

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Menurut Hutahaean (2014:2) memberikan batasan bahwa, “Dengan demikian sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan dan berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran tertentu”.

Berikut konsep dasar yang digunakan penulis adalah :

A. Sistem

Berikut sistem yang digunakan penulis untuk penyusunan tugas akhir adalah :

1. Definisi Sistem

Pengertian sistem menurut Romney dan Steinbart (2015:3):

Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar.

2. Definisi Sistem OOP (*Object Oriented Programming*).

Menurut Abdulloh (2018:1) memberikan batasan bahwa, “OOP (*Object Oriented Programming*) merupakan teknik pemrograman dengan menggunakan konsep objek”.

B. Program

Berikut program yang digunakan penulis untuk penyusunan tugas akhir adalah :

1. Konsep Dasar Program

Sebelum lebih dalam mengupas tugas akhir ini, alangkah baiknya apabila diketahui terlebih dahulu definisi serta uraian singkat mengenai konsep dasar dari program yang berhubungan dengan topik.

Yang dimaksud dengan program adalah kumpulan intruksi yang digunakan untuk mengatur komputer agar melakukan suatu tindakan tertentu. Tanpa adanya program komputer hanyalah perangkat keras (hardware) yang tidak bisa melakukan apa-apa.

Asropudin (2013:7), Program dibuat untuk mengerjakan atau menyelesaikan masalah-masalah khusus.

2. Karakteristik Program

Untuk membuat suatu program tahapan-tahapan yang dilakukan penulis menurut Afnisari, Herlawati, dan Merlina (2013:80), adalah :

a. Analisa kebutuhan software.

Suatu proses kebutuhan pengumpulan *software* untuk mengerti sifat-sifat program yang dibentuk *software engineering*.

b. Desain

Desain software sesungguhnya adalah proses *Multi step* yang memfokuskan pada 3 atribut program yang berbeda yaitu struktur data, arsitektur *software* dan rincian prosedur.

c. *Code Generation*

Mendesain apa saja yang diperlukan dalam pembuatan aplikasi ini menggunakan *Microsoft Visual Studio 2010*.

d. *Testing*

Setelah obyek program dihasilkan, pengetesan program dimulai.

e. *Support*

Software pasti mengalami perubahan karena kesalahan yang ditemukan atau karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahan lingkungan *eksternal*, karena pelanggan membutuhkan peningkatan kerja.

C. Basis Data

Menurut Indrajani (2015:34) memberikan batasan bahwa, “Sistem basis data adalah komponen mendasar suatu sistem informasi, dimana pengembangan atau penggunaannya harus dilihat dari prespektif yang lebih luas berdasarkan kebutuhan organisasi”.

Berikut database yang digunakan adalah :

1. MySQL

Menurut jubile *Enterprise* (2014:2) menyimpulkan bahwa, “MySQL Adalah *RDBMS* yang cepat dan mudah digunakan, serta sudah banyak di gunakan untuk berbagai kebutuhan. MySQL dikembangkan oleh MySQL AB Swedia, karena berlisensi open – source, sehingga dapat digunakan secara gratis. Dapat berkerja dengan banyak sistem operasi dan dengan bahasa – bahasa pemograman seperti PHP, PERL, C ,C++, JAVA dan lain lain”.

2. XAMPP

Menurut Alan Nur Aditya (2016:20) memberikan batasan bahwa, “XAMPP adalah sebuah perangkat lunak bebas, yang mendukung banyak sistem operasi, kompilasi dari beberapa program.

D. Model Pengembangan Perangkat Lunak

Menurut Sukamto, dan Shalahuddin (2018:28) memberikan batasan bahwa, “Model perangkat lunak yang digunakan oleh penulis menggunakan model *waterfall*. Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut seperti analisis, desain, pengkodean, pengujian, dan tahapan pendukung”.

Berikut penjelasan model *waterfall* menurut Sukamto dan Shalahuddin (2018:28), yaitu :

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memspifikasikan perangkat lunak agar dapat dipahami perangkat lunak seperti yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk di dokumentasikan.

2. Desain

Desain perangkat lunak adalah proses *multi* langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka dan prosedur pengkodean. Tahap ini mentrasiasi kebutuhan perangkat lunak dari tahap analisa kebutuhan ke representasi desain agar dapat diimplemntasikan menjadi porgram pada tahap selajutnya. Pada tahap ini akan dibuat rancangan sistem seperti desain *Unified*

Modeling Language (UML), *Entity Relationship Diagram (ERD)*. Desain perangkat lunak yang di hasilkan pada tahap ini juga perlu di dokumentasikan.

3. Pembuatan kode program

Desain harus ditrasnsisikan kedalam program perangkat lunak. Hasil dari tahapan ini adalah program komputer sesuai dengan desain yang telah di buat pada tahapan desain. Program atau aplikasi yang digunakan untuk membuat antar muka pengguna menggunakan aplikasi *Visual Studio 2010* dan *MySQL* sebagai databasenya.

4. Pengujian

Pengujian fokus pada perangkat lunak secara 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

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah masuk ke *login*. 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.

2.2. Teori Pendukung

Adapun teori pendukung yang diambil dari buku dan jurnal menjadi bahan dari tugas akhir. Untuk referensi sebagai pemahaman informasi dan untuk sistem berjalan, seperti *Entity Relationship Diagram* (ERD), dan *Unified Modelling Language* (UML).

Berikut beberapa teori pendukungnya adalah :

A. Definisi ERD (*Entity Relationship Diagram*).

Menurut Fridayanthie dan Mahdiati (2016:132) menyimpulkan bahwa, ERD (*Entity Relationship Diagram*) adalah model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model. Didalam hubungan ini tersebut dinyatakan yang utama dari ERD adalah menunjukkan objek data (*Entity*) dan hubungan (*Relationship*), yang ada pada *Entity* berikutnya.

Simbol-simbol dalam ERD (*Entity Relationship Diagram*) menurut Fridayanthie dan Mahdiati (2016:132), adalah sebagai berikut :

1. Entitas

Suatu yang nyata atau abstrak yang mempunyai karakteristik dimana kita akan menyimpan data.

2. Atribut

Ciri umum semua atau sebagian besar instansi pada entitas tertentu.

3. Relasi

Hubungan alamiah yang terjadi antara satu atau lebih entitas.

4. Link

Garis penghubung atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi.

Berikut contoh *Entity Relationship Diagram* (ERD) :



Sumber : Fridayanthie dan Mahdiati (2016:135).

Gambar II.1. Entity Relationship Diagram

B. LRS (*Logical Record Structured*).

Berikut adalah cara membentuk skema *database* atau LRS (*Logical Record Structured*) berdasarkan *Entity Relationship Diagram* menurut Fridayanthie dan Mahdiati (2016:133), yaitu :

1. Jika relasinya satu-ke-satu, maka *foreign key* diletakan pada salah satu dari dua entitas yang ada tau menyatukan kedua entitas tersebut.
2. Jika relasinya satu-ke-banyak, maka *foreign key* diletakan pada entitas *Many*.
3. Jika relasinya banyak-ke-banyak, maka dibua “*file konektor*” yang berisi dua *foreign key* yang berasal dari kedua entitas.

B. Definisi UML (*Unified Modelling Language*).

Menurut Sukamto dan Shalahuddin (2018:133) memberikan batasan bahwa, “UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk medefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek”.

Berikut diagram yang ada pada UML (*Unified Modeling Language*) adalah :

1. *Use Case Diagram*

Menurut Sukamto dan Shalahuddin (2018:155) memberikan batasan bahwa, “*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”.

2. *Activity Diagram*

Menurut Sukamto dan Shalahuddin (2018:161) memberikan batasan bahwa, “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

3. *Class Diagram*

Menurut Sukamto dan Shalahuddin (2018:141) memberikan batasan bahwa, “Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

4. *Sequence Diagram*

Menurut Sukamto dan Shalahuddin (2018:165) memberikan batasan bahwa, “Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek”.