

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Program

Konsep dasar program merupakan suatu gambaran dari program aplikasi yang akan dibangun. Sekarang ini, semua perusahaan pastinya sudah harus terkomputerisasi. Dengan adanya sistem yang terkomputerisasi dapat mempermudah setiap karyawan yang ada pada perusahaan tersebut dalam melakukan kegiatannya.

Untuk itu dengan adanya penelitian ini, penulis berinisiatif membuat sebuah program pada jasa tempat pencucian motor yang berbasiskan desktop. Dalam pembuatan program ini, penulis menggunakan *Borland Delphi 7*.

2.1.1. Program

Proses pembuatan kode program ini memiliki unsur yang terdiri dari *input*, proses, dan *output* yang dimana proses *input* berfungsi sebagai masukan, dan proses berfungsi untuk melakukan pengujian program sedangkan *output* ini adalah sebagai hasil akhir dari proses pembuatan program. Menurut Kurniawan (2011:65) menyatakan “Program adalah kumpulan instruksi yang disusun dengan urutan nalar yang tepat untuk menyelesaikan suatu persoalan”. Sedangkan menurut Sahyar (2016:2) “Program adalah perintah-perintah atau instruksi yang disusun berdasarkan algoritma dengan menggunakan bahasa pemrograman untuk penyelesaian suatu masalah”.

Berdasarkan kutipan diatas, dapat disimpulkan bahwa program merupakan barisan kata yang disusun sedemikian rupa untuk menghasilkan sebuah perintah yang telah dieksekusi oleh komputer.

2.1.2. Bahasa Pemrograman

Bahasa pemrograman sering kita jumpai di berbagai macam media informasi yang dimana bahasa pemrograman ini merupakan bahasa baku yang sering digunakan seorang *programmer* untuk mengetahui pemrograman jenis apa yang digunakan dalam proses pembuatan program nanti. Menurut Kusriani dan Koniyo (2007:171) “Bahasa pemrograman adalah perintah-perintah yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu”. Sedangkan menurut Simarmata (2010:394) “Bahasa pemrograman adalah teknik komando/instruksi standar untuk memerintah komputer”.

Berdasarkan kutipan diatas dapat disimpulkan bahwa pemrograman adalah tatanan atau susunan dalam proses penulisan program.

Dalam pembuatan Sistem Informasi Jasa Pencucian Motor Berbasis Desktop ini, penulis menggunakan bahasa pemrograman *delphi*. *Delphi* adalah suatu bahasa pemrograman yang menggunakan visualisasi sama seperti bahasa pemrograman *Visual Basic* (VB). Namun *Delphi* menggunakan bahasa yang hampir sama dengan *pascal* (sering disebut objek *pascal*) sehingga lebih mudah untuk digunakan.

Menurut Andoyo dan Suyono (2016a:1) mengemukakan bahwa “*Borland delphi* adalah sebuah alat pengembangan aplikasi-aplikasi untuk sistem operasi *microsoft windows*”. Menurut Andoyo dan Suyono (2016b:2) mengemukakan

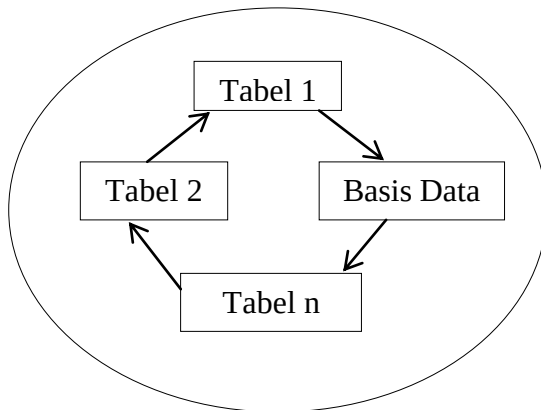
bahwa “*Delphi* merupakan bahasa pemrograman pertama yang memecahkan batasan antara tingkat yang tinggi, dan pengembangan aplikasi dengan cepat (*Rapid Application Development/RAD*)”.

2.1.3. Basis Data

Dalam pembuatan suatu aplikasi sangat dibutuhkan rancangan struktur data/*database* yang nantinya berfungsi untuk menentukan tabel-tabel yang akan kita buat. Menurut Priyadi (2014:2) mengemukakan “Basis data adalah sekumpulan fakta berupa representasi tabel yang saling berhubungan dan disimpan dalam media penyimpanan secara digital”. Sedangkan menurut Kusri (2007:2) “Basis data adalah kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan”.

Berdasarkan uraian diatas dapat disimpulkan bahwa basis data adalah tempat penampungan data-data yang telah diolah dan siap untuk digunakan.

Dalam pembuatan sebuah program tentu nya memerlukan sebuah *database* adapun aplikasi yang dapat digunakan untuk membuat *database* salah satunya adalah *MySQL*. *MySQL* menurut Wahana Komputer (2010:5) adalah “Program database yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. Sedangkan Menurut Anhar (2010:45) mengemukakan “*MySQL* adalah salah satu *Database Management System* (DBMS) dari sekian banyak DBMS seperti *Oracle*, *MS SQL*, *PostgreSQL* dan lainnya yang berfungsi untuk mengolah *database* menggunakan bahasa *SQL* dan bersifat *open source* atau gratis”.



Sumber: Rosa dan Shalahudin (2015:44)

Gambar II.1. Ilustrasi Basis Data

2.1.4. Model Pengembangan Perangkat Lunak

Model *waterfall* sangat dibutuhkan bagi seorang *programmer* yang dimana nantinya model ini bisa membantu dalam penyelesaian suatu program yang sudah dibuat. Model air terjun (*waterfall*) sering juga disebut model sekuensial (*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*). Menurut Rosa dan Shalahudin (2015:28), model *waterfall* memiliki 5 tahapan, yaitu :

1. Analisis kebutuhan perangkat lunak

Pada tahap ini penulis melakukan pengamatan langsung pada tempat pencucian motor untuk mengetahui seberapa besar minat masyarakat pada jasa ini dan dengan besarnya minat masyarakat tersebut maka suatu saat aplikasi ini akan dibutuhkan.

2. Desain

Pada tahap ini desain dilakukan untuk menggambarkan bentuk dari aplikasi yang akan dibangun.

3. Pembuatan kode program

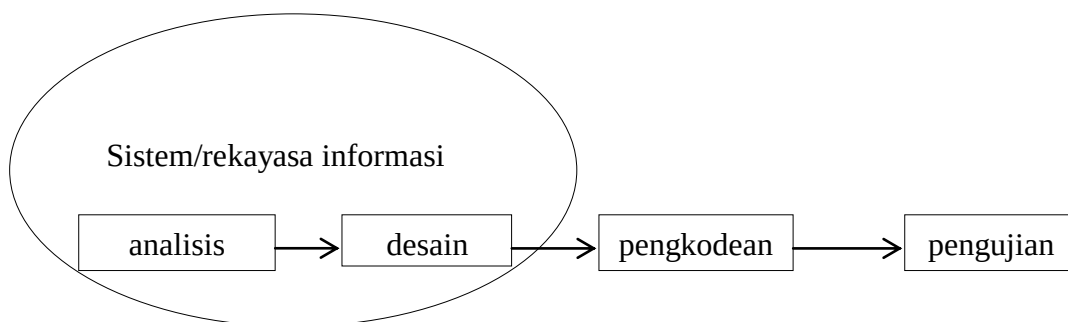
Desain yang telah digambarkan penulis harus dibuatkan kode program agar program yang dibangun dapat dipergunakan. Pembuatan kode program merupakan langkah ketiga yang dilakukan untuk dapat mendukung kinerja *user*. Pembuatan kode program menggunakan bahasa pemrograman *borland delphi*.

4. Pengujian

Pengujian merupakan tahapan dimana program yang akan dibangun harus melewati pengujian kelayakan. Pengujian ini dimaksudkan agar tercipta sebuah program aplikasi yang terhindar dari kesalahan (*error*).

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tahapan pendukung ataupun pemeliharaan ini merupakan tahapan terakhir dalam membangun sebuah aplikasi. Tahapan ini dimaksudkan sebagai pemeliharaan dari aplikasi yang telah dibangun. Tujuannya adalah supaya aplikasi yang dibangun selalu mengalami peningkatan dalam segi kualitas.



Sumber : Rosa dan Shalahudin (2015:29)

Gambar II.2. Model *Waterfall*

2.2. Peralatan Pendukung (*Tools Program*)

Adapun peralatan pendukung (*Tools Program*) yang digunakan dalam pembuatan program ini dijelaskan sebagai berikut :

2.2.1. *Enterprise Relationship Diagram (ERD)*

Sering kita ketahui bahwa ERD atau biasa disebut dengan *entity relationship diagram* merupakan aplikasi pendukung yang nantinya membantu dalam proses pembuatan kode program. Menurut Fatta (2007:121) “ERD adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis”. Sedangkan menurut Mulyani (2016:100) “ERD merupakan *tools* yang digunakan untuk memodelkan struktur data dengan menggambarkan entitas dan hubungan antara entitas (*relationship*) secara abstrak (konseptual)”.

Berdasarkan kutipan diatas, dapat disimpulkan bahwa ERD merupakan alat bantu dalam proses pembuatan *database* dan juga berfungsi untuk merelasikan sebuah tabel.

Adapun komponen *Entity Relationship Diagram (ERD)* menurut Kusri (2007:21), yaitu :

1. Entitas (*Entity*)

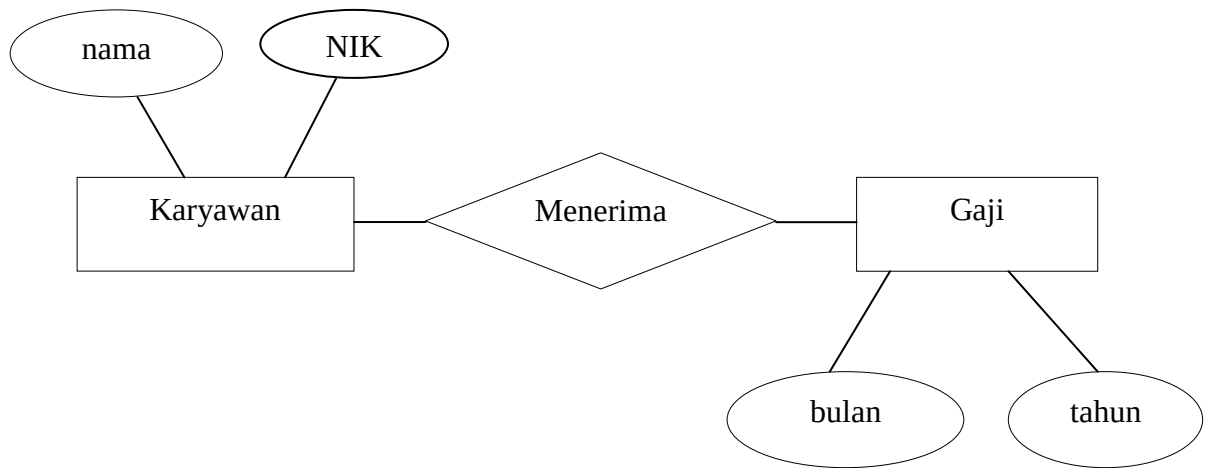
Sekumpulan entiti yang mempunyai tipe yang sama. Misal kumpulan orang yang belajar di perguruan tinggi didefinisikan sebagai mahasiswa.

2. Relasi (*Relationship*)

Relasi adalah hubungan di antara beberapa entiti. *Relationship set* adalah sekumpulan relasi yang mempunyai tipe yang sama.

3. Atribut (*Attribute*)

Atribut merupakan sebutan untuk mewakili suatu entiti.



Sumber : Mulyani (2016:101)

Gambar II.3. Enterprise Relationship Diagram

2.2.2. Logical Record Structure (LRS)

LRS pada umumnya digunakan *programmer* untuk membuat sebuah aturan-aturan yang diterapkan ke dalam *database*. Proses LRS itu sendiri membantu kinerja *programmer* dalam menyelesaikan suatu aplikasi. Menurut Hasugian dan Shidiq (2012:608) “LRS adalah sebuah model sistem yang digambarkan dengan sebuah diagram-ER akan mengikuti pola/aturan pemodelan tertentu dalam kaitannya dengan konvensi ke LRS”.

Aturan-aturan dalam melakukan transformasi E-R diagram ke *Logical Record Structure* menurut Ladjamudin (2013:159) adalah sebagai berikut :

1. Setiap *entity* akan diubah ke bentuk sebuah kotak dengan nama *entity* berada diluar kotak dan atribut berada di dalam kotak.
2. Sebuah relasi kadang disatukan dalam sebuah kotak bernama *entity*, kadang dipisah dalam sebuah tersendiri.

Aturan pokok diatas akan sangat dipengaruhi oleh elemen yang menjadi titik perhatian utama pada langkah transformasi yaitu *cardinality*/kardinalitas menurut Ladjamudin (2013:159) adalah sebagai berikut:

1. 1:1 (*one to one*)

Pada kardinalitas *one to one*, sebaiknya pernah diarahkan ke *entity* dengan jumlah atribut yang lebih sedikit.

2. 1:M (*one to many*)

Pada kardinalitas relasi *one to many*, maka relasi harus dibangun dengan *entity* pada pihak ini yang *many* dan tidak melihat banyak sedikitnya atribut pada *entity* tersebut.

3. M:M (*many to many*)

Pada kardinalitas *many to many*, maka relationship berubah status menjadi file konektor (yang akan berubah kardinalitas *many to many* seolah-olah menjadi *one to many*) sehingga baik *entity* maupun relasi akan menjadi struktur *record* tersendiri.

2.2.3. Pengkodean

Dalam proses pembuatan suatu program kita sering mendengar tentang kata pengkodean yang dimana pengkodean ini merupakan *script* yang membantu dalam pembuatan program karena tanpa *script* kita tidak akan bisa membuat program. Menurut Supranto (2007:82) “Pengkodean merupakan suatu kegiatan pemberian kode atau simbol pada keterangan-keterangan tertentu, kalau pengolahan akan dilakukan dengan komputer elektronik”.

Menurut Kusrini dan Koniyo (2007:23) ada beberapa macam tipe kode dalam kode akun meliputi :

1. Kode *Numerical*

Adalah cara pengkodean akun berdasarkan nomor urut yang dapat dimulai dari angka 1,2,3 dan selanjutnya.

2. Kode Desimal

Adalah cara pemberian kode akun dengan menggunakan lebih dari satu angka. Setiap angka mempunyai makna atau karakter sendiri. Kode desimal dapat dibedakan atas kode kelompok dan kode stelsel akun desimal.

3. Kode *Mnemonic*

Adalah cara pengkodean akun dengan menggunakan huruf tertentu.

4. Kode akun dengan sistem Kombinasi Huruf dan Angka

Adalah dengan cara pengkodean dengan kombinasi antara huruf dan angka.

2.2.4. Hieracky Plus Input-Process-Output (HIPO)

Diagram HIPO ini sangat membantu dalam prosedur pembuatan program. Diagram ini merupakan susunan alur aplikasi dalam proses awalnya program sampai proses program berakhir. Menurut Fatta (2007:150) “HIPO adalah teknik penggambaran modul-modul yang nantinya akan dikembangkan oleh *programmer* menjadi prosedur-prosedur dalam program sistem informasi”. Sedangkan menurut Ladjamudin (2013:211) “HIPO merupakan teknik untuk mendokumentasikan sistem pemrograman”.

Berdasarkan kutipan diatas, dapat disimpulkan bahwa HIPO adalah prosedur-prosedur yang nantinya dikembangkan seorang *programmer* supaya bisa menjadi program baku.

Tingkat Diagram HIPO terdiri dari tiga jenis diagram menurut Ladjamudin (2013:213), yaitu :

1. Daftar Isi Visual (DIV)

DIV merupakan diagram pertama HIPO yang terdiri dari satu atau lebih diagram hirarki.

2. Diagram Ringkas

Diagram ringkas ini merupakan diagram kedua dari paker HIPO yang menjelaskan fungsi dan referensi utama yang diperlukan dalam program detail untuk memperluaskan fungsi sehingga cukup rinci.

3. Diagram Rinci

Diagram rinci HIPO berisikan elemen-elemen dasar sistem, menerangkan fungsi-fungsi khusus, menampilkan item-item *input* dan *output* secara rinci (yaitu nama *field input* yang digunakan dan *output* yang dihasilkan), dan memberikan referensi terhadap diagram HIPO yang lain seperti *flowchart* dan *table* keputusan dari logika yang rumit.

2.2.5. Diagram Alir Program (*Flowchart*)

Dalam proses pembuatan program dibutuhkan suatu diagram yang mendukung kinerja program. Diagram alir data ini merupakan aliran dalam proses berjalannya program.

Menurut Kusrini dan Koniyo (2007:80) “*Flowchart* adalah bagan (*chat*) yang merupakan aliran (*flow*) didalam program atau prosedur sistem secara logika, digunakan terutama sebagai alat bantu komunikasi dan untuk dokumentasi”. Sedangkan menurut Sitorus (2015:14) “*Flowchart* merupakan

langkah-langkah penyelesaian masalah yang dituliskan dalam simbol-simbol tertentu”.

Berdasarkan kutipan diatas, dapat disimpulkan bahwa *flowchart* adalah gambaran umum mengenai sistem berjalan suatu rancangan analisis program untuk menyajikan kegiatan.

1. Program *Flowchart*

Dalam proses pembuatan program dibutuhkan suatu diagram yang mendukung kinerja program. *Flowchart* ini merupakan aliran dalam proses berjalannya program. Menurut Ladjamudin (2013:263) “Program *Flowchart* adalah bagan melihtakan urutan intruksi yang digambarkan dengan simbol tertentu untuk memecahkan masalah dalam suatu program”. Sedangkan menurut Kusrini dan Koniyo (2007:83) “Program *Flowchart* merupakan bagan yang menjelaskan secara rinci langkah-langkah proses program, dibuat dari derivikasi bagan alir sistem”.

2. Sistem *Flowchart*

Pembuatan suatu aplikasi sangat dibutuhkan yang namanya *system flowchart*, yang dimana sistem ini terdiri dari input output dan juga jenis penyimpanan dalam proses pengolahan data. Menurut Ladjamudin (2013:263) “Sistem *flowchart* adalah bagan yang memperlihatkan untuk proses dalam sistem dengan menunjukan alat media input output serta jenis media penyimpanan dalam proses pengolahan data”. Sedangkan menurut Kusrini dan Koniyo (2007:81) “Sistem *flowchart* merupakan bagan yang menunjukan arus pekerjaan dari sistem secara keseluruhan, menjelaskan urutan dari prosedur-prosedur yang ada didalam sistem serta menunjukan apa yang dikerjakan didalam sistem”.

3. Teknik Pembuatan

Dalam penulisan tugas akhir, penulis menggunakan *flowchart* sebagai alat pendukung. Adapun teknik pembuatan program *flowchart*, sebagai berikut :

a. *General Way*

Teknik pembuatan *flowchart* dengan cara ini umum digunakan dalam menyusun logika suatu program yang menggunakan proses secara tidak langsung.

b. *Iteration Way*

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