

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Program

Dalam bahasa sebenarnya program adalah pernyataan yang disusun menjadi kesatuan prosedur yang berupa urutan langkah yang disusun secara logis dan sistematis untuk menyelesaikan masalah. Secara umum bahasa pemrograman komputer dikelompokkan menjadi 4 kategori, yaitu bahasa pemrograman rendah (*low level language*), bahasa pemrograman tingkat menengah (*middle level language*), dan bahasa pemrograman tingkat tinggi (*high level language*). Dengan bahasa pemrograman komputer, manusia dapat membuat sebuah program yang dimanfaatkan untuk membantu menyelesaikan masalah.

Menurut Riana (2008:4) “Program merupakan susunan perintah yang diberikan kepada komputer untuk melakukan sesuatu”. Beberapa definisi menurut Yusono (2008:8), diantaranya:

1. Pemrograman merupakan proses mengimplementasikan urutan langkah untuk menyelesaikan suatu masalah dengan menggunakan suatu bahasa pemrograman.
2. Bahasa Pemrograman merupakan prosedur atau tata cara penulisan program. Pada bahasa pemrograman terdapat dua faktor penting, yaitu sintak dan semantik. Sintak adalah suatu aturan-aturan gramatikal yang mengatur tata cara penulisan kata, ekspresi dan pernyataan. Semantik adalah aturan-aturan untuk menyatakan suatu arti.

2.1.1. Aplikasi

software atau perangkat lunak komputer yang dibuat untuk melakukan tugas tertentu. Jika sistem operasi komputer (misalnya *Windows*) berfungsi untuk melakukan operasi dasar, program aplikasi tertentu bisa kita tambahkan (*install*) untuk melengkapi kemampuan sistem operasi komputer untuk melakukan tugas-tugas yang lebih spesifik.

Menurut Jogiyanto (2007:12) dalam bukunya menyatakan bahwa “penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*”.

Menurut Kamus Besar Bahasa Indonesia (2008:52) dalam bukunya menyatakan bahwa “penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu”.

Dari kutipan diatas dapat disimpulkan bahwa aplikasi merupakan software atau perangkat lunak komputer yang dibuat untuk melakukan tugas tertentu yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output.

2.1.2. Simpan Pinjam

Simpan pinjam adalah simpanan yang dikumpulkan bersama dan pinjamkan kepada anggota yang memerlukan pinjaman dalam berbagai usaha dimana anggota mengajukan permohonan tertulis kepada pengurus dengan mencantumkan jumlah uang yang diperlukan kemudian pengurus mempertimbangkan dan memutuskan permohonan pinjaman sesuai dengan kemampuan koperasi pada saat itu dimana

pengurus berhak menentukan besarnya jumlah pinjaman, syarat-syarat pengembalian, dan bentuk nilai.

Simpan Pinjam menurut Kamus Besar Bahasa Indonesia (2017) menyatakan bahwa “simpanan yang dikumpulkan bersama dan dipinjamkan kepada anggota yang memerlukan pinjaman dalam berbagai usaha”.

Menurut kasmir (2008:286) “koperasi adalah suatu kumpulan dari orang-orang yang mempunyai tujuan dan kepentingan bersama.

Dari kutipan di atas penulis simpulkan bahwa simpanan yang dikumpulkan bersama dari orang-orang yang mempunyai tujuan dan kepentingan bersama dan dipinjamkan kepada anggota yang memerlukan pinjaman dalam berbagai usaha.

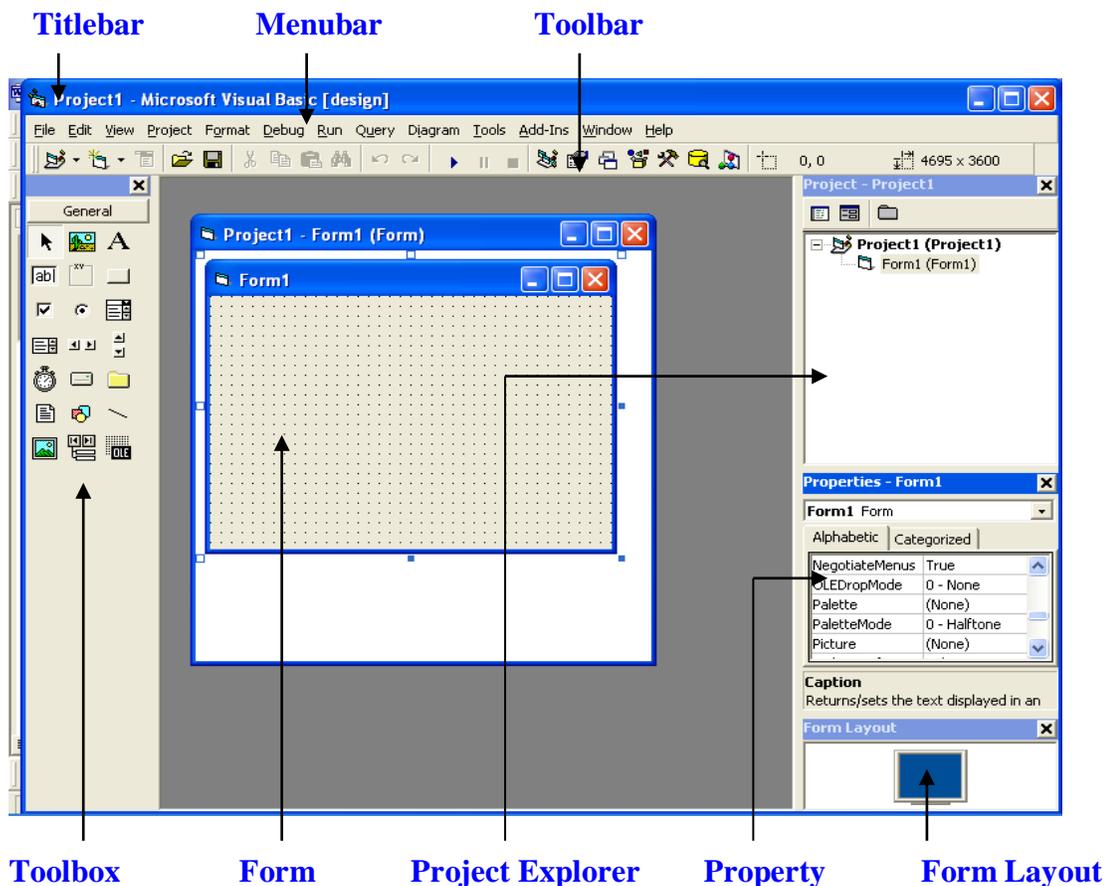
2.1.3. Microsoft Visual Basic 6.0

Visual Basic (VB) atau RAD (*Rapid Application Development*), yang memungkinkan *programmer* untuk membuat aplikasi Windows dalam waktu yang sangat sedikit. Ini adalah bahasa pemrograman yang paling populer di dunia, dan memiliki programmer lebih dan baris kode dari pada pesaingnya terdekat.

Menurut Koswara (2011:9) dalam jurnal media infotama “visual basic adalah sebuah bahasa pemrograman untuk membuat aplikasi visual berbasis sistem operasi Microsoft Windows”. Sedangkan menurut Kurniadi (2008:5) “Visual Basic adalah sebuah sarana pembuat program yang lengkap namun mudah. Siapapun yang bisa menggunakan windows, ia pasti bisa membuat program dengan visual basic”.

IDE visual basic terdiri atas beberapa jendela yang masing-masing memiliki peran tersendiri. Berikut ini adalah penjelasan komponen-komponen IDE visual basic menurut Kusriani dan Koniyo (2007:172) :

Dari kutipan diatas penulis menyimpulkan bahwa visual basic adalah sebuah bahasa pemrograman untuk membuat aplikasi visual berbasis sistem operasi Microsoft Windows dan bahasa pemograman yang paling populer didunia.



Sumber : Kusrini dan Kuniyo (2007: 172)

Gambar II.1. Jendela IDE Visual Basic 6.0

1. Menu Bar

Pada Bagian menu terdapat menampilkan menu yang berisi perintah-perintah pada Visual Basic, yaitu menu *File, Edit, View, Project, Format, Debug, Run, Query, Diagram, Tools, Add-Ins, Windows* dan *Help*.

2. *Toolbar*

Menu *Toolbar* merupakan menu berbentuk icon yang berisi perintah. *Toolbar* disediakan Visual Basic untuk mengakses berbagai fungsi yang ada dalam menu secara lebih cepat dan lebih mudah.

3. *Toolbox*

Toolbox adalah tempat dimana kontrol-kontrol diletakan. Kontrol-kontrol yang terdapat pada *toolbox* dipakai dalam pembuatan program aplikasi.

4. *Project Explorer*

Project Explorer adalah tempat untuk melihat daftar form dan modul yang digunakan dalam proyek. Melalui *Project Explorer* kita juga dapat memilih form yang akan dipakai.

5. *Property Window*

Property Window adalah tempat properti setiap objek kontrol yang digunakan untuk mengatur properti dari objek kontrol yang dipakai.

6. *Form Layout Window*

Form Layout Window berfungsi untuk melihat posisi form pada layar monitor pada waktu program di eksekusi.

7. *Form*

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

8. Kode Editor

Kode Editor adalah tempat dimana anda meletakkan atau menuliskan kode program dari program aplikasi anda.

Dapat Penulis simpulkan Visual Basic adalah sebuah aplikasi yang dilengkapi dengan *Interated Development Environmet* (IDE) dan komponen-komponen yang diperlukan untuk membuat dan menulis *listing* sebuah program komputer, agar dapat dimengerti dan dibaca oleh komputer yang berbasis windows.

2.1.4. Basis Data

Database atau basis data adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi.

Menurut Simarmata dan Paryudi (2010:1) dalam bukunya menyatakan bahwa “Basis data adalah mekanisme yang digunakan untuk menyimpan informasi atau data”. Sedangkan Kustiyahningsih (2010:145) dalam bukunya menyatakan bahwa “*database* atau basis data adalah sekumpulan informasi yang diatur agar mudah dicari”.

Dari kutipan diatas dapat disimpulkan bahwa *database* atau basis data merupakan suatu kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah dan diatur agar mudah dicari

2.1.5. MySQL

MySQL (*My Structure Query Language*) juga sangat diperlukan pemakai untuk mengakses data dalam *database* dan juga bisa menghubungkan *script* PHP (*HyperText Preprocessor*) menggunakan perintah *query*. MySQL (*My Structure Query Language*) sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*).

Menurut Sibero (2013:97) menyatakan bahwa “MySQL suatu RDBMS (*Relational Database Management System*) yaitu aplikasi sistem yang menjalankan fungsi pengolahan data”. Sementara Prasetio (2008:2) mengemukakan bahwa “MySQL adalah sebuah server *database Open Source* yang kayaknya paling populer keberadaanya”.

Dari kutipan di atas dapat penulis simpulkan bahwa MySQL RDBMS (*Relational Database Management System*) dimana aplikasi tersebut berfungsi untuk menjalankan fungsi-fungsi mengolah data yang sangat populer dan merupakan server yang *open source*

2.1.6. *Structure Query Language (SQL)*

SQL *Structured Query language (SQL)* sangat diperlukan pemakai untuk mengakses data dalam *database* relasional. SQL sebuah bahasa komputer yang mengikuti standar ANSI (*American Nasional Standard Institute*) yang digunakan dalam manajemen *database* relasional.

Menurut Deliana dkk (2009:6) dalam bukunya menyatakan bahwa “SQL (*Structure Query Language*) adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional”.

Menurut Rosa dan Shalahuddin (2013:28) dalam bukunya menyatakan bahwa “SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS”.

A. *Data Definition Language (DDL)*

Menurut Simarmata dan Paryudi (2010:110) dalam bukunya menyatakan bahwa “Bagian bahasa definisi data dari SQL memungkinkan tabel basisdata dibuat atau dihapus. Kita dapat pula membuat indeks, membuat hubungan antartabel, dan membuat batasan antartabel basisdata”.

Menurut Fathansyah (2015:18) dalam bukunya menyatakan bahwa “Struktur basisdata yang menggambarkan skema basisdata secara keseluruhan dan didesain dengan bahasa khusus”.

Menurut Simarmata dan Paryudi (2010:110) dalam bukunya menyatakan bahwa yang termasuk termasuk dalam *Data Definition Language* adalah:

1. *CREATE*

Untuk membuat, termasuk diantaranya membuat database dan tabel baru.

2. *ALTER*

Untuk mengubah struktur tabel yang telah di buat.

3. *DROP*

Untuk menghapus database dan tabel.

B. *Data Manipulation Language (DML)*

Menurut Fathansyah (2015:18) dalam bukunya menyatakan bahwa “Bahasa basisdata yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu basisdata”

Menurut Simarmata dan Paryudi (2010:110) dalam bukunya menyatakan bahwa yang termasuk termasuk dalam *Data Manipulation Language* adalah:

1. *SELECT*

Untuk mengambil data atau menampilkan data dari satu tabel atau beberapa tabel.

2. *UPDATE*

Untuk memperbaharui data lama menjadi data terkini.

3. *DELETE*

Untuk menghapus data dari tabel.

4. *INSERT*

Untuk menyisipkan atau memasukan dalam tabel.

C. *Data Control Language*

Menurut Al Fatta (2007:33) menyatakan bahwa “perintah yang fungsinya berperan untuk melakukan proses manipulasi user pada sebuah *database*, dan juga menyangkut hak akses dari sebuah *database*”.

1. *GRANT*

Perintah yang berhubungan dengan hak akses *user*. Dengan menggunakan perintah *grant* ini, maka *administrator* atau pemilik dari sebuah *server* dan juga *database* dapat menunjuk salah satu *user*, agar dapat memperoleh hak akses untuk melakukan manajemen dan juga pengaksesan terhadap *database*.

2. *REVOKE*

Kebalikan dari perintah *grant* sebelumnya. Apabila *grant* merupakan perintah yang digunakan oleh *administrator* untuk membuka hak akses kepada *user*, maka perintah *revoke* ini digunakan oleh *administrator* untuk menarik dan juga mencabut hak akses yang sudah diberikan sebelumnya

kepada *user*. Pencabutan hak akses ini bisa disebabkan oleh berbagai macam kondisi, tergantung keputusan dari *administrator*.

2.1.7. Database Managemen System (DBMS)

Untuk mengatasi kelemahan sistem yang berbasis berkas atau *file* dibutuhkan *Database Managemen System* (DBMS) yang merupakan sebuah sistem yang khusus dibuat untuk mengelola basis data.

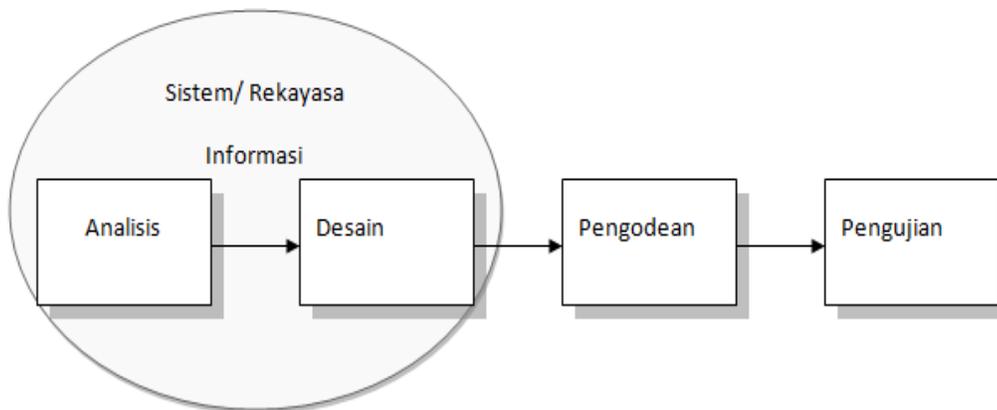
Menurut Simarmata dan Priyadi (2010:8) menyatakan bahwa “Sistem Manajemen Basis data (DBMS) adalah perangkat lunak yang didesain untuk membantu memelihara dan memanfaatkan kumpulan data yang besar”. Sedangkan menurut Rosa dan Shalahuddin (2015:44) mengemukakan bahwa “DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan mengelola dan menampilkan data”.

Sementara menurut Kadir dan Triwahyuni (2013:339) mengemukakan bahwa “DBMS(*Database Management System*) adalah sistem yang secara khusus yang dibuat untuk memudahkan pemakai dalam mengelola basis data”.

Dari beberapa pendapat di atas DBMS adalah aplikasi *software* yang dapat mengakses basis data untuk menyimpan, mengelola, dan menampilkan semua data.

2.1.8. Model Pengembangan Perangkat Lunak

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial *linier* atau alur hidup klasik. Model air terjun menyediakan alur hidup perangkat lunak secara sekuensial atau terurut menurut (Rosa dan Shalahuddin, 2013:28) dimulai dari:



Sumber: Rosa dan Shalahuddin (2015:29)

Gambar II.2. 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*.

2. Desain

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

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 secara logis dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan yang ada.

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

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

2.1.9. Crystal Report

Crystal Report awalnya dikembangkan oleh Crystal Report Inc, yang dalam perkembangannya sempat diakuisisi oleh Seagate dengan nama Seagate Crystal Report. Crystal Report kini dimiliki oleh SAP merupakan salah satu perusahaan software ERP terbesar di dunia.

Menurut Atmoko (2013:3) “ Crystal Report merupakan komponen yang akan kita gunakan untuk membuat *report* atau laporan dari program yang akan kita buat, agar dapat dipahami oleh pengguna, yang *report* tersebut diambil dari kumpulan data dari tabel yang tersimpan di dalam database SQL Server”.

Crystal Report adalah program pembuat laporan dari Seagate Corp yang dibuat untuk membantu user membuat laporan dengan mudah tanpa menggunakan Data Environment dan Data Report. Pada Crystal Report terdiri dari satu atau beberapa tabel, *query*, dan *report*.

2.2. Tools Pendukung

Dalam penulisan Tugas Akhir ini, penulis menggunakan peralatan pendukung (*tool System*) sebagai alat bantu dalam menyediakan Tugas Akhir ini, adapun peralatan yang digunakan, yaitu:

2.2.1. Entity Relationship Diagram (ERD)

Pengertian dari ERD (*Entity Relationship Diagram*) adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan simbol.

Menurut Sutanta (2011:91) “Entity Relationship Diagram (ERD) merupakan suatu model data yang dikembangkan berdasarkan objek.” Entity Relationship Diagram (ERD) digunakan untuk menjelaskan hubungan antar data dalam basis data

kepada pengguna secara logis. Entity Relationship Diagram (ERD) didasarkan pada suatu persepsi bahwa real world terdiri atas obyek-obyek dasar tersebut. Penggunaan Entity Relationship Diagram (ERD) relatif mudah dipahami, bahkan oleh para pengguna yang awam. Bagi perancang atau analis sistem, Entity Relationship Diagram (ERD) berguna untuk memodelkan sistem yang nantinya, basis data akan dikembangkan. Model ini juga membantu perancang atau analis sistem pada saat melakukan analisis dan perancangan basis data karena model ini dapat menunjukkan macam data yang dibutuhkan dan kerelasiaan antardata didalamnya

Sedangkan Menurut Mata-Toledo dan Cushman (2007 : 139) Mendefinisikan "Entity Relationship Diagram (ERD) merupakan representasi grafis dari logika database dengan menyertakan deskripsi detail mengenai seluruh entitas (*entity*), hubungan (*relationship*), dan batasan (*constraint*)."

Masih dalam buku Edhy Sutanta (2011:91) menjabarkan Komponen Entity Relationship Diagram adalah sebagai berikut :

1. Entitas merupakan suatu objek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Objek dasar dapat berupa orang, benda, atau hal yang keterangannya perlu disimpan didalam basis data. Untuk menggambarkan sebuah entitas digunakan aturan sebagai berikut
 - a. Entitas dinyatakan dengan simbol persegi panjang.
 - b. Nama entitas dituliskan didalam simbol persegi panjang.
 - c. Nama entitas berupa kata benda, tunggal.

- d. Nama entitas sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.
2. Atribut Atribut merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan dalam basis data. Atribut berfungsi sebagai penjelas pada sebuah entitas. Untuk menggambarkan atribut digunakan aturan sebagai berikut:
 - a. Atribut digambarkan dengan simbol ellips.
 - b. Nama atribut dituliskan didalam simbol ellips.
 - c. Nama atribut merupakan kata benda, tunggal.
 - d. Nama atribut sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.
 3. Relasi Relasi merupakan hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Aturan penggambaran relasi adalah sebagai berikut :
 - a. Relasi dinyatakan dengan simbol belah ketupat.
 - b. Nama relasi dituliskan didalam simbol belah ketupat.
 - c. Nama relasi berupa kata kerja aktif.
 - d. Nama relasi sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

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 kaitanya dengan konvensi ke LRS”.

Menurut Riyanto (2005:22) Mendefinisikan bahwa “LRS (logical record strukture) adalah representasi dari struktur record-record pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas”. Logical record strukture dibentuk dengan nomor dari tipe record. Logical record strukture 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 menggunakan model yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke LRS. Metode yang lain dimulai dengan ER-diagram dan langsung dikonversikan ke LRS.

- a. One to One (1-1) Tingkat hubungan ini menunjukkan hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, dan hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.
- b. One to Many (1-M) Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu, tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya, satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

- c. Many to Many (M-M) Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, dilihat dari sisi entitas yang pertama maupun dilihat dari sisi yang kedua.

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 kaitanya dengan konvensi ke LRS”.

2.2.2. Diagram Alir Program (*Flowchart*)

Untuk menggambarkan sebuah algoritma yang terstruktur dan mudah dipahami oleh orang lain (khususnya *programmer* yang bertugas mengimplementasikan program), maka dibutuhkan alat bantu yang berbentuk diagram alir (*flowchart*).

Menurut Yakub (2012:162) menyatakan bahwa Bagan alir (*flowchart*) adalah bagan yang menggambarkan urutan instruksi proses dan hubungan satu proses lainnya menggunakan simbol-simbol tertentu. Bagan alir digunakan sebagai alat bantu komunikasi dan dokumentasi.

Menurut Riana (2008:4) menyatakan bahwa “Langkah-langkah penyelesaian masalah yang disusun secara sistematis disebut diagram alur atau bahasa inggris disebut *flowchart*”.

Berdasarkan pendapat diatas dapat disimpulkan Diagram Alir Program (*Flowchart*) bagan yang menggambarkan simbol-simbol tertentu bertujuan untuk penyelesaian masalah yang disusun secara sistematis.

2.2.3. HIPO (*Hierarchy Input Process Output*)

HIPO (*Hierarchy Input Process Output*) suatu teknik pendokumentasian program yang dapat digunakan untuk mengkomunikasikan spesifikasi sistem kepada para *programer* melalui proses perancangan. Setiap modul HIPO digambarkan oleh fungsi utamanya.

Menurut Amansyah (2008:284) menyatakan bahwa “HIPO dibuat sebagai alat untuk mendokumentasikan program, secara jelas memperagakan apa yang dikerjakan suatu program, data apa yang digunakan, dan keluaran yang dihasilkan”.

Menurut Praptiningsih (2010:3) menyatakan bahwa “HIPO (*Hierarchy Plus Input Process Output*) adalah alat bantu yang digunakan untuk membuat spesifikasi program yang merupakan struktur yang berisi diagram dimana didalam program ini berisi input yang diproses dan menghasilkan *output*”.

Berdasarkan pendapat diatas penulis menyimpulkan bahwa HIPO merupakan metodologi pendokumentasian program yang dapat digunakan untuk mengkomunikasikan spesifikasi sistem kepada programer secara jelas memperagakan apa yang dikerjakan suatu program dan didalamnya berisi input yang diproses dan menghasilkan output.

2.2.4. Pengkodean

Pengkodean digunakan untuk tujuan mengklasifikasikan data, memasukkan data ke dalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengannya.

Menurut Hartono (2007:384) dalam bukunya “Kode dapat dibentuk dari kumpulan angka, huruf dan karakter-karakter khusus (misalnya %, /, -, \$, #, &, :, dan sebagainya). Angka merupakan simbol yang banyak digunakan pada sistem pengkodean. Akan tetapi kode yang berbentuk angka lebih dari 6 digit akan sulit untuk diingat”.

Menurut Hartono (2007:384) menyatakan bahwa ada beberapa hal yang perlu diperhatikan dalam pengkodean yaitu:

1. Harus mudah diingat.

Supaya kode mudah diingat, maka dapat dilakukan dengan cara menghubungkan kode tersebut dengan obyek yang diwakilinya.

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 tetap dapat diwakili oleh kode

4. Harus efisien.

Kode harus sependek mungkin, selain mudah diingat juga akan efisien bila direkam di simpanan luar 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 pemakaian bagi yang menggunakan kode tersebut

7. Spasi dihindari.

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

8. Hindari karakter yang mirip.

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

9. Panjang kode harus sama.

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

Beberapa tipe dari kode :

a. Kode Urut (*Serial code*)

yaitu kode yang menggunakan nilai secara berurutan untuk item data.

b. Kode *Mnemoik*

yaitu kode yang digunakan dengan mengambil sebagian karakter di item data.

c. Kode Blok (*Block code*)

yaitu kode yang mengklasifikasikan item ke dalam kelompok blok tertentu yang mencerminkan satu klasifikasi tertentu.

d. Kode grup

yaitu kode yang berdasarkan *field* data, dengan tiap *field* mempunyai arti tertentu.

e. Kode desimal

yaitu kode yang mengklasifikasikan kode berdasarkan sepuluh unit angka desimal dari 0 sampai 12.