

## **BAB II**

### **LANDASAN TEORI**

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

Konsep dasar sistem merupakan gabungan dari komponen atau elemen-elemen yang merupakan penjelasan yang lebih luas dibandingkan sistem yang memprioritaskan prosedur, meliputi beragam aspek dan sudut pandang berbeda-beda sesuai dengan hal-hal yang berhubungan dengan sistem.

##### **2.1.1. Sistem**

(Jogiyanto, 2017) Menulis pada buku yang mempunyai judul Sistem Informasi Manajemen, memberitahukan bahwa, “Sistem merupakan satu kesatuan yang terdapat di dalamnya interaksi subsistem yang mempunyai misi yang sama”.

(Langi et al., 2019) menyatakan bahwa, “Sistem ialah sekelompok yang terdiri dari beberapa bagian dalam struktur berwujud maupun tidak berwujud yang bertautan satu sama lain dan bergerak secara kompak yang mempunyai tujuan tertentu”.

(Muslihudin & Oktafianto, 2016) mengemukakan bahwa, “Sistem adalah kombinasi dari komponen atau jaringan prosedur yang saling bertautan dan berkolaborasi untuk membangun jaringan yang mempunyai sasaran atau tujuan tertentu”.

Dari Beberapa Pandangan di atas maka bisa diambil sebuah kesimpulan bahwa, sistem adalah kombinasi dari komponen atau prosedur yang mempunyai sasaran tujuan tertentu.

### **2.1.2. Informasi**

(Richard Antonio Rompas et al., 2018, p. 222) memberitahukan bahwa, “informasi ialah data yang sudah dikelola dan diolah untuk memberikan makna dalam meningkatkan proses pengutipan keputusan”.

Dari Pandangan diatas maka bisa diambil kesimpulan bahwa informasi ialah data yang sudah dikelola dan diolah kemudian akan bermanfaat oleh pengguna dan dapat meningkatkan kemampuan seseorang yang memakai data tersebut. Selain itu juga bisa menunjang proses pengutipan keputusan.

### **2.1.3. Sistem Informasi**

(Mania et al., 2016) menyatakan bahwa, “Sistem Informasi ialah satu sistem yang diciptakan sama manusia pada dasarnya terdiri dari bagian segmen berbasis pc dan tradisional dibuat untuk menyimpan dan mengatur data serta menyajikan informasi produk kepada para pemakainya”.

(Ganda Yoga Swara & Pebriadi, 2016) mengemukakan bahwa, “Sistem informasi ialah bagian dari penerimaan (*reception*), pencampuran (*mixing*), pengarsipan (*filig*) dan pemakaian (*usage*) informasi yang mempunyai suatu tujuan dari dibuatnya suatu sistem”.

#### **2.1.4. Penerimaan**

Mempermudah proses penerimaan siswa dan memperlancar dalam proses pendataan. Berdasarkan Kamus Besar Bahasa Indonesia, Penerimaan adalah proses, cara, perbuatan menerima; penyambutan: penerimaan tamu; penerimaan murid baru.

#### **2.1.5. Peserta Didik**

Berdasarkan Undang-Undang Dasar Republik Indonesia No.20 Tahun 2003 Perihal Sistem Pendidikan Nasional memberitahukan bahwa “peserta didik atau ialah komponen masyarakat yang berupaya memperbaiki potensi diri melalui siklus pembelajaran yang dapat diakses dengan cara, tingkatan, dan jenis pengajaran tertentu”.

Sementara itu berdasarkan KBBI “peserta didik atau adalah seorang (peserta didik) yang sedang mencari ilmu (berguru, menimba ilmu, belajar) terpenting di jenjang sekolah menengah dan dasar”.

### **2.2. Peralatan Pendukung (*Tools System*)**

#### **2.2.1. *Unified Modeling Language (UML)***

*Unified Modeling Language* bisa disebut juga dengan UML merupakan metode perbaikan sistem yang memakai bahasa realistik sebagai instrument pencatatan dan penetapan pada sistem.

(Mulyani, 2016) memberitahukan bahwa UML pertama kali kali dipopulerkan oleh James Rumbaugh dan Grady Booch tahun 1994 untuk penggabungan dua metodologi terkenal yaitu Booch dan OMT, lalu Ivar Jacobson, yang menciptakan

*Object Oriented Software Engineering* (OOSE) ikut bergabung. Standar UML dioperasikan sama *Object Management Group* (OMG).

### **A. Use Case Diagram**

(Destiningrum & Adrian, 2017) mengemukakan bahwa, *Use Case Diagram* adalah mendemonstrasikan pelaksanaan kerangka data yang akan dibuat.

Berikut ini elemen-elemen yang digunakan pada *use case diagram*.

#### 1. Sistem

Adalah batas proses yang sudah kita gambarkan dalam suatu sistem.

#### 2. Aktor

Adalah bagian pembuat sistem, aktor bisa berupa manusia, alat maupun sistem lain yang berhubungan pada *use case*.

#### 3. Use Case

Adalah bagian proses yang mana potongan pada sistem.

#### 4. Association

Komponen yang menjelaskan kerja sama antara *use case* dan *actor*.

#### 5. Dependency

komponen yang menggambarkan hubungan (koneksi) antara dua kasus penggunaan. Terdapat dua jenis kondisi yang mengkombinasikan dua situasi penggunaan di mana, sebuah *use case* memerlukan *use case* lainnya sedangkan *expands* adalah semacam ketergantungan yang mengaitkan dua situasi penggunaan di mana sebuah *use case* dalam beberapa kasus dapat menghadirkan *use case* lainnya, bergantung terhadap kondisi.

## 6. *Generalization*

Elemen yang mendeskripsikan *use case* akan mewarisi suatu *properties* ke aktor maupun terhadap *use case* yang lainnya maupun pewaris terhadap dua aktor atau *use case* dimana salah satu.

## B. *Activity Diagram*

(Destiningrum & Adrian, 2017) memberitahukan, “*Activity diagram* mendeskripsikan *work flow* (aliran kerja) atau aktivitas dari suatu sistem atau proses bisnis maupun menu yang terdapat pada perangkat lunak”.

(Heriyanto, 2018) mengemukakan, “*Activity Diagram* mendeskripsikan alur kerja (*work flow*) maupun kegiatan pada suatu sistem maupun proses bisnis”.

Yang dapat diperhatikan disini ialah *activity diagram* mendeskripsikan kegiatan sistem bukan yang akan dilakukan oleh aktor, jadi kegiatan dilaksanakan oleh sistem.

Dibawah ini di jelaskan elemen yang terdapat di *activity diagram*.

### 1. *Activities*

Ialah elemen yang dipakai untuk mendeskripsikan aktivitas.

### 2. *Transitions*

Ialah elemen yang dipakai untuk mendeskripsikan peralihan dari elemen yang satu kebagian elemen yang lainnya.

### 3. *Decisions*

Ialah elemen yang dipakai untuk percabangan nalar. sering kita lihat elemen Ini pada *flowchart* lebih-lebih *flowchart* yang dipakai untuk mendeskripsikan suatu algoritma.

#### 4. *Merge point*

Untuk menghubungkan percabangan proses digunakanlah elemen ini. Elemen *decisions* merupakan kebalikan dari elemen ini, apabila *decisions* dipakai untuk percabangan, sementara itu untuk menghubungkan percabangan maka digunakanlah *merge point*.

#### 5. *Start point*

Ialah elemen yang dipakai untuk mengawali *activity diagram*.

#### 6. *End point*

Ialah elemen yang dipakai untuk menutup *activity diagram*.

#### 7. *Concurrency*

Ialah Elemen yang dipakai sebagai percabangan proses (bukan percabangan logika). Proses yang ada didalam elemen ini, bisa dilaksanakan secara Acak (tidak berurutan).

#### 8. *Synchronization*

Ialah Elemen yang dipakai untuk menghubungkan proses yang dibagi oleh *concurrency*.

#### 9. *Swimline*

Ialah elemen yang dipakai untuk membagi antara aktor dan sistem maupun antara aktor yang satu dengan aktor yang lain, atau sebaliknya pada suatu sistem dengan sistem yang lainnya.

### **C. *Sequence Diagram***

(Hendini, 2016) mengungkapkan bahwa, *Sequence Diagram* mendeskripsikan kegiatan suatu objek di *use case* menggunakan cara menggambarkan batas

hidup suatu objek juga pesan yang dikirim lalu diterima antara objek satu dengan yang lainnya.

Haviluddin dalam (Suendri, 2018, p. 3) mengungkapkan, *sequence diagram* ialah deskripsi fase demi fase, termasuk perkembangan (tahapan) peralihan secara *logic* yang semestinya dilakukan untuk membentuk *use case diagram* dengan sepantasnya.

#### **D. Class Diagram**

(Destiningrum & Adrian, 2017) mengemukakan bahwa, *Class diagram* mendeskripsikan kondisi sistem fungsi-fungsi dan kebutuhan yang akan bertautan dengan bagian utama dan koneksi database.

Whitten (Suendri, 2018) menyatakan bahwa, Kelas mempunyai atribut dan perilaku yang sama seperti sebuah set objek, terkadang kelas juga disebut kelas objek. *Class* mempunyai tiga sektor pokok yaitu :

1. Nama, kelas atau kelas objek wajib memiliki nama.
2. Atribut, ialah perlengkapan yang terdapat di kelas. Suatu kelas mempunyai sebuah nilai bahwa kelas dapat diolah hanya sampai atribut yang dipunyai.
3. Operasi, ialah pengolahan yang bisa kelas lakukan, yang bisa kelas lakukan itu sendiri maupun terhadap kelas yang lain.

#### **E. Deployment Diagram**

(Andini & Pratiknyo, 2017) mengemukakan bahwa, *Deployment diagram* memberitahukan tata letak suatu sistem dalam bentuk berwujud, menampakan komponen-komponen *software* yang terdapat di komponen-komponen *hardware*.

Rossa dan Shalahuddin (Ermawati et al., 2018) mengemukakan bahwa, *diagram deployment* atau *deployment diagram* memberitahukan susunan komponen yang terdapat di proses menjalankan aplikasi tersebut.

### **2.2.2. Perancangan Database.**

#### **A. Database**

(Ganda Yoga Swara & Pebriadi, 2016) menyatakan bahwa, Basis data atau Database adalah gabungan informasi yang tertata juga suatu kesatuan yang tersimpan di komputer (perangkat keras) secara terstruktur sehingga perangkat lunak yang dapat diolah dan digunakan.

#### **B. *Entity Relationship Diagram (ERD)***

Menurut Marmata (Ramdhani et al., 2019) mengemukakan bahwa, *Entity Relationship Diagram (ERD)* ialah alat pembentuk data primer dan dapat membantu mengelola data pada suatu proyek ke bagian *entity-entity* dan menyimpulkan interaksi antar *entity*. Proses membolehkan analisis menyimpan juga diambil secara efisien dalam mendapatkan struktur basis data.

(Faizal & Putri, 2017) mengemukakan bahwa *Entity Relationship Diagram* ialah sebuah teknik untuk mendeskripsikan informasi yang diperlukan dalam sistem, dan ikatan antara data-data tersebut.

#### **C. *Logical Record Structured (LRS)***

(Larasati, H., & Masripah, 2017) menyatakan bahwa, *Logical Record Structured (LRS)* merupakan suatu bentuk sistem yang dideskripsikan pada suatu diagram-ER tentu menyertai pola maupun hukum permodelan tertentu yang ada kaitannya terhadap pedoman ke LRS.

Pandangan Simarmata dan Pahyudi (Ermawati et al., 2018) menyatakan bahwa, *Logical Record Structured (LRS)* ialah perwakilan terhadap susunan *record-record* yang ada di tabel-tabel yang tercipta karena hubungan pada himpunan entitas”. Memastikan kardinalitas, *Foreign Key (FK)* dan total table.

#### **D. MySQL**

Adi Nugroho dalam (Destiningrum & Adrian, 2017): “*My Structured Query Language (MySQL)* merupakan *Relational Database managemnt System (RDBMS)* maupun satu sistem basis data *relation* yang dapat bekerja secara cepat dan mudah digunakan.

#### **E. XAMPP**

(IdCloudHost, 2020) XAMPP adalah free software (pemrograman gratis), yang merupakan kumpulan sebagian program, sistem operasi banyak didukungnya. Kapasitas XAMPP sendiri adalah sebagai server yang independent worker (localhost), yang terdapat bagian-bagian program diantaranya : Apache HTTP Server, penerjemah bahasa yang dicatat menggunakan bahasa pemrograman PHP, Perl MySQL database.