

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

A. Perancangan

Menurut Nafisah (2013:2) “Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi sebagai perancangan sistem dapat dirancang dalam bentuk bagan alir sistem (system flowchart), yang merupakan alat bentuk grafik yang dapat digunakan untuk menunjukkan urutan-urutan proses dari sistem”

B. Sistem

Sistem merupakan kumpulan sub-sub sistem (elemen) yang saling berkolerasi satu dengan yang lainnya untuk mencapai tujuan tertentu (Latif, Mohamad & Said, 2018:1). Pendapat lain dikemukakan oleh Hutahaen (2014:2) bahwa sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau melakukan sasaran tertentu. Sistem dikatakan baik apabila memiliki karakteristik yaitu: komponen, batasan, lingkungan luar, penghubung, masukan, keluaran, pengolah dan sasaran sistem.

C. Sistem Informasi

Sistem informasi menurut Arif (2019:7) adalah “kesatuan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya”.

D. Penjualan

Wijaya (2011:92) menjelaskan bahwa “penjualan adalah sebuah transaksi pendapatan barang atau jasa yang dikirim pelanggan untuk imbalan kas suatu kewajiban untuk membayar”.

E. Website

Website adalah kumpulan informasi atau kumpulan *page* yang biasa diakses lewat jalur internet, dengan kata lain *website* ialah sekumpulan halaman yang terdiri dari beberapa laman yang informasi dalam bentuk data digital (Junirianto, 2018:1).

F. Database

Menurut (Indrajani, 2015) mendefinisikan sebuah basis data merupakan kumpulan data yang saling berhubungan secara logis, dan merupakandata yang dibutuhkan oleh sebuah organisasi. Kadir (2013:411) mengemukakan bahwa “*Database* adalah istilah yang menyatakan kumpulan data yang disimpan dalam bentuk yang mudah diakses oleh pemakai”.

Pendapat lain diutarakan Wayteg (2013:1) bahwa “*Database* merupakan sekumpulan data yang berhubungan secara logis beserta deskripsinya, yang digunakan secara bersama-sama dan dirancang untuk memenuhi kebutuhan informasi di suatu tempat”. Dalam database, semua data diintegrasikan untuk menghindari adanya duplikasi data.

G. Model Pengembangan Perangkat Lunak

Model pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah *prototype*. Menurut (Yurindra, 2017:25), *prototype* adalah “suatu proses yang memungkinkan *developer* membuat sebuah model *software*”. Secara ideal *prototype* berfungsi sebagai mekanisme untuk mengidentifikasi kebutuhan-kebutuhan *software*.

2.2. Teori Pendukung

A. Entity Relationship Diagram

Sutanta (2012:91) berpendapat bahwa, “*Entity Relationship Diagram (ERD)* merupakan suatu model data yang dikembangkan berdasarkan objek.” *Entity*

Relationship Diagram (ERD) digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis. *Entity Relationship Diagram* (ERD) didasarkan pada suatu persepsi bahwa real world terdiri atas obyek-obyek dasar tersebut. Penggunaan *Entity Relationship Diagram* (ERD) relatif mudah dipahami, bahkan oleh para pengguna yang awam. Bagi perancang atau analis sistem, *Entity Relationship Diagram* (ERD) berguna untuk memodelkan sistem yang nantinya, basis data akan di kembangkan. Model ini juga membantu perancang atau analis sistem pada saat melakukan analisis dan perancangan basis data karena model ini dapat menunjukkan macam data yang dibutuhkan dan kereliasian antar data didalamnya.

Komponen *Entity Relationship Diagram* menurut Sutanta (2012:91) adalah sebagai berikut:

a. Entitas

Entitas merupakan suatu objek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Objek dasar dapat berupa orang, benda, atau hal yang keterangannya perlu disimpan didalam basis data. Untuk menggambarkan sebuah entitas digunakan aturan sebagai berikut: entitas dinyatakan dengan simbol persegi panjang, nama entitas dituliskan didalam simbol persegi panjang, nama entitas berupa kata benda, tunggal dan sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

b. Atribut

Atribut merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan dalam basis data. Atribut berfungsi sebagai penjelas pada sebuah entitas. Untuk menggambarkan atribut digunakan aturan sebagai

berikut: atribut digambarkan dengan simbol ellips, nama atribut dituliskan didalam simbol ellips, nama atribut merupakan kata benda, tunggal dan sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

c. Relasi

Relasi merupakan hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Aturan penggambaran relasi adalah sebagai berikut:

- 1) Relasi dinyatakan dengan simbol belah ketupat
- 2) Nama relasi dituliskan didalam simbol belah ketupat
- 3) Nama relasi berupa kata kerja aktif dan sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

B. *Logical Record Structure*

Logical record structure terdiri dari *link-link* diantara tipe *record*, *link* ini menunjukkan arah dari satu tipe *record* lainnya. Riyanto (2015:22) mendefinisikan bahwa “LRS (*logical record structure*) adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas”. *Logical record structure* dibentuk dengan nomor dari tipe *record*.

Banyak *link* dari LRS yang diberi tanda *field-field* yang kelihatan pada kedua *link* tipe *record*, penggambaran LRS mulai dengan menggunakan model yang dimengerti, dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke LRS.

Metode yang lain dimulai dengan ER-diagram dan langsung dikonversikan ke LRS.

- a. *One to One (1-1)* Tingkat hubungan ini menunjukkan hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, dan hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.
- b. *One to Many (1-M)* Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu, tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya, satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.
- c. *Many to Many (M-M)* Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, dilihat dari sisi entitas yang pertama maupun dilihat dari sisi yang kedua.

C. *Unified Modelling Language*

Rosa dan Shalahuddin (2014:133) menjelaskan bahwa 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”. Dalam laporan ini, terdapat beberapa diagram yang digunakan diantaranya *use case diagram*, *activity diagram*, *component diagram* dan *deployment diagram*.

a. *Use Case Diagram*

Use Case adalah cara spesifik penggunaan sistem dengan menggunakan beberapa bagian dari fungsi tersebut. Urutan khusus yang terkait dilakukan

oleh *actor* dalam sistem. Use case adalah suatu pola atau gambaran yang menunjukkan kelakuan suatu sistem. Setiap *use case* adalah suatu rentetan transaksi yang saling berhubungan dan dilakukan oleh *actor* dan sistem dalam sebuah bentuk dialog. *Use case* diagram dibuat untuk menggambarkan hubungan antara *actor* dan *use case*. *Use case* diagram mempresentasikan kegunaan atau fungsi-fungsi sistem dari sudut pandang pengguna.

b. *Activity Diagram*

Suatu diagram yang menggambarkan aliran kegiatan baik dalam sistem, bisnis, alur kerja atau yang lainnya. Diagram ini fokus pada kegiatan-kegiatan yang dilakukan dan siapa yang berkepentingan atas kegiatan itu. Elemen-elemen/*nodes* dalam *Activity* diagram diantaranya *node action*, *node control* dan *object node*.

c. *Component Diagram*

Component diagram adalah Sebuah jenis klasifikasi terstruktur dan bekerjasama dengan komponen lain melalui *interface* yang terdefinisi dengan baik untuk menyajikan fungsi sebuah sistem.

d. *Deployment Diagram*

Deployment diagram adalah diagram yang digunakan untuk menampilkan alokasi artefak untuk *node* dalam sebuah desain fisik sistem dan digunakan untuk menunjukan koleksi *node* dan berfungsi sebagai *platform* dari sistem yang akan di kembangkan.