

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Konsep Dasar Sistem**

##### **2.1.1. Visual basic**

Menurut Stefano (2014:2) mengemukakan bahwa “visual basic merupakan sebuah bahasa pemrograman yang menawarkan *Intergated Development Environment* (IDE) visual untuk membuat program perangkat lunak berbasis operasi *Microsoft Windows* menggunakan model pemrograman (COM).”

##### **2.1.2. Basis Data**

###### **1. Pengertian Basis Data**

Menurut Rosa dan Shalahuddin (2018:43), “sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”.

###### **2. Aplikasi Basis Data Microsoft Access**

Aplikasi basis data yang digunakan untuk program penyimpanan benih laboratorium Bank Gen yaitu Microsoft Access.

##### **2.1.3. Model Pengembangan Perangkat Lunak**

Metode yang penulis pilih sebagai dasar untuk menentukan pembuatan aplikasi berdasarkan model pengembangan perangkat lunak yaitu model air terjun

(*waterfall*). Menurut Rosa dan Shalahuddin (2018:28), “Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain pengodean, pengujian dan tahap pendukung (*support*)”.

Adapun metode air terjun menurut Rosa dan Shalahuddin (2018:29) yaitu:

a. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara insentif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

b. 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 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.

c. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

#### d. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi *logic* dan fungsional serta memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

#### e. 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.

### 2.1.4 LRS (Logical Record Structure)

Menurut Simarmata dan Paryudi dalam Fridayanthie dan Tias (2016:132) LRS adalah representasi dari struktur record-record pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas.

## 2.2. Teori Pendukung

### 2.2.1. Entity Relationship Diagram (ERD)

ERD adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan. Menurut Rosa dan Shalahuddin (2018:50) “ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional”.

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 menunjukan objek data (*Entity*) dan hubungan (*Relationship*), yang ada pada *Entity* berikutnya.(Fridayanthie & Mahdiati, 2016).

*Entity Relationship Diagram* (ERD) digunakan untuk menggambarkan hubungan secara logika antar entitas yang terlibat pada suatu sistem *database*.(Abdillah, 2006)

### **2.2.2. Unified Modelling Language (UML)**

UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa dan Shalahuddin, 2018:137).

Jenis- jenis UML adalah sebagai berikut:

#### **1. Use Case Diagram**

Rosa dan Shalahuddin (2018:155), “*use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”.

#### **2. Class Diagram**

Rosa dan Shalahuddin (2018:141), “*class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”. Diagram kelas dibuat agar pembuat program atau *programmer* membuat

kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

Class diagram Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan. (Hendini, 2016)

### 3. *Activity Diagram*

Rosa dan Shalahuddin (2018:161) “*activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Activity diagram menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang, bagaimana masing- masing alir berawal, decision yang mungkin terjadi dan bagaimana mereka berakhir.(Destiana & Fajrin, 2014)

### 4. *Sequence Diagram*

Rosa dan Shalahuddin (2018:165), “diagram sekuen menggambarkan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek”.