarmata dan Paryudi (2010:63) mengatakan, “Pemetaan kardinalitas menyatakan jumlah entitas di mana entitas lain dapat dihubungkan ke entitas tersebut melalui sebuah himpunan relasi”.

Pemetaan kardinalitas sangat berguna dalam menentukan himpunan relasi biner meskipun pemetaan dapat berperan dalam deskripsi himpunan relasi yang melibatkan lebih dari dua himpunan entitas.

Untuk suatu himpunan relasi biner  $R$  antara himpunan entitas  $A$  dan  $B$ , pemetaan kardinalitas harus salah satu dari berikut:

- a. *One-to-One*, sebuah entitas pada  $A$  berhubungan dengan paling banyak satu entitas pada  $B$  dan sebuah entitas pada  $B$  berhubungan dengan paling banyak satu entitas pada  $A$ .
- b. *One-to-Many*, sebuah entitas pada  $A$  berhubungan dengan nol atau lebih entitas pada  $B$ . Sebuah entitas pada  $B$  dapat dihubungkan dengan paling banyak satu entitas pada  $A$ .
- c. *Many-to-One*, sebuah entitas pada  $A$  berhubungan dengan paling banyak satu entitas pada  $B$ . Sebuah entitas pada  $B$  dapat dihubungkan dengan nol atau lebih entitas pada  $A$ .
- d. *Many-to-Many*, sebuah entitas pada  $A$  berhubungan dengan nol atau lebih entitas pada  $B$  dan sebuah entitas pada  $B$  dapat dihubungkan nol atau lebih entitas pada  $A$ .

### 3. Pengertian Logical Record Structure (LRS)

Menurut Hasugian dan Shidiq (2012:608) memberikan batasan bahwa LRS adalah, “Sebuah model sistem yang digambarkan dengan sebuah diagram-ER akan mengikuti pola atau aturan permodelan tertentu dalam kaitannya dengan konvensi ke LRS.”

Menurut Frieyadie (2007:13), LRS (*Logical Record Structure*) adalah, “Hasil dari pemodelan *Entity Relationship* (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antar entitas”. Sebelum tabel dibentuk dari *field* atau atribut entitas secara fisik atau *level internal*, maka harus dibuatkan suatu bentuk *relational model* yang dibuat secara *logic* atau *level external* dan konsep, dari pernyataan tersebut dibutuhkan yang disebut *Logical Record Structure* (LRS).

#### 2.2.2. Pengkodean

##### 1. Pengertian Pengkodean

Menurut Mustakini (2014:384), “Kode digunakan untuk tujuan mengklarifikasi data, memasukkan data ke dalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengannya.” Kode dapat dibentuk dari kumpulan angka, huruf dan karakter-karakter khusus (misalnya %, /, -, \$, #, &, :, @ dan lain sebagainya).

Menurut Kadir (2012:109), “Kode adalah suatu kerangka (*framework*) yang menggunakan angka, huruf, atau kombinasi angka dan huruf untuk memberi tanda terhadap klasifikasi yang sebelumnya telah dibuat”. Kode digunakan untuk mengklasifikasi data, memasukkan data ke dalam komputer dan mengambil bermacam-macam informasi yang berhubungan dengan cara menyingkat kata,

mengurutkan data dalam blok-blok tertentu. Kode dapat dibuat dalam berbagai struktur kode yang berbeda. Setiap struktur mempunyai kelebihan dan kelemahan. Oleh karena itu, perlu dipilih suatu struktur kode yang sesuai sehingga tujuan pemberian kode dapat tercapai.

## 2. Macam-macam Kode

Berikut ini macam-macam kode yang dapat digunakan, yaitu:

### a. Kode Mnemonik (*Mnemonic Code*)

Merupakan kode yang digunakan untuk tujuan mudah diingat. Kode ini dibuat dengan dasar singkatan atau mengambil sebagian karakter dari item yang akan diwakili dengan kode ini. Umumnya kode mnemonic menggunakan huruf, akan tetapi dapat juga menggunakan gabungan huruf atau angka, kebaikan kode ini adalah mudah diingat dan kelemahannya kode dapat terlalu panjang.

### b. Kode Urut (*Sequence Code*)

Merupakan kode yang lainya urut antara satu kode dengan kode yang lainya. Kebaikan kode ini adalah sangat sederhana, mudah diterapkan, kode dapat pendek tapi unik, mudah dicari, baik untuk pengendalian. Kelemahan kode ini antara lain penambahan kode hanya dapat ditambahkan pada akhir urutan dan tidak dapat disisipkan, tidak mempunyai dasar logika tentang informasi item yang diwakilinya, tidak fleksibel bila terjadi perubahan kode.

### c. Kode Blok (*Block Kode*)

Mengklasifikasikan item ke dalam kelompok blok tertentu mencerminkan satu klasifikasi tertentu atas dasar pemakaian maksimum

yang diharapkan. Kebaikan dari kode ini antara lain nilai dari kode mempunyai arti, mudah diperluas, kode dapat ditambah atau dihapus sebagian, proses pembuatan laporan keuangan dapat dilakukan dengan mudah. Kelemahan kode ini adalah panjang kode tergantung dari jumlah bloknya.

d. Kode kelompok (*Group Code*)

Kode kelompok merupakan kode yang berdasarkan field-field dan tiap-tiap field kode mempunyai arti tertentu. Kebaikan kode ini antara lain nilai dari kode mempunyai arti, dapat menunjukkan jenjang dari data. Kelemahannya adalah kode dapat menjadi panjang.

e. Kode Desimal (*Decimal Code*)

Mengklasifikasikan kode atas dasar sepuluh unit angka desimal dimulai dari angka nol (0) sampai angka sembilan (9) atau sepuluh (10) sampai dengan (99) tergantung dari banyaknya kelompok.

3. Syarat merancang kode

Kode digunakan untuk mengklasifikasikan data, memasukan data kedalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan denganya, kode dapat dibentuk dari kumpulan angka, huruf dan karakter-karakter khusus, didalam merancang suatu kode harus diperhatikan beberapa hal yaitu :

a. Harus Mudah Diingat

Agar kode mudah diingat, maka dapat dilakukan dengan cara menghubungkan kode tersebut dengan objek yang diwakili dengan kodenya.

b. Harus unik

Kode tidak boleh sama antara satu dengan yang lainnya.

c. Harus Fleksibel

Kode harus fleksibel sehingga memungkinkan perubahan-perubahan atau penambahan item baru dapat diwakili oleh kode.

d. Harus Efisien

Kode harus sependek mungkin, sehingga mudah diingat dan juga akan efisien bila direkam atau disimpan dikomputer.

e. Harus Konsisten

Kode harus konsisten dengan kode yang telah dipergunakan

f. Harus Distandarisasi

Kode harus distandarisasi untuk seluruh tingkatan dan departemen dalam organisasi. kode yang tidak standar akan mengakibatkan kebingungan, salah pengertian dan dapat cenderung terjadi kesalahan pemakai begitu juga dengan yang menggunakan kode tersebut.

g. Hindari Spasi

h. Hindari Karakter Yang Mirip

Karakter-karakter yang hampir serupa bentuk dan bunyi pengucapannya sebaiknya tidak digunakan dalam kode.

i. Panjang kode harus sama

### 2.2.3. HIPO (*Hirarcy Input Process Output*)

HIPO banyak digunakan sebagai alat desain dan teknik dokumentasi dalam siklus pengembangan sistem yang akan dibuat.

#### 1. Pengertian HIPO

Menurut Mustakini (2014:787), HIPO (*Hirarchy Plus Input Process Output*) merupakan metodologi yang dikembangkan dan didukung oleh IBM. HIPO sebenarnya alat dokumentasi program. Akan tetapi, sekarang HIPO juga banyak digunakan sebagai alat disain dan teknik dokumentasi dalam siklus pengembangan sistem. HIPO berbasis pada fungsi, yaitu tiap-tiap modul di dalam sistem digambarkan oleh fungsi utamanya.

#### 2. Tingkatan Diagram HIPO

Untuk menyediakan output yang tepat dan sesuai dengan kebutuhan-kebutuhan pemakai. Fungsi dari sistem yang digunakan HIPO terdapat tiga tingkatan, yaitu:

##### a. *Visual Tabel of Content (VTOC)*

Diagram ini menggambarkan hubungan dari fungsi-fungsi di sistem secara berjenjang.

##### b. *Overview Diagram*

Menunjukkan secara garis besar hubungan *input*, *process*, dan *output*.

##### c. *Detail Diagram*

Merupakan diagram tingkatan yang paling rendah di diagram HIPO.

### 2.2.4. Diagram Alir Program (*Flowchart*)

#### 1. Pengertian *Flowchart*

Menurut Mustakini (2014:795) mengatakan, “Bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau

prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat komunikasi dan untuk dokumentasi”.

Menurut Putra (2016:11) “Flowchart merupakan bagan aliran yang menggambarkan proses suatu sistem secara mendetail. Flowchart juga menggambarkan hubungan proses dalam sistem”.

## 2. Bentuk *Flowchart*

Dalam pembuatan program komputer biasanya ada dua bentuk *flowchart* yang sering digunakan, yaitu:

### a. *Program Flowchart*

Simbol-simbol yang menggambarkan proses secara rinci dan detail antara instruksi yang satu dengan instruksi yang lainnya di dalam suatu program yang bersifat logika.

### b. *Sistem Flowchart*

Simbol-simbol yang menggunakan urutan prosedur secara detail di dalam suatu sistem komputerisasi dan bersifat fisik.

## 3. Teknik Pembuatan *Flowchart*

Sebelum kita membuat suatu program komputer, yang harus kita lakukan terlebih dahulu adalah membuat *flowchart*. Jenis *flowchart* yang sering digunakan adalah program *flowchart* sehingga akan mempermudah dalam penjelasan alur dari program itu sendiri. Adapun teknik pembuatan program *flowchart* dibagi menjadi dua, yaitu:

- a. *General way*, yaitu teknik pembuatan *flowchart* yang digunakan dalam menyusun logika suatu program yang menggunakan pengulangan proses secara tidak langsung (*non direct loop*).

- b. *Iteration way*, yaitu teknik pembuatan *flowchart* untuk logika program yang cepat serta bentuk permasalahan yang rumit dan pengulangan proses terjadinya bersifat langsung (*direct loop*).