

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Program

Bentuk dari berbagai macam jenis aplikasi yang dipergunakan dalam bidang bisnis ataupun dalam bidang ilmiah yang berguna untuk menghasilkan suatu laporan, informasi atau tujuan yang diinginkan berupa rangkaian instruksi dalam bahasa *computer* yang disusun secara logis dan sistematis biasa disebut dengan program.

Pemrograman menurut indrajani (2007:22) adalah adalah “perangkat lunak atau software yang dapat digunakan dalam proses pembuatan program yang melalui beberapa tahapan-tahapan penyelesaian masalah”

A. Program

Menurut yulikuspartono (2009:54), Program adalah “sederetan instruksi atau *statement* dalam bahasa yang dimengerti oleh komputer yang bersangkutan”. Proses pemrograman komputer bukan hanya sekedar menulis suatu urutan instruksi yang harus dikerjakan oleh komputer, akan tetapi bertujuan untuk memecahkan suatu masalah serta membuat mudah pekerjaan yang diinginkan oleh pemakai.

Salah satu dari tahap pembangunan suatu program adalah menerjemahkan atau mengkodekan rancangan terinci yang telah dibuat menjadi suatu program komputer yang siap pakai. Dengan menerjemahkan berarti penulisan program dengan menggunakan satu bahasa pemrograman yang kita kuasai. Bahasa pemrograman



merupakan prosedur atau tata cara penulisan program. Pada bahasa pemrograman terdapat dua faktor penting yaitu sintaks dan simatik.

Sintaks (*syntax*) adalah aturan-aturan gramatikal yang mengatur tata cara penulisan kata, ekspresi dan pernyataan, sedangkan simatik adalah aturan-aturan untuk menyatakan suatu arti. Bahasa pemrograman merupakan suatu proses guna mengimplementasikan algoritma dengan menggunakan suatu bahasa program, satu hal yang cukup penting sebelum seorang programmer mulai menyusun program adalah memilih bahasa pemrograman yang akan digunakan. Fungsi bahasa pemrograman adalah sebagai media untuk menyusun dan memahami serta sebagai alat komunikasi antara programmer dengan komputer, meskipun dapat juga digunakan sebagai alat komunikasi antara orang yang satu dengan yang lain.

Secara umum bahasa pemrograman dapat dibagi dalam empat kelompok yaitu:

1. Bahasa Tingkat Rendah (*Low-Level-Language*)

Bahasa tingkat rendah merupakan bahasa pemrograman yang berorientasi pada mesin. Pemrograman yang menggunakan bahasa ini harus dapat berfikir berdasarkan logika mesin komputer, sehingga bahasa ini dimulai kurang fleksibel dan sulit untuk dipahami pemula.

2. Bahasa Tingkat Menengah (*Middle-Level-Language*)

Bahasa tingkat menengah merupakan bahasa pemrograman yang menggunakan aturan-aturan gramatikal dalam penulisan ekspresi atau pernyataan dengan standar bahasa yang mudah dipahami oleh manusia.



3. Bahasa Tingkat Tinggi (*High-Level-Language*)

Bahasa tingkat tinggi merupakan bahasa pemrograman yang memiliki aturan-aturan gramatikal dalam penulisan ekspresi atau pernyataan dengan standar bahasa dapat dipahami secara langsung oleh manusia.

4. Bahasa Berorientasi Obyek (*Object-Oriented-Language*)

Bahasa berorientasi pada obyek merupakan bahasa pemrograman yang mengandung “kapsul-kapsul” yang berisi fungsi-fungsi untuk menyelesaikan suatu masalah. Dengan bahasa ini pemrograman tidak lagi harus menuliskan secara detail semua pernyataan dan ekspresi seperti pada bahasa tingkat tinggi.

B. Bahasa Pemrograman

Menurut kusrini (2007:171) “Visual basic adalah salah satu bahasa pemrograman komputer. Visual basic merupakan salah satu *development tool*, yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi windows.

“Visual” dalam hal ini merupakan bahasa pemrograman yang menyerahkan berbagai macam desain dengan model GUI (*Graphical User Interface*). Hanya dengan mengetik sedikit kode pemrograman dan sudah dapat menikmati program dengan tampilan menarik.

“Basic” menunjukan bahasa pemrograman *BASIC (Biginer All-Purpose Symbolic Instruction Code)*. Visual basic dikembangkan dari bahasa *BASIC* yang ditambah ratusan perintah tambahan, *function*, *keyword*, dan banyak yang berhubungan langsung dengan GUI (*Graphical User Interface*).



Visual basic merupakan *event-driven programming* (Pemrograman Terkendali Kejadian) artinya program menunggu sampai dan merespon dari pemakai berupa event atau kejadian tertentu seperti tombol diklik, pilih menu dan lain-lain. Dalam pengembangan aplikasi, visual basic menggunakan pendekatan visual untuk merancang *user interface* dalam bentuk form, sedangkan untuk kodingnya menggunakan bahasa *basic* yang mudah dipelajari.

Visual basic berbasiskan prinsip *Object Oriented Programming* (OOP) dan dikembangkan dengan basic visual yang berarti menggunakan sarana grafis untuk mengembangkannya. Visual basic juga bersifat *modular programming* karena kode-kode program letaknya tersebar di dalam modul-modul yang terpisah-pisah.

C. Basis data (*Database*)

Database menurut Connolly, T. M., & Begg, C. E. (2010,65) merupakan kumpulan berbagai data logika dan deskripsi, yang dirancang untuk kebutuhan informasi organisasi. *Database* merupakan salah satu komponen yang penting dalam system informasi, karena berfungsi sebagai basis penyedia informasi bagi para penggunanya (*user*).

Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

1. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*Redudancy*) .
2. Himpunan kelompok data atau arsip yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah

3. Kumpulan file yang saling berhubungan yang disimpan dalam media elektronik.

Basis data memiliki prinsip kerja dan tujuan. Prinsip utamanya adalah pengaturan data. Sedangkan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data.

Penulis menjelaskan aplikasi basis data yang digunakan pada program yang dibangun, yaitu:

- a. Xampp

Menurut Wicaksono (2008:7) *XAMPP* adalah “sebuah software yang berfungsi untuk menjalankan website yang berbasis php dan dengan menggunakan pengolahan data *MySQL*”. *XAMPP* merupakan software yang mudah digunakan, gratis dan mendukung instalasi di Linux dan Windows. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia *Apache Web Server*, *MySQL Database Server*, *PHP Support* (*PHP 4* dan *PHP 5*) dan beberapa *module* lainnya. Hanya bedanya kalau yang versi untuk Windows sudah dalam bentuk instalasi grafis dan yang Linux dalam bentuk file terkompresi tar.gz. Kelebihan lain yang berbeda dari versi untuk Windows adalah memiliki fitur untuk mengaktifkan sebuah server secara grafis, sedangkan Linux masih berupa perintah-perintah di dalam console. Oleh karena itu yang versi untuk Linux sulit untuk dioperasikan. Dulu *XAMPP* untuk Linux dinamakan *LAMPP*, sekarang diganti namanya menjadi *XAMPP FOR LINUX*.

b. ODBC (*Open Database Connectivity*)

Menurut Santoso (2006:41) memberi batasan bahwa “*Open Database Connectivity* (ODBC) merupakan driver yang berguna untuk menghubungkan atau jembatan antar muka pemrograman dengan aplikasi lainnya”. ODBC memungkinkan kita untuk dapat mengakses program tanpa harus tergantung dengan satu program aplikasi database yang ada.

MyODBC atau *MySQL connector* ODBC merupakan driver yang disediakan oleh *MySQL* untuk dapat diakses aplikasi lainnya, seperti *Visual Basic*, *Borland Delphi*, dan lainnya selama program aplikasi tersebut dapat menggunakan ODBC sebagai antar mukanya. Dengan adanya *MyODBC*, kita dapat membuat, mengolah, database di *MySQL* dan dihubungkan dengan *Visual Basic*.

c. *MySQL*

Menurut Anhar (2010:21) menjelaskan “*MySQL* adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Manajemen System*) atau DBMS yang mampu mengirim dan menerima data dengan sangat cepat dan multi user”. *MySQL* memiliki dua bentuk lisensi, *Free software* dan *shareware*. Penulis sendiri dalam menjelaskan buku ini menggunakan *MySQL free software* karena bebas menggunakan database untuk keperluan pribadi atau usaha tanpa harus membeli.

D. Model Pengembangan Perangkat Lunak

Menurut Pressman (2010:39) adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Nama model ini sebenarnya adalah “*linear Sequential Model*”. Model ini termasuk ke dalam *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh winstons Royce sekitar tahun 1970



sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam *Software Engineering* (SE).

Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

Waterfall adalah suatu metodologi pengembangan perangkat lunak yang mengusulkan pendekatan kepada perangkat lunak sistematis dan sekuensial yang mulai pada tingkat kemajuan sistem pada seluruh analisis, *design*, kode, pengujian dan pemeliharaan.

Masalah dengan *waterfall*:

1. *Waterfall* pada umumnya digunakan untuk rekayasa sistem yang besar yaitu dengan proyek yang dikerjakan di beberapa tempat berbeda, dan dibagi menjadi beberapa bagian sub-proyek.
2. Karena sifatnya kakunya model ini cocok ketika kebutuhan dikumpulkan secara lengkap sehingga perubahan bisa ditekan sekecil mungkin. Tapi pada kenyataan jarang sekali konsumen yang bisa memberikan kebutuhan secara lengkap.
3. Perubahan sulit dilakukan karena sifatnya yang kaku



2.2 Tools Program

Merupakan alat yang digunakan untuk menggambarkan bentuk logika model dari suatu sistem dengan menggunakan simbol, lambang, diagram yang menunjukkan secara tepat arti dan fungsinya. Fungsinya sendiri adalah untuk menjelaskan kepada user bagaimana fungsi dari sistem dapat bekerja dengan bentuk *logical model* dan *physical model*.

Adapun peralatan *pendukung* yang penulis gunakan dalam pembuatan perancangan program ini sebagai berikut:

A. Entity Relantionship Diagram (ERD)

Menurut Wahana Komputer (2010:30) ERD adalah “Sebuah diagram yang secara konseptual memetakan hubungan antara penyimpanan pada diagram”.

Sedangkan LRS (*Logical Record Structure*) adalah di bentuk dengan nomor dari tipe *record*. Beberapa tipe *record* digambarkan oleh kotak empat persegi panjang dengan nama yang unik. Lrs juga terdiri dari hubungan diantara tipe *record*. Dua metode yang dapat di konversikan ke LRS. Metode lain di mulai dengan ERD dan langsung di konversikan ke LRS. Pada masing-masing *relational* hubungan antar file direlasikan dengan kunci relasi (*relation key*) yang merupakan kunci utama dari masing-masing file.

Hubungan attribute yang ada di dalam satu atau dua file, yaitu sebagai berikut:

1. *One to one relationship*

Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding satu

2. *One to many relationship*

Hubungan antar *file* pertama dengan *file* kedua adalah satu berbanding banyak atau dapat pula dibalik banyak lawan satu.

3. *Many to many relationship*

Hubungan antara *file* pertama dengan *file* kedua adalah banyak berbanding banyak

4. Relasi one to one

Hubungan antara satu *atribut* dengan *attribute* yang lain dalam satu *file* yang sama mempunyai hubungan satu lawan satu.

5. Relasi many to one

Hubungan antara satu *atribut* dengan *atribut* lainnya dalam satu *file* yang sama mempunyai hubungan satu lawan banyak.

6. Relasi many to many

Hubungan antara satu *atribut* dengan *atribut* yang lain dalam satu *file* yang sama mempunyai hubungan banyak lawan banyak.



B. Pengkodean

Menurut Kuarini dan Koniya (2007:23) Pengkodean adalah “Pemberian tanda atau nomor tertentu dengan memakai angka, huruf, kombinasi angka dan huruf pada setiap akun dan rekening”. Tujuan Pengkodean untuk mengklasifikasikan data memasukan data kedalam komputer. Kode dapat dibentuk dari kumpulan angka, huruf dan karakter-karakter khusus.

Dalam merancang kode yang baik ada beberapa hal yang harus diperhatikan, yaitu sebagai berikut:

1. Harus unik

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

2. Hindari karakter yang mirip

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

3. Harus konsisten

Kode harus konsisten dengan kode yang telah digunakan.

4. Harus mudah diingat

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

5. Harus fleksibel

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

6. Harus distandarisasi

Kode harus distandarisasi untuk seluruh tingkatan dan departemen dalam organisasi. Kode yang tidak standartakan mengakibatkan kebingungan, salah pengertian dan cen derungdapat kerja dikesalahan pemakai begitu juga dengan yang menggunakan kode tersebut.



7. Hindari spasi

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

Ada beberapa macam tipe kode yang dapat digunakan dalam sistem informasi, antara lain:

a. Kode Blok (*Block Code*)

Mengklasifikasikan *item* kedalam kelompok blok tertentu yang mencerminkan satu klasifikasi tertentu atas dasar pemakaian maksimum yang diharapkan.

b. Kode Grup (*Group Code*)

Kode yang berdasarkan field-field dan tiap-tiap field kode mempunyai arti tertentu.

c. Kode Urut (*Sequential Code*)

Disebut juga dengan kode seri, merupakan kode yang nilainya urutan antara satu kode dengan kode berikutnya.

d. Kode Mnemonik (*Mnemonic code*)

Bertujuan supaya kode mudah diingat, dibuat dengan dasar singkatan atau mengambil sebagian karakter dari *item* yang akan diwakili dengan kode ini.

e. Kode decimal (*Desimal Code*)

Mengklasifikasikan kode atas dasar 10 unit angka decimal dimulai dari angka 0 sampai dengan angka 9 atau dari 00 sampai dengan 99 tergantung dari banyaknya kelompok.



C. HIPO (*HierarchyPlus Input Process Output*)

Menurut jogiyanto (2005:787) HIPO (*HierarchyPlus Input Process Output*) adalah alat desain dan teknik dokumentasi dalam siklus pengembangan sistem. Hipo berisi sebuah set diagram secara grafis yang menjelaskan fungsi sebuah sistem. Bagian-bagian HIPO adalah sebagai berikut:

1. Index Program

Merupakan nomor acuan yang menunjukkan nomor layar dialog.

2. Nama Program

Merupakan nama layar dialog atau nama suatu program.

3. Escape Program

Merupakan nomor layar dialog sebelumnya yang akan dituju balik.

Sasaran HIPO:

- a. Untuk menyediakan suatu struktur guna memahami fungsi dari sistem.
- b. Untuk lebih menekankan fungsi-fungsi yang harus diselesaikan oleh program.
- c. Untuk menyediakan *output* yang tepat dan sesuai dengan kebutuhan pemakai.
- d. Untuk menyediakan penjelasan yang jelas dari *input* dan *output* pada masing-masing tingkatan dari HIPO.

Tiga macam diagram dalam tingkatan HIPO yaitu:

1. Overview Diagram

Overview diagram menunjukkan secara garis besar hubungan dari *input*, proses dan *output*.



2. *Visual Table Of Contents (VTOC)*

Diagram ini menggambarkan hubungan dari fungsi-fungsi dalam sistem berjenjang.

3. *Detail Diagram*

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

D. **Diagram Alir Data (Flowchart)**

Flowchart menurut Landjamudin (2005:263) adalah “Bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah”. Flowchart digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi dalam membuat suatu algoritma.

1. Flowchart sendiri terdiri dari tiga struktur, yaitu:

a. Struktur Sederhana (*Sequence Structure*)

Diagram yang alurnya mengalir secara berurutan dari atas ke bawah dengan kata lain tidak adanya percabangan ataupun perulangan.

b. Struktur Perulangan (*Looping Structure*)

Pemutaran kembali, terjadi kendali mengalihkan arus diagram alur kembali keatas, sehingga beberapa alur berulang beberapa kali.

c. Struktur Percabangan (*Branching Structure*)

Diagram alurnya banyak terjadi alih control berupa percabangan dan terjadi apabila kita dihadapkan pada suatu kondisi dengan dua pilihan benar atau salah.



2. Bentuk *flowchart* yang sering digunakan dalam proses pembuatan suatu program terdiri dari:

a. Sistem *Flowchart*

Bagan alur yang menggambarkan urutan prosedur secara detail di dalam suatu sistem komputerisasi dan bersifat fisik.

b. Program *Flowchart*

Simbol-simbol yang menggambarkan proses secara rinci dan detail diantaranya instruksi yang lainnya didalam suatu program komputer yang bersifat logis.

3. Teknik pembuatan program *Flowchart* ini dibagi menjadi dua bagian yaitu:

a. *Iteration way*

Yaitu teknik pembuatan yang dipakai untuk logika program yang cepat serta bentuk permasalahan yang kompleks yang bersifat langsung atau *direct loop*.

b. *General way*

Yaitu teknik pembuatan dengan cara ini, lazim digunakan dalam menyusun logika suatu program, yang menggunakan pengulangan proses secara tidak langsung atau *non direct loop*.

