

## BAB III

### LANDASAN TEORI

#### 3.1 Rancang Bangun

Menurut Jogiyanto (2005), Rancang Bangun (desain) adalah tahap dari setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhankebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem.

#### 3.2 Aplikasi

Aplikasi adalah suatu sistem yang dirancang dan disusun sedemikian rupa untuk menghasilkan informasi yang terpadu dengan menggunakan komputer sebagai sarana penunjang. Dalam penerapannya aplikasi terdiri dari beberapa bagian, yaitu (Jogiyanto, 2005):

1. *Input*, menggambarkan suatu kegiatan untuk menyediakan data yang akan diproses.
2. *Proses*, menggambarkan bagaimana suatu data diolah untuk menghasilkan suatu informasi yang bernilai tambah.
3. *Output*, kegiatan untuk menghasilkan laporan dari proses diatas.
4. *Penyimpanan*, kegiatan untuk memelihara dan menyimpan data.

5. Kontrol, aktifitas untuk menjamin bahwa sistem informasi tersebut berjalan sesuai dengan yang diharapkan.

### 3.3 Data

Data adalah bahan keterangan yang berupa himpunan fakta-fakta, angka-angka, huruf-huruf, kata-kata, grafik, tabel, gambar, dan lambang-lambang yang menyatakan sesuatu pemikiran, objek, koordinasi, dan situasi. (Siska, 2013)

Data adalah fakta atau keterangan yang dapat dipertanggungjawabkan kebenarannya. Sebagian besar tujuan penelitian adalah memperoleh data yang relevan, dapat dipercaya, dan dapat dipertanggungjawabkan. (Sugiyono, 2000)

Data merupakan deskripsi tentang benda, kejadian, aktivitas, dan transaksi yang tidak mempunyai makna atau tidak berpengaruh secara langsung kepada pemakai. Data juga dapat diartikan suatu bahan mentah yang kelak dapat diolah lebih lanjut untuk menjadi sesuatu yang lebih bermakna. Dan data inilah yang nantinya akan disimpan dalam database. (Kadir, 2009)

Ada beberapa jenis data dilihat dari berbagai sisi, seperti (Siska, 2013):

1. Menurut cara memperolehnya:
  - a. Data primer, yaitu data yang dikumpulkan dari tangan pertama dan diolah oleh organisasi atau perorangan.
  - b. Data sekunder, yaitu data yang diperoleh suatu organisasi atau perorangan yang diperoleh dari pihak lain yang telah mengumpulkan dan mengolahnya.
2. Menurut sifatnya:
  - a. Data kualitatif, yaitu data yang tidak berbentuk angka.

- b. Data kuantitatif, yaitu data yang berbentuk angka.
3. Menurut sumbernya:
  - a. Data internal, yaitu data yang menggambarkan keadaan di dalam suatu organisasi seperti negara, perusahaan, departemen, dan lain-lain.
  - b. Data eksternal, yaitu data yang menggambarkan sesuatu di luar organisasi.
4. menurut waktu pengumpulannya:
  - a. *Cross-section* data, yaitu data yang dikumpulkan pada waktu tertentu untuk menggambarkan keadaan waktu itu.
  - b. *time-series* data, yaitu data yang dikumpulkan dari waktu ke waktu untuk menggambarkan suatu perkembangan atau pertumbuhan.

### 3.4 Pegawai

Pegawai adalah orang pribadi yang bekerja pada pemberi kerja, berdasarkan perjanjian atau kesepakatan kerja baik secara tertulis maupun tidak tertulis, untuk melaksanakan suatu pekerjaan dalam jabatan atau kegiatan tertentu dengan memperoleh imbalan yang dibayarkan berdasarkan periode tertentu, penyelesaian pekerjaan, atau ketentuan lain yang ditetapkan pemberi kerja, termasuk orang pribadi yang melakukan pekerjaan dalam jabatan negeri. (Wibowo, 2014)

### 3.5 *Undefined Modeling Language (UML)*

*Unified Modeling Language* merupakan salah satu alat bantu yang dapat digunakan dalam bahasa pemrograman yang berorientasi objek, saat ini UML akan mulai menjadi standar masa depan bagi industri pengembangan sistem atau

perangkat lunak yang berorientasi objek sebab pada dasarnya UML digunakan oleh banyak perusahaan raksasa seperti IBM, Microsoft, dan sebagainya. Notasi UML dibuat sebagai kolaborasi dari Grady Booch, DR. James Rumbaugh, Ivar Jacobson, Rebecca Wirfs-Brock, Peter Yourdon, dan lainnya. Penggabungan beberapa metode menjadi UML dimulai tahun 1993. (Arie, 2013)

### 3.5.1 Diagram *Use case*

Diagram *use case* menyajikan interaksi antara *use case* dan aktor dalam sistem yang akan dikembangkan. *Use case* sendiri adalah fungsionalitas atau persyaratan-persyaratan sistem yang harus dipenuhi oleh sistem yang akan dikembangkan tersebut menurut pandangan pemakai sistem sedangkan aktor bisa berupa orang, peralatan atau sistem lain yang berinteraksi terhadap sistem yang akan dibangun (Sholih, 2010). Elemen-elemen yang digunakan dalam pemodelan *use case* sistem yaitu:

1. Aktor

Aktor merupakan semua yang ada di luar ruang lingkup sistem perangkat lunak dan berinteraksi dengan sistem perangkat lunak tersebut (Sholih, 2010).

2. Notasi *Use Case*

*Use case* digunakan untuk menyatakan bagian tingkat tinggi fungsionalitas yang disediakan oleh sistem. Dengan kata lain, *use case* menggambarkan bagaimana seseorang sebagai pengguna berinteraksi dengan sistem (Sholih, 2010).

### 3. Relasi

Relasi digunakan untuk membentuk diagram *use case* agar saling berhubungan. Secara umum relasi diagram dalam *use case* terbagi menjadi tiga macam yaitu :

- a. Relasi antara aktor dengan *use case* yang disebut relasi asosiasi.
- b. Relasi antara *use case* dan *use case* lainnya ada tiga bentuk yaitu relasi *include*, *relasi extend*, dan relasi generalisasi.
- c. Relasi antara aktor hanya digunakan untuk satu relasi yaitu relasi generalisasi (Sholih, 2010).

#### 3.5.2 *Flow Of Events*

*Flow of events* digunakan untuk mendokumentasikan aliran logika dalam *use case*, yang menjelaskan secara rinci apa yang pemakai akan lakukan dan apa yang sistem itu sendiri lakukan. Namun pada bagian ini tidak dijelaskan apakah sistem dibangun menggunakan bahasa pemrograman Java, *Visual Basic*, atau PHP. (Sholih, 2010)

#### 3.5.3 *Sequence Diagram*

Diagram sekuensial (*sequence diagram*) adalah diagram yang digunakan untuk menunjukkan alur (*flows*) fungsionalitas yang melalui sebuah *use case* yang disusun dalam urutan waktu (Sholih, 2010). Setiap diagram sekuensial merepresentasikan satu *flow* dari beberapa *flow* di dalam *use case*.

### 3.5.4 *Class Diagram*

Diagram kelas menunjukkan interaksi antar kelas-kelas di dalam sistem. Kelas juga dapat dianggap sebagai *blueprints* dari objek-objek di dalam sistem. Sebuah kelas dibuat dalam bentuk bujur sangkar yang terbagi dalam tiga bagian yaitu (Sholih: 2010):

1. Bagian pertama menunjukkan nama kelas.
2. Bagian kedua menunjukkan anggota kelas yang memuat informasi atau atribut.
3. Bagian ketiga menunjukkan operasi-operasi dari sebuah kelas. Operasi dari sebuah kelas adalah tingkah laku yang disediakan oleh kelas.

