

BAB I

LANDASAN TEORI

1.1 Konsep Dasar Program

Dalam penulisan tugas akhir ini penulis tidak lepas dari teori-teori yang mendukung dan memberikan kemudahan dalam mempelajari dan merancang program aplikasi ini. Pada bab ini penulis mencoba mengemukakan beberapa teori yang didefinisikan oleh para ahli dan sumber. Namun teori-teori tersebut tidak penulis jelaskan secara keseluruhan, melainkan hanya bagian tertentu yang berkaitan dan mendukung dalam pembuatan program.

1.1.1 Program

Menurut (Sugiyono, 2014) mengemukakan bahwa :

Program merupakan sekumpulan instruksi yang diwujudkan dalam bentuk bahasa, kode skema, ataupun bentuk lain, yang dimengerti oleh komputer sehingga mampu bekerja untuk melakukan fungsi-fungsi khusus termasuk persiapan dalam merancang instruksi-instruksi tersebut.

1.1.2 Bahasa Pemrograman

Menurut (Kadir, 2017) mengemukakan bahwa :

Komputer bekerja atas dasar kode biner atau kode yang mempunyai dua keadaan berupa 0 dan 1, jika dinyatakan dalam keadaan lampu, kode 0 menyatakan keadaan padam dan kode 1 menyatakan keadaan menyala atas dasar inilah, program ada masa awal terciptanya komputer dibuat dengan menggunakan bahasa pemrograman yang berbasis pada kode biner dan dinamakan bahasa mesin.

Bahasa Pemrograman adalah bahasa perantara yang digunakan oleh manusia untuk memerintahkan sesuatu perintah kepada komputer dengan menggunakan intruksi-intruksi atau source code (Yuli Endra, 2015). Secara umum bahasa pemrograman dibagi menjadi 3 tingkatan yaitu:

- a. Bahasa Tingkat Tinggi (*High Language Program*), kenapa dikatakan bahasa tingkat tinggi karena bahasa pemrograman ini sudah mendekati seperti bahasa manusia. Contohnya bahasa pemrograman JAVA, PHP, dan lain-lain.
- b. Bahasa Tingkat Menengah (*Medium Language Program*), kenapa dikatakan bahasa tingkat menengah, bahasa ini bisa masuk kedalam bahasa tingkat tinggi ataupun bahasa tingkat rendah. Contohnya bahasa pemrograman C, Fortran atau Pascal.
- c. Bahasa Tingkat Rendah (*Low Language Program*), kenapa dikatakan bahasa tingkat rendah, karena bahasa ini sangat jauh berbeda dengan bahasa manusia. Contohnya bahasa Assembly atau bahasa mesin.

Dalam pembuatan program dalam Tugas Akhir ini penulis membuat kode program dengan menggunakan bahasa program Java dan menggunakan aplikasi NetBeans IDE 8.2.

1. Java

Java adalah bahasa pemrograman yang powerful dan serbaguna untuk pengembangan perangkat lunak yang berjalan di perangkat seluler, komputer dekstop, dan server (Enterprise, 2017). Java bersifat *Write Once, Run Anywhere* (Program yang ditulis satu kali dan dapat berjalan pada banyak platform) (Enterprise, 2015).

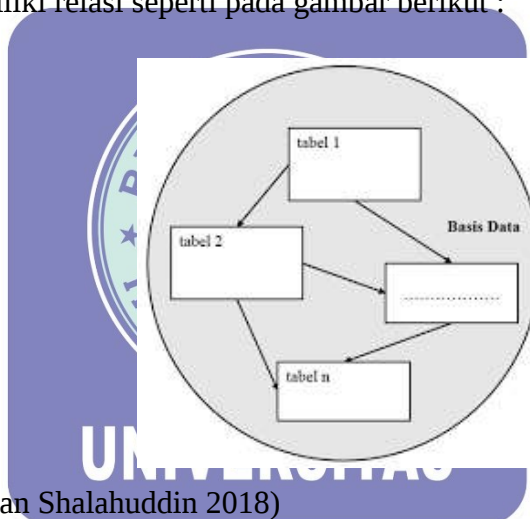
2. NetBeans

NetBeans IDE (*Integrated Development Environment*) untuk membuat aplikasi dengan Java, PHP, C, C++, dan HTML5 (Enterprise, 2015). NetBeans dimulai pada tahun 1996 sebagai Xelfi yang merupakan proyek IDE pada Universitas Charles di Praha. Pada tahun 1997, Roman Stanek mengkomersialkan IDE NetBeans dan

kemudian dibeli Oleh Sun Microsystem pada tahun 1999. Pada tahun 2010 Sun (dan NetBeans) diakuisisi oleh Oracle.

1.1.3 Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan (Rosa A.S, 2018). Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Berikut adalah ilustrasi gambar basis data relasional yang diimplementasikan dengan tabel-tabel yang saling memiliki relasi seperti pada gambar berikut :



Sumber : (Rosa dan Shalahuddin 2018)

Gambar II.1

Ilustrasi Basis Data

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa *file* teks ataupun *Database Management System* (DBMS). Kebutuhan basis data dalam sistem informasi meliputi :

- a. Memasukkan, menyimpan, dan mengambil data
- b. Membuat laporan berdasarkan data yang telah disimpan

Tujuan dari dibuatnya tabel-tabel disini adalah untuk menyimpan data ke dalam tabel-tabel agar mudah diakses. Oleh karena itu, untuk merancang tabel-tabel yang akan dibuat maka dibutuhkan pola pikir penyimpanan data nantinya jika dalam bentuk baris-baris data (*record*) dimana setiap baris terdiri dari beberapa kolom.

1. XAMPP

XAMPP merupakan program paket PHP dan MySQL berbasis *open source* yang saat ini merupakan andalan para programmer PHP dalam melakukan programming dan melakukan testing hasil programnya (Yudhanto, 2014).

Komponen dan fungsi XAMPP menurut (Setiawan, 2019) terdiri atas :

- a. Htdoc : adalah folder untuk meletakkan file yang akan dijalankan seperti, file PHP, HTML dan script lainnya.
- b. PhpMyAdmin : adalah bagian untuk mengelola database MySQL yang di komputer.
- c. Control Panel : yang berfungsi untuk mengelola layanan (*service*) XAMPP, seperti stop *service* (berhenti), atau start (mulai).

2. MySQL

Menurut (Madiun, 2016) mengungkapkan bahwa: “MySQL adalah sistem manajemen database SQL yang bersifat Open Source dan paling populer saat ini”. Sistem Database MySQL mendukung beberapa fitur seperti multithreaded, multi-user, dan SQL database management system (DBMS). Database ini dibuat untuk keperluan sistem database yang cepat, handal, dan mudah digunakan.

3. PhpMyadmin

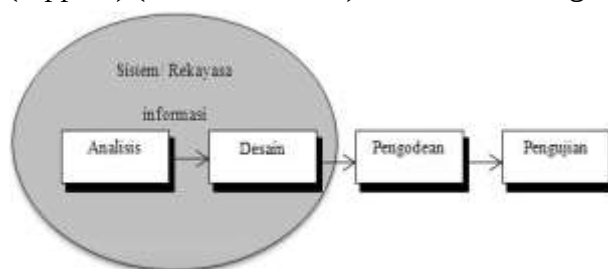
Menurut (Setiawan, 2019) mengemukakan bahwa :

PhpMyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui *World Wide Web* (WWW). *phpMyAdmin* mendukung berbagai operasi

MySQL, diantaranya mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relation*), indeks, pengguna (*users*), perijinan (*permission*), dan lain-lain.

1.1.4 Model Pengembangan Perangkat Lunak

Salah satu metode pengembangan perangkat lunak (*System Development Life Cycle*) adalah dengan model *waterfall*. Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alurhidup klasik (*classic life cycle*)". Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (*support*) (Rosa A.S, 2018). Berikut adalah gambar model air terjun:



Sumber: (Rosa dan Shalahuddin 2018)

Gambar II.2
Ilustrasi Model Waterfall

1. Analisa kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara *intensif* untuk mespesifikasikan 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 fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. 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 kedalam 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 dari segi logik dan fungsional dan 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 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.

1.2 Konsep Dasar Klinik

Penyelenggaraan klinik di Indonesia telah diatur dalam Peraturan Menteri Kesehatan (PMK) No. 28 tahun 2011 tentang klinik. Definisi klinik berdasarkan Peraturan Menteri Kesehatan Republik Indonesia Nomor 028/Menkes/Per/I/2011 dalam (Mahendra, 2017). Klinik adalah fasilitas pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan yang menyediakan pelayanan medis dasar dan atau spesialisik, diselenggarakan oleh lebih dari satu jenis tenaga kesehatan dan dipimpin oleh seorang tenaga medis. Tenaga medis adalah dokter,

dokter spesialis, dokter kandungan atau dokter kandungan spesialis, sedangkan tenaga kesehatan adalah setiap orang yang mengabdikan diri dalam bidang kesehatan serta memiliki pengetahuan dan atau keterampilan melalui pendidikan di bidang kesehatan yang untuk jenis tertentu memerlukan kewenangan untuk melakukan upaya kesehatan (Permenkes 028/Menkes/ Per/I/2011).

1.2.1 Pengertian Pelayanan

Pelayanan adalah setiap tindakan atau kegiatan yang dapat ditawarkan oleh suatu pihak kepada pihak lain, yang pada dasarnya tidak berwujud dan tidak mengakibatkan kepemilikan apapun (Nur Aeni Hidayah, 2015). Selanjutnya Sianipar memaparkan dalam (Iskandar, 2016) pengertian pelayanan sebagai berikut: Cara melayani, menyiapkan atau menjamin keperluan seseorang atau kelompok orang. Melayani adalah meladeni atau membantu mengurus keperluan atau kebutuhan seseorang sejak diajukan permintaan sampai penyampaian atau penyerahannya.

1.2.2 Pengertian Pelayanan Kesehatan

Menurut Notoadmojo dalam (Iskandar, 2016) mengemukakan bahwa, Upaya kesehatan adalah setiap kegiatan untuk memelihara dan meningkatkan kesehatan yang dilakukan oleh pemerintah atau masyarakat. Sejalan dengan hal tersebut, berdasarkan UU RI No. 36 Tahun 2009 pasal 1 ayat 11 pengertian upaya atau pelayanan kesehatan adalah “setiap kegiatan dan/atau serangkaian kegiatan yang dilakukan secara terpadu, terintegrasi dan berkesinambungan untuk memelihara dan meningkatkan derajat kesehatan, pengobatan penyakit, dan pemulihan kesehatan oleh pemerintah dan/atau masyarakat”.

1.3 Tools Program

Untuk mendesain suatu , maka diperlukan beberapa peralatan pendukung (tools program) untuk menggambarkan bentuk sistem secara structural dan actual dengan suatu pendekatan analisa terstruktur, tools program juga berguna untuk mempermudah dalam pembuatan program aplikasi yang dibuat, seperti dalam pembuatan logika dan algoritma program mulai input, sampai pembuatan output.

1.3.1 ERD (*Entity Relationship Diagram*)

Menurut (Sukanto R.A., 2015) ERD adalah bentuk paling awal dalam melakukan perancangan basis data relasional. Jika menggunakan OODMS maka perancangan ERD tidak perlu dilakukan. Entity relationship diagram (ERD) merupakan diagram untuk merancang tabel-tabel yang nantinya akan diimplementasikan pada basis data (Feri Sulianta, 2015). ERD ini dibentuk berdasarkan 3 elemen, yaitu entitas, atribut, dan relasi.

Entitas adalah objek dalam bentuk fisik atau konsep. Entitas ini akan dibuat unik atau berbeda dengan entitas lainnya, misalnya entitas mahasiswa, dosen, mata kuliah, dan sebagainya. Atribut adalah karakteristik atau properti dari entitas, misalnya entitas mahasiswa memiliki atribut nomor induk mahasiswa (NIM), nama, alamat, jenis kelamin, hobi, dan lain-lain. Relasi adalah hubungan antara entitas yang satu dengan yang lainnya, misalnya terdapat hubungan antara entitas mahasiswa dengan mata kuliah karena mahasiswa nantinya akan mengambil mata kuliah.

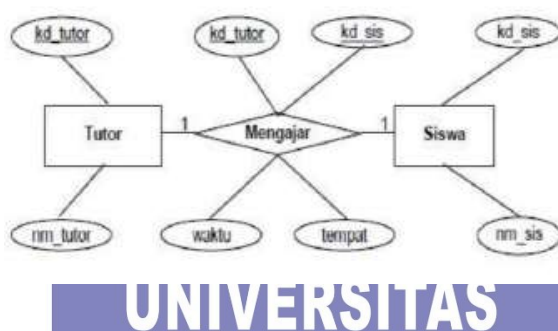
Tahapan proses pemodelan ERD menurut (Priyadi, 2014) adalah:

1. Identifikasi entitas.
2. Menentukan atribut key.

3. Identifikasi relasi.
4. Membuat desain perkiraan.
5. Menentukan kardinalitas.
6. Melengkapi desain dengan atribut deskriptif.

Menurut (Fathansyah, 2015), kardinalitas relasi yang terjadi di antara dua himpunan entitas dapat berupa :

1. Relasi Satu ke Satu (*One to One*) Artinya satu record pada entity A ber-relasi paling banyak satu record juga pada entity B, begitu juga sebaliknya, satu record pada entity B, berelasi paling banyak satu record juga dengan entity A. relasi ini disimbolkan dengan angka 1.

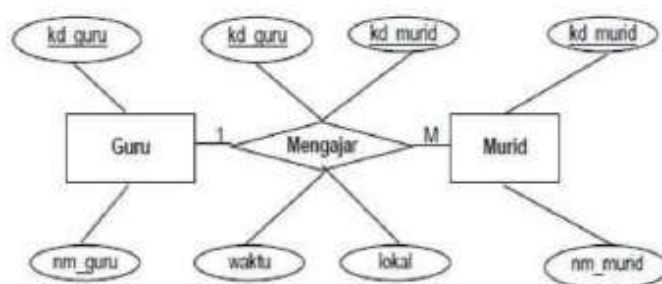


Sumber : (Fathansyah 2015)

Gambar II.3

Ilustrasi Model One to One

1. Relasi Satu ke Banyak (*One to Many*) Artinya satu record pada entity A ber-relasi dengan beberapa record pada entity B, tapi tidak sebaliknya, setiap record pada entity B ber-relasi paling banyak satu record dengan entity A. Dalam diagram E-R, relasi ini disimbolkan dengan angka 1 untuk menyatakan satu dan huruf M atau N untuk menyatakan banyak.

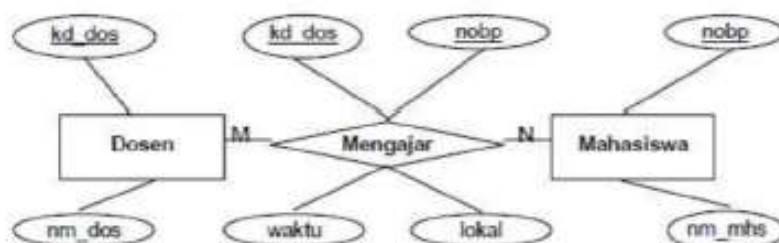


Sumber : (Fathansyah 2015)

Gambar II.4

Ilustrasi Model One to Many

2. Relasi Banyak ke Banyak (*Many to Many*) artinya beberapa record pada entity A dapat ber-relasi dengan beberapa record juga pada entity B, begitu juga sebaliknya, beberapa record pada entity B dapat ber-relasi dengan beberapa record juga pada entity A.



Sumber : (Fathansyah 2015)

Gambar II.5

Ilustrasi Model Many to Many

1.3.2 LRS (*Logical Record Structure*)

Menurut Frieyadie (2011) dalam (Iskandar, 2016) mengatakatakn bahwa, “LRS kepanjangan dari *Logical Record Structure* merupakan hasil dari pemodelan *Entity Relationship* (ER) berserta atributnya sehingga bisaterlihat hubungan-hubungan

antara entitas”. Dalam pembuatan LRS terdapat tiga hal yang dapat mempengaruhi, yaitu:

1. Jika tingkat hubungan (cardinality) satu pada satu (one-to-one), maka digabungkan dengan entitas yang lebih kuat (strong entity), atau digabungkan dengan entitas yang memiliki atribut yang lebih sedikit.
2. Jika tingkat hubungan (cardinality) satu pada banyak (one-to-many), maka hubungan relasi atau digabungkan dengan entitas yang tingkat hubungannya banyak.
3. Jika tingkat hubungan (cardinality) banyak pada banyak (many-to-many), maka hubungan relasi tidak akan digabungkan dengan entitas manapun, melainkan menjadi sebuah LRS.

1.3.3 Pengkodean

Menurut (Jogiyanto, 2014), “Kode digunakan untuk tujuan mengklasifikasikan data, memasukan data ke dalam komputer untuk mengambil bermacam-macam informasi yang berhubungan dengannya”. Dalam pembuatan struktur kode ada petunjuk yang diperlukan, yaitu:

- a. Harus Mudah diingat
- b. Harus Unik
- c. Harus Fleksibel
- d. Harus Efisien
- e. Harus konsisten
- f. Harus Distandarisasi
- g. Hindari pemakaian spasi

- h. Hindari karakter yang mirip
- i. Panjang kode harus sama

Ada beberapa macam tipe dari kode yang dapat digunakan di dalam sistem informasi, diantaranya adalah kode mnemonik (*mnemonic code*), kode urut (*sequential code*), kode blok (*block code*), kode grup (*group code*), dan kode desimal (*decimal code*), masing-masing tipe dari kode tersebut mempunyai kebaikan dan kelemahannya tersendiri. Dalam praktek, tipe-tipe kode yang ada dapat dikombinasikan.

1. Kode Mnemonik (*mnemonic code*)

Digunakan untuk tujuan supaya mudah di ingat. Kode mnemonik dibuat dengan dasar singkatan atau mengambil sebagian karakter dari item yang akan di wakili dengan kode ini. Sebagai contoh kode “P” untuk mewakili pria dan kode “W” untuk wanita akrab untu mudah diingat. Umumnya kode mnemonik menggunakan huruf, akan tetapi dapat juga menggunakan gabungan huruf dan angka misalnya barang dagangan komputer IBM pc dengan ukuran memori 640 Kb, colour monitor, dapat dikodekan menjadi K-IBM- PC-640- CO supaya lebih mudah diingat. Kebaikan dari kode ini adalah mudah diingat dan kelemahannya adalah kode dapat menjadi terlalu panjang.

2. Kode Urut (*Sequential Code*)

Kode yang ini disebut juga kode seri (*serial code*) merupakan kode yang nilainya urut antara satu kode dengan kode berikutnya. Contoh kode urut adalah sebagai berikut:

- a. 001 Kass
- b. 002 Piutang Dagang

- c. 003 Persediaan Produk Selesai
- d. 004 Persediaan Produk Dalam Proses
- e. 005 Persediaan Bahan Baku
- f. 006 Biaya Dibayar Dimuka

3. Kode Blok (*block code*)

Kode blok (*block code*) mengklasifikasikan item ke dalam kelompok blok tertentu yang mencerminkan suatu klasifikasi tertentu atas dasar pemakaian maksimum yang diharapkan. Contoh kode blok adalah sebagai berikut:

Rekening-rekening dalam buku besar dapat diberi kode dengan mengklasifikasikannya ke dalam kelompok rekening utama sebagai berikut:

BLOK KELOMPOK

- 1000-1999 AKTIVA LANGGAR
- 2000-2999 AKTIVA TETAP
- 3000-3999 HUTANG LANGGAR
- 3500-3999 HUTANG JANGKA PANJANG
- 4000-4999 MODAL



4. Kode Desimal (*desimal code*)

Kode desimal (*desimal code*) mengklasifikasikan kode atas dasar 10 unit angka desimal dimulai dari angka 0 sampai dengan angka 9 atau dari 00 sampai dengan 99 tergantung dari banyaknya kelompok.

5. Kode Grup (*group code*)

Kode grup (*group code*) merupakan kode yang berdasarkan field-field, dan tiap field-field nya mempunyai arti.

6. Kode Batang (*barcode*)

Sebagai kumpulan kode yang berbentuk garis, dimana masing-masing ketebalan setiap garis berbeda sesuai dengan isi kodenya

1.3.4 HIPO (*Herarky Input Proses Output*)

Menurut (Jogiyanto, 2014) “HIPO (Hierarchy plus Input-Procces-Output) merupakan metodologi yang dikembangkan dan didukung oleh IBM”. HIPO sebenarnya alat dokumentasi program. Akan tetapi sekarang, HIPO juga banyak digunakan sebagai alat desain dan teknik dokumentasi dalam siklus pengembangan sistem. HIPO berbasis pada fungsi, yaitu tiap-tiap modul didalam sistem digambarkan oleh fungsi utama nya.

HIPO dapat digunakan sebagai alat pengembangan sistem dan teknik dokumentasi program dan penggunaan HIPO ini mempunyai sasaran utama sebagai berikut :

1. Untuk menyediakan suatu struktur guna memahami fungsi-fungsi dari sistem
2. Untuk lebih menekankan fungsi-fungsi yang harus diselesaikan oleh program, bukan nya menunjukan statemen-statemen program yang digunakan untuk melaksanakan fungsi tersebut.
3. Untuk menyediakan penjelasan yang jelas dari input yang harus digunakan output yang harus dihasilkan oleh masing-masing fungsi dari tiap-tiap tingkatan dari diagram-diagram HIPO.
4. Untuk menyediakan output yang tepat dan sesuai dengan kebutuhan-kebutuhan pemakai.

Adapun jenis-jenis dari diagram HIPO yaitu:

1. *Visual table of content (VTOC)*

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

2. *Overview diagram*

Overview diagram menunjukan secara garis besar hubungan dari input, proses dan output. Bagian input menunjukan item-item data yang akan digunakan oleh bagian proses. Bagian proses berisi sejumlah langkah-langkah yang menggambarkan kerja dari fungsi. Bagian output berisi dengan item-item data yang dihasilkan atau dimodifikasi oleh langkah-langkah proses.

3. *Detail diagram*

Detail diagram merupakan tingkatan yang paling rendah di diagram HIPO. Diagram ini berisi dengan elemen-elemen dasar dari paket yang menggambarkan secara rinci kerja dari fungsi.

1.3.5 Diagram Alir Program (*Flowchart*)

1. *Pengertian Flowchart*

Flowchart simbol untuk menggambarkan urutan suatu proses, termasuk proses pengolahan data (Sarosa, 2017). Flowchart sering digunakan untuk menggambarkan algoritma suatu aplikasi, urutan proses, prosedur, maupun aliran kerja (work flow). Simbol flowchart banyak, setiap langkah dan setiap entitas mendapat simbol tersendiri. Untuk menggambar flow chart dibutuhkan alat bantu seperti template atau aplikasi seperti Microsoft Visio. Secara umum simbol flowchart dikelompokkan menjadi empat yaitu keluaran dan masukan, pengolahan, penyimpanan, dan simbol lainnya.

Flowchart sudah lama digunakan, bahkan semenjak masa awal komputasi. Banyak simbol flowchart yang mungkin sekarang tidak ditemui lagi seperti misalnya simbol pita kertas, kartu plong, dan lainnya.

Meskipun sudah cukup lama, flowchart masih digunakan secara intensif. Diagram dalam dokumentasi manajemen kualitas ISO menggunakan flowchart. Panduan untuk menggambarkan flowchart adalah sebagai berikut:

- a. Mendokumentasikan sistem kedalam bentuk narasi.
- b. Dari narasi diidentifikasi bagian atau departemen internal dan pihak eksternal yang terlibat dalam proses, dokumen yang digunakan, proses yang dilakukan, serta aliran dokumen dan data.
- c. Untuk setiap proses, dokumen, data dan aliran data mulai dapat dicari simbol yang tepat
- d. Tiap entitas eksternal maupun internal dibuatkan kolom tersendiri. Aktivitas setiap entitas akan digambarkan dalam tiap kolom secara terpisah.
- e. Aliran flowchart secara umum dari atas kebawah dan dari kiri ke kanan

2. Bentuk Flowchart

Bentuk diagram alur (flowchart) yang sering digunakan dalam proses pembuatan suatu program komputer adalah sebagai berikut:

- a. Program Flowchart

Simbol-simbol yang menggambarkan urutan prosedur secara rinci dan detail antara instruksi yang satu dengan yang lainnya di dalam suatu program computer yang bersifat logic.

b. Sistem Flowchart

Simbol-simbol yang menggambarkan urutan prosedur secara detail di dalam suatu sistem komputer, dan bersifat fisik. Sebelum kita membuat suatu program komputer, yang harus dilakukan terlebih dahulu adalah membuat diagram alir (flowchart), yang sering digunakan adalah program Flowchart.

3. Teknik Pembuatan

Sebelum kita membuat sebuah program komputer, yang harus kita lakukan sebelumnya adalah membuat flowchart yang sering digunakan adalah program flowchart. Teknik pembuatan program flowchart terbagi menjadi dua bagian, yaitu :

a. General Way

Cara ini sering digunakan dalam penyusunan logika suatu program, yang menggunakan pengulangan proses secara tidak langsung (Non-Direct Loop).

b. Iteration Way

Cara ini sering dipakai untuk logika program yang cepat, serta bentuk permasalahan yang kompleks, pengulangan proses yang terjadi bersifat langsung (Direct Loop).



1.3.6 Pengujian Website

Implementasi sistem merupakan tahapan yang akan membahas atau menceritakan sistem yang telah dibangun, sehingga akan diketahui bagaimana proses kerja dari sistem tersebut. Pengujian adalah satu set aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan (R.A. Sukanto, dan Shalahuddin, 2015). Aktifitas pengujian terdiri dari satu set atau sekumpulan langkah dimana dapat menempatkan desain.

Kasus uji yang spesifik dan metode pengujian. Secara umum pola pengujian pada perangkat lunak adalah sebagai berikut:

1. Pengujian dimulai dari level komponen hingga integrasi antar komponen menjadi sebuah sistem.
2. Teknik pengujian berbeda-beda sesuai dengan berbagai sisi atau unit uji dalam waktu yang berbeda-beda pula bergantung pada pengujian pada bagian mana yang dibutuhkan.
3. Pengujian dilakukan oleh pengembang perangkat lunak, dan jika untuk proyek besar, pengujian bisa dilakukan oleh tim uji yang tidak terkait dengan tim pengembangan perangkat lunak (independent test group (ITG)).
4. Pengujian dan penirkutuan (*debugging*) merupakan aktifitas yang berbeda, tapi penirkutuan (*debugging*) harus diakomodasi pada berbagai strategi pengujian. Pengujian lebih fokus untuk mencari adanya kesalahan (*error*) baik dari sudut pandang orang secara umum atau dari sudut pandang pengembangan tanpa harus menemukan lokasi kesalahan pada kode program. Penirkutuan (*debugging*) adalah proses mencari lokasi kesalahan (*error*) pada kode program sehingga dapat segera diperbaiki oleh pembuat program (*programmer*).

Pengujian perangkat lunak adalah sebuah elemen sebuah topik yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi (*verification*) dan validasi (*validation*) (V&V). Verifikasi mengacu pada sekumpulan aktifitas yang menjamin bahwa perangkat lunak mengimplementasikan dengan benar sebuah fungsi yang spesifik. Validasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin

bahwa perangkat lunak yang dibangun dapat ditelusuri sesuai dengan kebutuhan pelanggan (*customer*).

Pengujian untuk validasi memiliki beberapa pendekatan sebagai berikut:

1. *Black-Box Testing* (Pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji yang dibuat adalah:

- Jika user memasukan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
- Jika user memasukan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.