BAB II

LANDASAN TEORI

2.1. Konsep Dasar Program

A. Program

Menurut (Kadir, 2017) “Perangkat lunak adalah kumpulan instruksi yang ditiunjukan kepada komputer”. Istilah program dan aplikasi lebih sering disebut uttuk menyatakan perangkat lunak. Dikalangan profesional teknologi informasi, istilah program bisa digunakan untuk menyatakan hasil karya mereka berupa instruksi-instruksi untuk mengendalikan komputer. Disisi pemakai, hal itu disebut aplikasi. Jadi istilah yang digunakan untuk menyatakan hal yang sama lebih ditentukan oleh masalah persepsi.

Terkait dengan program, terdapat istilah pemogram dan pemograman. Pembuat program dinamakan pemrogram (*programer*). Tugasnya adalah menulis program dan memastikan bahwa program sesuai dengan spesifikasi yang dikehendaki, adapun yang dinaamkan pemograman adalah proses untuk menyelesaikan masalah dalam bentuk langkah-langkah penyelesaian yang dapat dikerjakn oleh komputer (yang disebut algoritma) hingga ke penerjemahan kode dalam suatu bahasa pemograman, sehingga maslah tersebut benar-benar bisa dieksekusi oleh komputer.

B. Bahasa Pemograman

Komputer bekerja atas dasar kode biner atau kode yang mampunyai dua keadaan hingga berupa 0 dan 1. Jika dinyatakan dalam keadaan lampau, kode 0 menyatakan keadaan padam dan kode 1 menyatakan keadaan menyala. Atas dasar inilah, program

pada masa awal terciptanya komputer dengan menggunakan bahasa pemgroman yang berbasis pada kode biner dan dinamakan bahasa mesin.

Bahasa mesin dan bahasa rakitan tergolong sebagai bahasa beraras rendah (low level language), yang lenihberorientasi pada mesin. Artinya pemoraman harus benar-benar memahami hal-hal mengenai mesin bersangkutan untuk dapat menuliskan program. Untuk mengatasi kelemahan ini, muncul bahasa beraras tinggi (*high level languageI*). C, C++, Java, Phyton, dan Rubytermasuk kategori bahsa beraras tinggi.

Program, apapun bentuknya dibuat menggunakan bahasa pemograman. Kode yag digunakan untuk menyusun program biasa disebuut kode program atau kode sumber (*source code*).

Menurut (Kadir, 2017) “Java adalah bahasa pemograman yang dapat dijalankan diberbagai komputer, termsuk telpon genggam”. Dikembangkan oleh *Sun Microsystems* dan dirilis 1995. Java adlah bahasa pemograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai platform sistemoperasi. Perkembangan java tidak hanya berfokus pada suatu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source*.

C. Basis Data

Basis data yang juga dikenal dengan *database* yang merupakan catatan atas kumpulan fakta yang mewakili suatu objek. Data memiliki ciri bersifat mentah dan tidak memiliki konteks. Sendangkan basis atau *base* dapat diartikan sebagai markas, tempat berkumpul dari suatu objek atau representasi objek. Berikut merupakan beberapa definisi basis data :

Basis data didefinisikan menurut Chou (Jayanti & Sumiar, 2018) “ Basis data sebagai kumpulan informasi bermanfaat yang iorganisasikan kedalam tata cara yang

Berdasarkan basis data dapat didefinisikan sebagai sekumpulan data yang terintegrasi, yang diorganisasi utnuk memenuhi kebutuhan para pemakai didalam suatu organisasi. Dalam inplementasinya, untuk memudahkan dalam pengaksesan data, data disusun suatu struktur logis yang menjelaskan bahwa :

a. Kimpulan tabel menyusun basis data.
b. Tabel tersusun atas sejumlah record
c. Sebuah record mengandung sejmlah *field.*
d. Sebuah *field* disimpan dalam bentuk kumpulan bit.

Tujuan dalam merancang basis data adalah

1. Kecepatan dan kemudahan (*speed*)

   Pemanfaatan basis data memungkinkan untuk dapat menyimpan data ataumelakukan perubahan/manipulasi terhadap data aatu menampilkan kembali data tersebut dengan cepat dan mudah.

2. Efeiseni ruang penyimpanan (*space*)

   Penggunaan ruang penyimpanan didalam basis data dilakukan untuk mengurangi jumlah *radudancy* (pengulangan) data, bak dengan melakukan penerapan sejunlah pengkodean atau dengan membuat relasi-relasi (dalam bentuk file) antar kelompok data yang saling berhubungan.

3. Keakuratan (*accuracy*)

   Pemanfaatan pegodena atau pembentukan relasi anatar data bersama dengan penerapan aturan / batasan tipe data, domain data, keunikan data, dan sebagainya, yang diterapkan dalam basis data, snagat berguna untuk menentukan ketidakakuratan pemasukan atau penrimpanan data.

4. Ketrsediaan (*avaibility*)

   Pertumbuhan data (baik dari jumlah maupun jenisnya) sejalan dengan waktu akan semakin mebutuhkan ruang penyimpanan yang besar

5. Kelengkapan (*completeness*)

   Lengkap tau tidalkny asuatu data yang dikelola bersifata relatif, baik terhadap kebutuhan pemakai maupun terhadap waktu. Dalam basis data , struktur dari basis data harus disimpan.

6. Keamanan (*securuty*)

   Sistem kemanan diguankan untuk dapat menetukan siapa saja yang boleh menggunakan basis data dan menetukan jenis opersai apa saja yang boleh dilakukan.

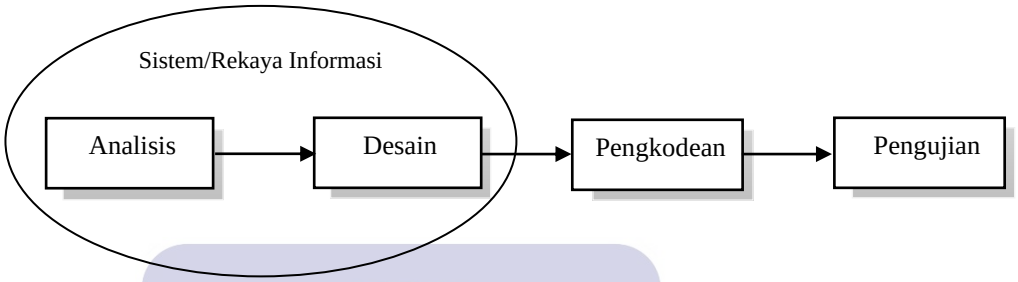
7. Kebrsamaan (*shareability*)

   Pemakai basis data sering kali tidakk terbatas hanya pada asustu pemakaina saja atau oleh suatu sisten aplikasi saja. Basis data yang dikelola oleh sistem yang mendukung lingkunhan multi*user*, akan dpat memenuhi kebutuhan ini, tetapi dengan menjaga/mengjindari terhadap mnculnya pesoalan batru seperti inkonsistensi data (karena data yang sama diubah oleh banyak pemakai pada saat bersamaan).

D. Model Pengembangan Perangkat Lunak

Model pengembangan sistem yang digunakan dalam penulisan ini adalah metode pengembangan sistem dengan *waterfall*. Menurut (A.S & Shalahuddin, 2015) mengemukakan bahwa “Model SDLC air terjun *(waterfall*) sering juga disebut 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*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisa, desain, pengkodean, pengujian, dan tahap pendukugan (*suppport*)". Beriku adalah gambar model air terjun :



Sumber : A.S & Shalahuddin, 2016

Gambar II.1. 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 perangkat lunak pada tahap ini perlu untuk didokumentasikan.
2. *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. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan kode program 12 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 segi lojik 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 yang baru.

2.2. Tools Program

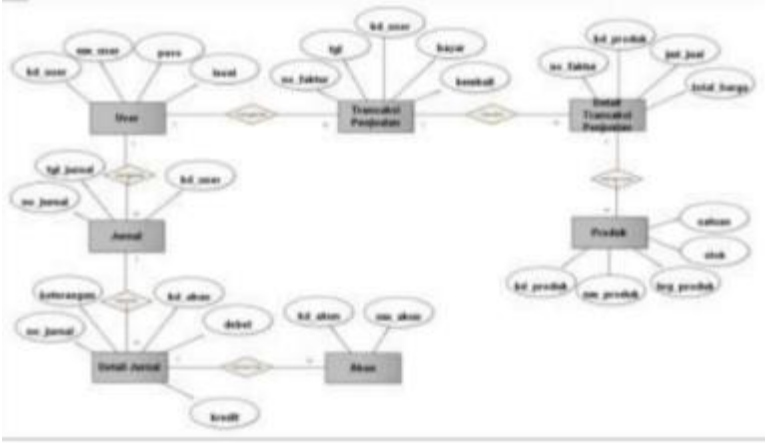
A. *Enterprise relationship diagram (ERD)*

1. Pengertian dan komponen *Enterprise Relation-ship Diagram* (ERD)

Menurut (Trisyanto, 2017) *"ERD* adalah gambaran atau diagram yang menunjukan informasi dibuat, disimpan, dan digunkan dalam sistem bisnis". Entitas biasnya menggambarkan jenis informasi yang sama. Dalam entitas

digunakan untuk menghubungkan antar entitas yang sekaligus menunjukan hubungan antar data. Pada akhirnya *ERD* bisa juga digunakan untuk menunjukan aturan-aturan bisnis yang ada pada sistem informasi yang dibangun.bagimana menggunakan *ERD* untuk menunjukan aturan bisnis? Ada beberapa poin yang bisa dilihat untuk menjawab petanyaan ini :

a. Atutan bisnis adalah batasan yang harus diikuti ketika sistem beroperasi.
b. Simbol *ERD* hanya menunjukan satu *instance* dari entitas harus ada sebelum *instance* dari suatu entitas.
c. Simbol *ERD* dapat menunjukan ketika salah satu *instance* dari suatu entitas dapat direlasasikan dengan suatu anggota atau lebih dari entitas lainnya.
d. Simbol *ERD* juga menunjukan ketika eksistensinya dari suatu *instance* dalam suatu I adalah opsional untuk sebuah relasi dengan *instance* lain dari suatuj entitas



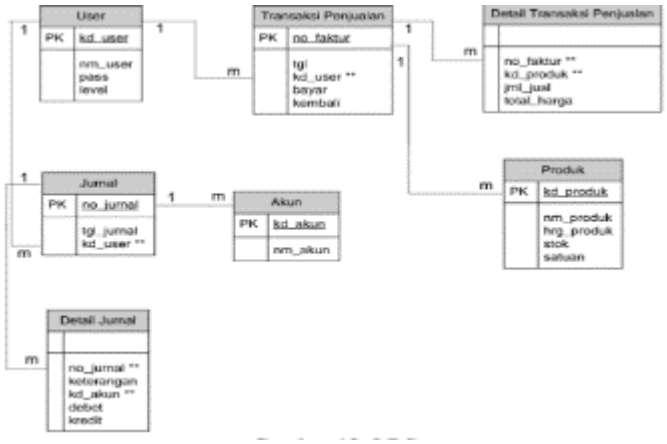
Sumber : (Mesran, 2019)

Gambar II.2 Enterprise relationship diagram (ERD)

2. LRS (*logical Record Structure)*

Menurut Deni Kuswono (Kadir, 2017)(Setyawati, 2019) “Logical record structureberasal dari setiap entity yang diubah ke dalam bentuk sebuah kotak dengan nama entity berada di luar kotak dan atribut berada di dalam kotak”. Untuk menetukan kardinalitas, jumlah table dan Foreign Key sebagai berikut:

a. One-to-one Satu entitas berhubungan dengan paling banyak satu entitas lain.
b. One-to-many
   Satu entitas dapat berhubungan dengan lebih dari satu entitas lain.
c. Many-to-many
   Beberapa entitas dapat berhubungan dengan beberapa entitas lain.



Sumber : Jurnal MIB Volume 3 No 1 Januari 2019

Gambar II.3 Logical Record Structure (LRS)

B. Pengkodean

Nur, 2016 “Pengkodean yaitu melaku-kan penerapan hasil rancangan ke dalam bentuk yang dapat dibaca dan di mengerti oleh komputer. Pada tahap ini hasil dari peran-cangan mulai diterjemahkan ke dalam bahasa mesin melalui bahasa pemprograman.

C. HIPO *(Hierarki Input Proses Output)*

1. Pengertian HIPO

Menurut (Trisyanto, 2017) HIPO (*Hirarchy Plus Input* - proses – *output*) merupakan metodologi yang dikembangkan dan didukung oleh IBM. HIPO sebenarnya adalah alat dokumentasi program. HIPO dapat digunakan sebagai alat pengembangan sistem dan teknik dokumentasi program dan penggunaan HIPO ini mempunyai sasran utama sebagi berikut:

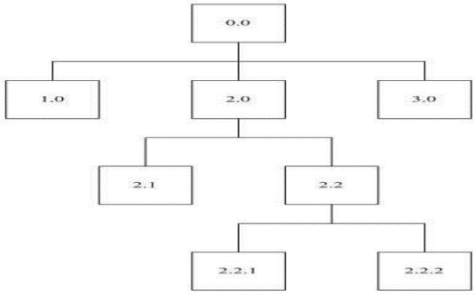
a. Untuk menyediakan suatu struktur guna memahami fungsi dari sistem.
b. Untuk lebih menekankan fungsi-fungsi yang harus diselesaikan oleh program, bukannya menunjukan perintah-perintah program yang digunakan untuk melaksanakan fungsi tersebut.
c. Untuk menyediakan penjelasan yag jelas dari input yang yang harus digunakan dan *output* yang harus dihasilkan oleh masing-masing fungsi pada tiap-tiap tingkatan dari diagram-diagram HIPO.
d. Utuk menyediakan *output* yang tepat sesuai dengan kebutuhan pemakai.

2. Tingkatan Diagram HIPO

Fungsi-fungsi dari sistrem digambarkan oleh HIPO dalam tiga tingkatan. Untuk masing-masing tingkatan digambarkan dalam bentuk diagram tersendiri. Dengan demikina HIPO menggunakan tiga macam diagram untuk masing-masing tingkatannya, yaitu sebagai berikut.

a. *Visual Table of Contents* (VTOC)

   Diagram ini menggambarkan hubungan dari fungsi-fungsi disistem secara berjenjang.



Sumber : Trisyanto, 2017

Gambar II.4. HIPO

b. *Overview Diagrams*

Overview diagrams menunjukan secatra garis besr hubungan dari input, proses, dan output. Bagian *input* merupakan item-item data yang akan digunkan oleh bagian proses. Bagia proses berisi sejumlah langklah-langkah yang menggambarkan kerja dari fungsi. Bagian *output* berisi dengan *item-item* data yang dihasilkna atau dimodifikasi oleh langkah-langkah proses.

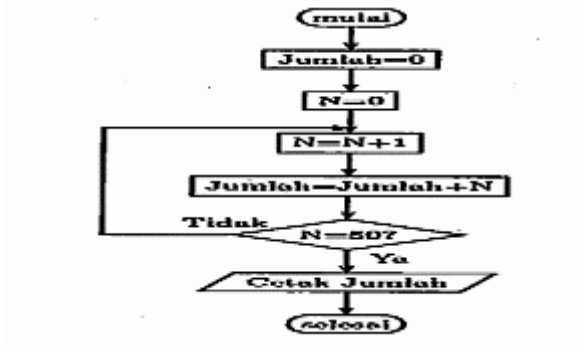
c. *Detail Diagrams*

Detail Diagrams merupakan diagram tingkatanyang paling rendah di tangan HIPO. Diagram ini berisi dengn elemen-elemen dasar dari paket yang menggambarkan secara rinci kerja dari fungsi.

D. Diagram alir program *(flowchart)*

1. Pengertian

Menurut (Suyanto, 2018) "*Flowchart* adalah representasi secara diagram yang menggambarkan urutan operasi dalam menyelesaikan suatu masalah". *Flowchart* biasanya dibuat oada tahap awal sebelum pembuatan program. *Flowchart* juga merupakan fasilitas komunikasi antara programer dan pebisnis. Selain itu *flowchart* juga berguna untuk memahami logika progaram apalagi yang panjang dan kompleks. Dengan adanya *flowchart* program dibuat dengan mudah.



Sumber : Suyanto, 2018

Gambar II.5 Flowchart penjualan 50 bilangan asli pertama

Keutungan yang diperoleh dengan pemanfaatan *flowchart* adalah

a. Komunikasi : *Flowchart* dapat menjadi sarana komunikasi bagi pihak-pihak yang terlibat dalam pengembangan program.
b. Analisis : *Flowchart* dapat digunakan untuk analisis masalah.
c. Penulisan program : *Flowchart* membantu kosentrasi penulisan program karena alurnya sudah jelas.
d. Pelalcakan kessalahan : *Flowchart* membantu juga dalam proses pelalcakan kesalahan program.
e. Pemeliharaan program : Bagian program yang diubah dapat ditentukan dengan melihat *flowchart* sehingga lebuih efisien waktu.

Keterbatasan *Flowchart*

a. Logika rumit : Kadang logika program menjadi sedemikian kompleks sehingga pengguna *flowchart* menjadi begitu rumit.
b. Modifikasi : Jika *flowchart* perlu dimodifikasi maka arus dilakukan penggambaranyang seutuhnya.

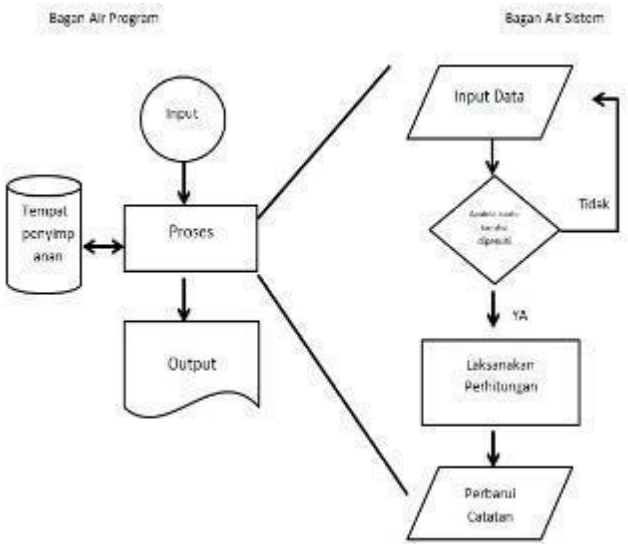
c. Reproduksi : Karena *flowchart* berbentuk simbol-simbol dan tidak dapat diketik saja, maka proses produksi dalam media lain menjadi masalah tersendiri.

d. Inti dari ‘apa yang dikerjakan’ dengan muda menjadi kabus kerena terjadi perincian ‘bagaiman mengerjakannya’.

2. Bentuk *flowchart*

a. Program *Flowchart*

Menurut (Dawan, 2019) *Flowchart* program adalah baganalir yang menjelaskan secara rinci langkah-langkah dari proses program. Program *flowchart* dibuat dari deriviksi bagan alir sistem. Bagan alir logika program digunakan untuk menjelaskan tiap-tiap langkah di dalam program komputer secara logika. Bagian alir logika program dipersiapkam oleh analisis sistem.



Sumber : Dawan, 2019

Gambar II. 6 Bagan alir logika program

b. Sistem *Flowchart*

Menurut (Dawan, 2019) menjelakan bahwa “Flowchart sistem atau bagan alir sistem merupakan bagan yang menggambarkan arus pekerjaan secara keseluruhan dari sistem”. Bagian ini menjelaskan setiap urutan dari prosedur-prosedur yang ada di dalam suatu sistem. Flowchart sistem menunjukkan apa yang dikerjakan di sistem.

Contoh : *flowchart* sistem dalam suatu pabrik, alur kerja produksi suatu barang, dan sebagainya.

3. Teknik pembuatan

a. *General Way*

General way adalah teknik pembuatan *flowchart* dengan menggunakan proses pengulangan secara tidak langsung *(non direct loop).*

b. *Iteration Way*

Iteration Way adalah teknik pembuatan *flowchart* dengan menggunakan proses secara langsung.

E. Implementasi Dan Pengujian Unit

Testing atau pengujian unit menurut (Iriadi & Rosdiana, 2017) yaitu “Pengujian perangkat lunak yaitu dengan menggu-nakan metode *black box testing*. Metode uji coba *blackbox testing* memfokuskan keperluan fungsional dari *software* Uji coba blackbox testing berusaha menemukan kesalahan dalam fungsi-fungsi yang salah atau hilang”. *Black box* itu sendiri adalah menurut

Mustaqbal, Firdaus, & Rahmadi, 2015 "pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, pengujian dapat mengartikan kumpulan kondisi input dan melakukan pengujian pada spesifikasi fungsional program."