

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Program

Konsep dasar pemrograman sangat erat kaitannya dengan pembuatan program, karena memegang peranan penting dalam perancangan serta pembangunan sebuah program. Suatu program terdiri dari banyak sekali intruksi-intruksi, dan program tidak lebih dari susunan intruksi yang harus benar dan urutan yang benar pula, dan karena harus tersusun dengan benar itu pula maka program harus diurutkan dengan logika program agar dapat memudahkan dalam pembuatan program.

Pembahasan mengenai konsep dasar dari program yang dibahas dalam ruang lingkup penulisan tugas akhir ini adalah pencatatan yang masih menggunakan cara manual akan menjadi terkomputerisasi.

A. Program

Menurut Sutarman (2009:3) “Program adalah barisan perintah atau instruksi yang disusun sehingga dapat dipahami oleh komputer dan kemudian dijalankan sesuai barisan perhitungan *numeric*, dimana barisan perintah tersebut berhingga, berakhir dan menghasilkan *output*”.

Dengan menterjemahkan berarti kita melakukan penulisan program dengan menggunakan suatu bahasa pemrograman komputer yang kita kuasai. Sedangkan konsep dasar program adalah suatu rangkaian instruksi-instruksi dalam bahasa komputer yang disusun secara logis dan sistematis. Proses pemrograman komputer

bukan hanya sekedar menulis suatu urutan instruksi yang harus dikerjakan oleh komputer, akan tetapi bertujuan untuk memecahkan masalah serta membuat pekerjaan atau lainnya yang diinginkan oleh *user*.

Bahasa yang digunakan untuk menginstruksi komputer disebut bahasa pemrograman. Menurut Munir (2007:15) “Bahasa pemrograman adalah bahasa yang dipakai oleh programmer untuk menulis kumpulan instruksi-instruksi program kedalam komputer dengan kata lain bahasa pemrograman merupakan sarana kedalam komputer yang menjembatani antara manusia dan komputer”.

Menurut tingkatannya bahasa pemrograman dibagi menjadi tiga tingkatan yaitu:

1. Bahasa Pemrograman Tingkat Rendah (*Low level language*).

Merupakan bahasa pemrograman generasi pertama, bahasa pemrograman ini sangat sulit untuk dimengerti karena instruksinya menggunakan bahasa mesin. Biasanya yang mengerti hanyalah pembuatnya saja isi programnya berupa kode-kode mesin.

2. Bahasa Pemrograman Tingkat Menengah (*Middle level language*)

Merupakan bahasa pemrograman dimana pengguna instruksi sudah mendekati bahasa sehari-hari, walaupun masih sulit dimengerti karena banyak menggunakan singkatan-singkatan seperti “STO” artinya bahasa (“*STORE*”) dan “MOV” artinya pindahkan (“*MOVE*”). Yang tergolong bahasa ini adalah *assembler*.

3. Bahasa Pemrograman Tingkat Tinggi (*High level language*)

Merupakan bahasa yang memiliki ciri lebih terstruktur, mudah dimengerti karena menggunakan bahasa sehari-hari, contoh bahasa level ini adalah *Delphi, Pascal, ORACLE, MS.SQL, Per, Phython, Basic, Visual Studio (Visual Basic, Visual Fox Pro)* dll. Seperti bahasa *Java, PHP, ASP,XML*, biasanya digunakan untuk pemrograman internet.

B. Bahasa Pemrograman

Bahasa pemrograman, atau sering diistilahkan juga dengan bahasa komputer atau bahasa pemrograman komputer, adalah instruksi standar untuk memerintah komputer. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi. Bahasa program komputer yang kita kenal antara lain adalah:

Visual dBase, Visual FoxPro, Delphi, Visual C, Visual Basic. Salah satu tahap diri pengembangan suatu program adalah menerjemahkan atau mengkodekan rancangan terinci yang telah dibuat menjadi suatu program komputer yang siap pakai. Dengan menerjemahkan berarti kita melakukan penulisan program dengan menggunakan bahasa komputer yang kita kuasai. Sedangkan untuk pembuatan program dalam Tugas Akhir ini dibuat dengan menggunakan bahasa pemrograman *Microsoft Visual Basic 6.0*.

1. Microsoft Visual Basic 6.0

Menurut Mangkulo (2011:1) menyatakan bahwa:

Visual basic 6.0 merupakan salah satu bahasa pemrograman visual. Dengan *Visual basic 6.0* programmer dapat dengan mudah untuk membuat suatu program aplikasi. Walaupun kemudahan diberikan dalam pembuatan program aplikasi, tetapi program aplikasi yang dihasilkan juga baik. Ini disebabkan dalam pengembangan program aplikasi *Visual basic 6.0* didukung oleh banyak fasilitas. Dalam membangun sebuah program aplikasi database dengan *visual basic 6.0*,

anda dapat melakukannya dengan mudah sesuai dengan keinginan anda. Dengan adanya kontrol-kontrol *activex* yang mudah untuk digunakan, membuat anda lebih mudah lagi dalam membuat program aplikasi *database*.

a. IDE (*Integrated Development Environment*) *Visual Basic 6*

Untuk dapat menggunakan *Visual basic 6.0* anda harus mengetahui IDE (*Integrated Development Environment*) atau lingkungan kerja dari *Visual Basic 6.0* itu sendiri. IDE pada *Visual Basic 6.0* dibagi menjadi delapan bagian besar, yaitu *menu*, *toolbar*, *toolbox*, *project explorer*, *properties windows*, *form layout*, *form* dan *kode editor*.

b. *Menu*

Pada bagian menu terdapat tiga belas menu utama, yaitu *menu file*, *edit*, *view*, *project*, *format*, *debug*, *run*, *query*, *diagram*, *tools*, *add-ins*, *windows* dan *help*.

Untuk menggunakannya yaitu tinggal mengklik pada menu utama kemudian memilih pada sub menu.

c. *Toolbar*

Toolbar fungsinya sama seperti fungsi dari menu, hanya saja pada *toolbar* pilihan-pilihan berbentuk ikon. Untuk memilih suatu proses yang akan dilakukan. *Icon-icon* pada *toolbar* adalah pilihan-pilihan pada menu yang sering digunakan dalam membuat program aplikasi. Dengan adanya *toolbar*, memudahkan untuk memilih proses yang sering dilakukan tanpa harus memilihnya pada menu.

d. *Toolbox*

Toolbox adalah tempat dimana kontrol-kontrol diletakan. Kontrol-kontrol yang terdapat pada *toolbox* dipakai dalam pembuatan aplikasi. Untuk membuat objek kontrol pada form program aplikasi diambil dari kontrol-kontrol yang ada pada *toolbox*.

e. *Project explorer*

Project explorer adalah tempat untuk melihat daftar dari *form* dan *module* yang digunakan dalam proyek melalui *project explorer*.

f. *Properties window*

Properties window adalah tempat untuk properti dari setiap objek kontrol. *Properties window* juga dipakai untuk mengatur properti dari objek kontrol yang dipakai.

g. *Form layout window*

Form layout window berfungsi untuk melihat posisi form pada layar monitor pada waktu program dieksekusi. Untuk menggeser posisi *form*, klik dan geser posisi *form* pada *form layout window* sesuai dengan posisi yang diinginkan pada layar monitor.

h. *Form*

Form adalah tempat untuk membuat tampilah (*user interface*) untuk program aplikasi. Pada *form*, dapat meletakkan atau menambahkan objek kontrol.

i. *Kode editor*

Kode editor adalah tempat melakukan atau menuliskan kode program dari program aplikasi yang dibuat.

Untuk menggunakan *kode editor*, dapat melakukannya dengan lima cara. Cara-cara tersebut adalah sebagai berikut:

- 1) Klik menu *view* lalu klik kode.
- 2) Klik ganda pada *form* atau *objek control* yang terdapat pada *form*.
- 3) Klik kanan pada *form* lalu klik *view code*.
- 4) Klik kana pada *project exploler* lalu klik *view code*.
- 5) Klik pada *icon view code* pada *project exploler*.

C. **Basis Data**

Menurut Yakub (2012:55) menyatakan bahawa:

“Basis data merupakan koleksi dari data-data yang terorganisi dengan cara sedemikian rupa sehingga data tersebut mudah disimpan dan dimanipulasi.

Sebuah sistem basis data dapat memiliki beberapa basis data. Setiap basis data dapat memiliki sejumlah objek basis data (seperti tabel, indeks, dan lain-lain). Disamping berisi atau menyimpan data, setiap basis data juga mengandung/menyimpan definisi struktur“.

Menurut Sukanto dan M.Shalahudin (2015:43) “ Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan “. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

1. DBMS (*Data Base Management System*)

DBMS (Database Management System) atau dalam Bahasa Indonesia sering disebut sebagai sistem manajemen basis data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola dan menampilkan data. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut:

- 1) Menyediakan fasilitas untuk mengelola akses data
- 2) Mampu menangani integritas data
- 3) Mampu menangani akses data yang dilakukan secara
- 4) Mampu menangani backup data.

Karena pentingnya data bagi suatu organisasi/perusahaan, maka hampir sebagian besar perusahaan memanfaatkan DBMS dalam mengelola data yang mereka miliki. Pengelolaan DBMS sendiri biasanya ditangani oleh tenaga ahli yang spesialis menangani DBMS yang disebut sebagai *DBA (Database Administrator)*.

2. *Crystal Report*

Crystal Report menurut Kursini (2007:264), sebagai berikut: “*Crystal Report* merupakan program yang dapat digunakan untuk membuat, menganalisis dan menterjemahkan informasi yang terkandung dalam database atau program ke dalam berbagai jenis laporan yang sangat fleksibel.

Crystal Report adalah salah satu peranti lunak yang digunakan khusus untuk membuat laporan. Berbagai fasilitas yang dimilikinya memudahkan dan juga memperindah laporan yang kita buat. *Crystal report* memiliki berbagai fungsi yang sangat memudahkan dalam pembuatan laporan seperti untuk membuat pengelompokan data laporan, melakukan proses matematis data laporan, formula dan fungsi untuk berbagai macam kebutuhan dan lain sebagainya. *Crystal report* sangat cocok sekali digunakan untuk membuat laporan yang datanya bersumber dari *database*.

3. *Microsoft Access*

Microsoft Access 2010 merupakan anggota dari aplikasi Microsoft Office 2010. Ms. Access menyimpan data dalam format sendiri berdasarkan Access Jet Database Engine. Hal ini dapat mengimpor atau link langsung ke data yang disimpan dalam aplikasi lain dan database seperti Excel, SharePoint lists, text, XML, Ms Outlook, HTML, dBase, Paradox, Lotus 1-2-3, ODBC compliant data kontainer, meliputi: Microsoft SQL Server, Oracle, MySQL, PostgreSQL. Pada Microsoft Access 2010 terdapat beberapa yang menarik dibanding Microsoft

Access versi sebelumnya. Sebelum memulainya, kita harus mengetahui komponen-komponen dalam database.

1. **Tables**, merupakan data sebagai komponen utama dalam database atau dengan kata lain tempat menyimpan data.
2. **Queries**, berfungsi menyaring data dari berbagai kriteria dan urutan yang dikehendaki. Query adalah permintaan yang diberikan oleh unsur untuk mengambil informasi yang tersimpan dalam database. Fungsi dari Query adalah untuk menyaring dan menampilkan data dari berbagai kriteria dan urutan yang kita kehendaki.
3. **Form**, berfungsi memasukkan data, menampilkan data, serta mengedit data dari suatu tabel dengan tampilan formulir yang telah kita rancang sendiri. Form digunakan untuk mempresentasikan ke user atau menerima inputan dari user data-data dalam tabel/query dalam bentuk interface grid, tombol, dan lain-lain kontrol windows.
4. **Reports**, berfungsi mencetak data dalam bentuk laporan.
5. **Pages**, berfungsi menciptakan halaman web berupa data access pages.
6. **Macros**, berfungsi mengotomatiskan perintah-perintah yang kita kehendaki dalam mengolah data.
7. **Modules**, berfungsi untuk merancang bermacam-macam modul aplikasi dalam mengolah database tingkat lanjut sesuai yang kita kehendaki.

D. Model Pengembangan Perangkat Lunak

Teknik pengembangan perangkat lunak menggunakan metode incremental yang sudah dikembangkan dari waterfall model, karena metode incremental ini terdiri dari tahap-tahap yang memberikan kemudahan, jika pada suatu tahap tidak sesuai atau mengalami kesalahan maka dapat kembali ketahap sebelumnya.

Dengan berkembangnya perangkat lunak, pembuat program (*programmer*) secara umum langsung pembuatan pengkodean perangkat lunak, tanpa menggunakan prosedur perangkat lunak.

SDLC (*Software Development Life Cycle*) diperkenalkan pada tahun 1960-an, SDLC pertama digunakan oleh para pengusaha besar jaman itu. Dimana sistem-sistem yang dibangun untuk mengelolah informasi kegiatan dan aktivitas yang berpotensi memiliki data yang besar.

Menurut (sommerville, 2007:66) “Model air terjun (*Waterfall*) metode yang digunakan pada pengembangan perangkat lunak terbagi menjadi tiga tahapan yaitu:

1. Analisa kebutuhan

Pada metode ini dari segi pengguna adalah admin, dimana admin dapat mengakses login admin serta menambahkan, mengedit, menghapus, melihat daftar rekam medis rawat jalan dan juga bisa mencetak laporan hasil kunjungan pasien rawat jalan.

2. Perancangan sistem dan perangkat lunak

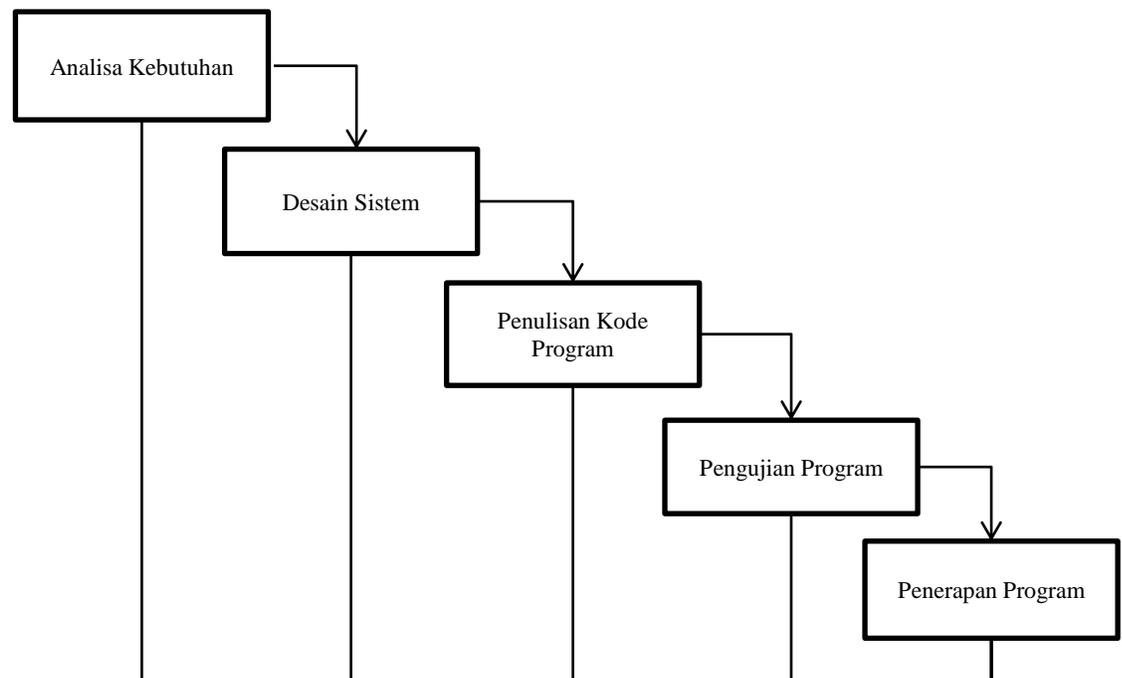
Proses perancangan sistem ini difokuskan pada beberapa atribut, yaitu LRS (*Logical Record Structure*) dan ERD (*Entity Relationship Diagram*). Penulis

juga menggunakan perangkat lunak Microsoft Visual Basic untuk melakukan *coding*.

3. Implementasi dan pengujian unit

Pada tahap ini dilakukan kerja untuk menciptakan perangkat lunak berdasarkan analisa dan perancangan yang telah dilakukan dengan melakukan pengkodean. Sehingga hasil dari tahap ini adalah basis data dan *source code* perangkat lunak. Hasil dari fase implementasi akan menjadi input pada fase pengujian dan perawatan metode pengujian yang digunakan pada pengembangan aplikasi ini menggunakan *blackbox testing*.

Berikut adalah gambar model air terjun:



Sumber: Sommerville (2007:66)

Gambar II.1. Diagram Metode *Waterfall*

Penjelasan:

1. Analisa Kebutuhan adalah kondisi yang harus dipenuhi dalam suatu produk baru atau perubahan produk, yang mempertimbangkan berbagai kebutuhan yang bersinggungan antar berbagai pemangku kebutuhan.
2. Batasan sistem adalah batasan yang diberikan kepada sistem, agar sistem tersebut tidak menyimpang dari prosedur yang diinginkan.
3. Penulisan Kode Program adalah desain program yang diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan.
4. Pengujian Program adalah penyatuan unit-unit program kemudian diuji secara keseluruhan (*system testing*).
5. Penerapan Program adalah mengoperasikan program dilingkungannya dan melakukan penyesuaian atau perubahan adaptasi dengan situasi sebenarnya.

2.2. Tools Program

Dalam perancangan suatu sistem diperlukan alat bantu. Alat bantu ini merupakan *representasi* grafik yang dapat mempermudah dalam menggambarkan komponen-komponen yang ada, proses yang terjadi dan membuat usulan pemecahan masalah secara logika.

A. Enterprise Relationship Diagram (ERD)

1. Pengertian ERD

Menurut Sukanto dan M. Shalahudin (2015:50), ERD (*Entity Relationship Diagram*) adalah dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika

penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD.

2. Komponen ERD

Menurut Sukamto dan M.Shalahudin (2015:50) di dalam ERD memiliki beberapa notasi dari berbagai aliran, dan salah satu aliran sering dipakai yaitu notasi Peter Chen. Berikut adalah simbol-simbol yang digunakan pada ERD:

1) Entitas (*Entity*)

Adalah data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer.

2) Atribut

Field atau kolom data yang butuh disimpan dalam suatu entitas.

3) Atribut kunci primer

Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan.

4) Atribut Multinilai (*Multivalue*)

Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.

5) Relasi

Adalah relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.

6) Asosiasi

adalah penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaiannya maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas.

2. LRS (*Logical Record Structure*)

Menurut Simarmata (2008:12) "*LRS (Logical Record Structure)* adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas. Menentukan kardinalitas, jumlah tab el dan *foreign key*." *Logical Record Structure* dibentuk dengan nomor dan tipe *record*. Beberapa tipe *record* digambarkan oleh kotak persegi panjang dan dengan nama unik. Beda *LRS* dan *ERD* nama dan tipe *record* berada di luar kotak field tipe *record* ditempatkan. *Logical Record Structure* terdiri dari link-link diantara tipe *record*. Link ini menunjukkan arah dari satu tipe *record* lainnya. Banyak link dari *LRS* yang diberi tanda *field-field* yang kelihatan pada kedua link tipe *record*. Penggambaran *LRS* mulai dengan model yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke *LRS*. Metode lain yang di mulai dengan *ERD* dan langsung dikonversikan ke *LRS*.

B. Pengkodean

Menurut Mustakini (2014:384) menyimpulkan bahwa:

“Kode digunakan untuk tujuan mengklasifikasikan data, memasukan data ke dalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengan data tersebut. Kode dapat dibentuk dalam kumpulan angka, huruf dan karakter-karakter khusus. Angka merupakan simbol yang banyak digunakan pada sistem kode”.

Di dalam merancang suatu kode harus diperhatikan beberapa hal, yaitu sebagai berikut:

1. Harus mudah diingat

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

2. Harus unik

Kode harus unik, untuk masing-masing *item* yang diwakilinya. Unik berarti tidak ada kode yang kembar.

3. Harus fleksibel

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

4. Harus efisien

Kode harus sependek mungkin, selain mudah diingat juga akan efisien bila direkam dan disimpan diluar komputer.

5. Harus konsisten

Bilamana mungkin, kode harus konsisten dengan kode yang telah dipergunakan.

6. 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 pemakain bagi yang menggunakan kode tersebut.

7. Spasi dihindari

Spasi di dalam kode sebaiknya dihindari, karena dapat menyebabkan kesalahan didalam menggunakannya.

8. Hindari karakter yang mirip

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

9. Panjang harus sama

Masing-masing kode yang sejenis harus mempunyai panjang yang sama.

Ada beberapa macam tipe dari kode yang dapat digunakan di dalam system informasi, antara lain:

a. Kode Mnemonik

Kode Mnemonik (*mnemonic code*) digunakan untuk tujuan supaya mudah diingat. Kode mnemonik dibuat dengan dasar singkatan atau mengambil sebagian karakter dari item yang akan diwakili dengan kode ini. Misalnya kode “P” untuk mewakili pria dan kode “W” untuk mewakili wanita akan mudah diingat. Contoh lainnya adalah kode “YG” untuk Yogyakarta. “SM”

untuk kode Semarang. Kebaikan kode ini adalah mudah diingat dan kelemahannya adalah kode dapat menjadi terlalu panjang.

b. Kode Urut

Kode Urut (*Sequential Code*) disebut juga dengan kode seri (*serial kode*) merupakan kode yang nilainya antara satu kode dengan kode berikutnya.

Kebaikannya:

- 1) Sangat sederhana.
- 2) Mudah diterapkan.
- 3) Kode dapat pendek tetapi baik.
- 4) Mudah dicari bila kodenya sudah diketahui.
- 5) Cocok untuk rekaman di *file* yang menggunakan nomor *record* relatif sehingga nomor *record* sama dengan kodenya. Dengan demikian *file* tidak perlu di indeks.
- 6) Baik untuk pengendalian, karena kode yang hilang dapat diketahui.

Kelemahannya:

- 1) Penambahan kode hanya ditambahkan pada akhir urutan dan tidak dapat disisipkan.
- 2) Tidak mempunyai dasar logika tentang informasi item yang diwakilinya, kecuali hanya berdasarkan urutannya saja.
- 3) Tidak fleksibel bila terjadi perubahan kode.

c. Kode Blok Urut

Kode Blok Urut (*Block Code*) mengklasifikasikan *item* kedalam kelompok blok tertentu yang mencerminkan satu klasifikasi tertentu atas dasar pemakaian maksimum yang diharapkan.

Kebaikannya:

- 1) Nilai dari kode mempunyai arti, yaitu masuk dalam blok yang sudah tertentu.
- 2) Mudah diperluas
- 3) Kode dapat ditambah atau dibuang sebagian.

Kelemahannya:

- 1) Panjang kode tergantung dari jumlah bloknya, akibatnya kode menjadi cukup panjang.
- 2) Kurang mudah diingat

d. Kode Group

Kode Group (*Group Code*) merupakan kode yang berdasarkan *field-field* dan tiap-tiap *field* kode mempunyai arti.

Kebaikannya:

- 1) Nilai dari kode mempunyai arti.
- 2) Mudah diperluas
- 3) Dapat ditambah atau dibuang sebagian
- 4) Dapat menunjukkan jenjang dari data.

Kelemahannya:

1) Kode dapat menjadi panjang

e. Kode Decimal

Kode Decimal (*Decimal Kode*) mengklasifikasikan kode atas dasar sepuluh unit angka decimal dimulai dari angka nol sampai dengan sembilan atau nol-nol sampai dengan sembilan-sembilan tergantung dan banyaknya kelompok.

C. HIPO (*Hierarchy Input Proses Output*)

HIPO merupakan alat dokumentasi program yang berdasarkan fungsinya untuk meningkatkan efisiensi usaha perawatan program. Dokumen ini dilaksanakan dengan mempercepat lokasi dalam kode pada fungsi program yang akan dimodifikasi. Atau dapat dikatakan bahwa HIPO dikembangkan agar tersedia suatu teknik untuk mendokumentasikan fungsi program. Pembentukan HIPO ini dilakukan pada tahap pengembangan sistem informasi.

1. Pengertian HIPO

Menurut Yakub (2012:156) menyatakan bahwa “ bagian berjenjang (*hierary cahrt*) adalah diagram yang digunakan untuk mempersiapkan penggambaran diagram arus data ke level-level bawah lagi. Bagian bejenjang dapat digambarkan dengan menggunakan notasi proses pada diagram arus data (*data flow diagram*).

2. Tingkatan Diagram HIPO

Diagram HIPO didesain untuk menyediakan dokumentasi dari setiap tingkatan, ada tiga tingkatan diagram yang tercakup, yaitu diagram daftar isi, diagram ringkas dan diagram terperinci.

1) Diagram daftar isi

Pada diagram ini sedapat mungkin secara ringkas menjelaskan fungsi system dengan cara memberikan indeks pemberian sistem.

2) Diagram ringkas

Diagram ringkas menunjukkan secara umum input atau output dengan kegiatan pemrosesan yang akan menjelaskan masing-masing fungsi pokok. Hubungan antar fungsi ditunjukkan dengan tanda anak panah. Kadang-kadang tanda ini belum mampu menjelaskan bagaimana cara kerja suatu langkah khusus tertentu sehingga masih perlu diberikan catatan tertulis pada lembaran yang disebut lembar pemerian lanjutan.

3) Diagram terperinci

Pada diagram ini melukiskan secara lengkap fungsi khusus dengan menyajikan sebanyak mungkin diagram ditambah dengan catatan jauh yang diperlukan. Perinciannya ditunjukkan untuk memudahkan pemrograman mengambil informasi sesuai dengan tata letak laporan.

D. Diagram Alir Program (*Flowchart*)

1. Pengertian *Flowchart*

Menurut Yakub (2012:162) “ Bagian alir (*flowchart*) adalah bagan yang menggambarkan urutan instruksi proses dan hubungan satu proses dengan proses lainnya menggunakan simbol-simbol tertentu. Bagan alir digunakan sebagai alat bantu komunikasi dan dokumentasi “. Pada saat akan menggambarkan suatu bagan alir analisis sistem atau programmer dapat mengikuti pedoman-pedoman sebagai berikut:

- a. Bagan alir sebaiknya digambarkan dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman
- b. Kegiatan di dalam bagan alir harus ditunjukkan dengan jelas
- c. Harus ditunjukkan dimana kegiatan akan dimulai dan dimana akan berakhirnya,
- d. Masing-masing kegiatan di dalam alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan
- e. Masing-masing kegiatan di dalam bagan alir harus dalam urutan yang semestinya.
- f. Kegiatan yang terpotong dan akan disambung di tempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
- g. Gunakan bagan alir yang standar.

2. Bentuk *flowchart*

Dalam suatu pembuatan program komputer terdapat dua jenis *flowchart* yang sering digunakan yaitu:

a. Program *flowchart*

Merupakan simbol-simbol yang menggambarkan proses secara terperinci dan detail antara instruksi yang satu dengan yang lainnya didalam suatu program komputer yang bersifat *logic* dan sistematis.

b. Sistem *flowchart*

Merupakan simbol-simbol yang menggambarkan urutan prosedur secara detail didalam suatu sistem komputerisasi dan yang digunakan untuk proses pengolahan data serta menggambarkan hubungan antara peralatan tersebut yang bersifat fisik.

3. Tehnik Pembuatan

a. *General Way*

Tehnik pembuatan *flowchart* dengan cara ini digunakan dalam menyusun logika suatu program, yang menggunakan proses secara tidak langsung (*Non-Direct-Loop*).

b. *Iteration Way*

Tehnik pembuatan *flowchart* dengan cara ini biasanya dipakai untuk logika program yang cepat dan juga bentuk permasalahan yang kompleks.