

BAB II

LANDASAN TEORI

A. Sistem Informasi

Sistem informasi manajemen (SIM) berfungsi untuk merencanakan dan mengendalikan sekumpulan operasi. Seiring berkembangnya jaman, (SIM) mempunyai alat untuk mengimplementasikan operasi berbasis teknologi yaitu komputer. Komputer memiliki peranan penting untuk meningkatkan kecepatan, ketelitian, dan volume data selain itu dapat memberikan beberapa alternatif dalam suatu keputusan yang di dalam suatu organisasi terdiri atas jumlah unsur, orang yang mempunyai bermacam-macam peran dalam organisasi, kegiatan atau tugas yang harus diselesaikan, tempat kerja, wewenang pekerjaan, serta hubungan komunikasi yang mengikat bersama organisasi tersebut (Sutabri, 2012). Jadi, sistem informasi manajemen (SIM) merupakan penerapan sistem informasi di dalam organisasi untuk mendukung informasi-informasi yang dibutuhkan untuk semua tingkatan manajemen

B. Monitoring

Monitoring adalah pemantauan yang dapat dijelaskan sebagai kesadaran (*awareness*) tentang apa yang ingin diketahui untuk mencapai suatu tujuan. *Monitoring* akan memberikan informasi tentang status dan kecenderungan bahwa pengukuran dan evaluasi yang diselesaikan berulang dari waktu ke waktu (Wikipedia Contributor, Monitoring).

Monitoring menyediakan data mentah untuk menjawab pertanyaan sedangkan evaluasi adalah meletakkan data-data tersebut agar dapat digunakan dan dengan demikian memberikan nilai tambah. Evaluasi adalah tempat belajar kejadian, pertanyaan yang perlu dijawab, rekomendasi yang harus dibuat, menyarankan perbaikan.

Menurut Mardiani (2013: 36), tujuan *monitoring* adalah:

1. Mengkaji apakah kegiatan-kegiatan yang telah dilaksanakan sesuai dengan rencana.
2. Mengidentifikasi masalah yang timbul agar langsung dapat diatasi.
3. Melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan kegiatan.
4. Mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan.
5. Menyesuaikan kegiatan dengan lingkungan yang berubah, tanpa menyimpang dari tujuan.

C. Website

Menurut Hidayat (2010), *website* atau situs dapat diartikan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam ataupun bergerak, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis. Perkembangan dunia *website* pada saat ini lebih mengarah pada pengelolaan *content* sebuah *website*.

D. Basis Data

Menurut Yanto (2016), *basis data* adalah suatu kumpulan data yang memberikan fakta mewakili suatu objek seperti manusia, barang, hewan, peristiwa, keadaan dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya. *Basis data* dapat didefinisikan dalam beberapa sudut pandang seperti:

- a. Himpunan kelompok data yang saling berhubungan dan diorganisasi sedemikian rupa agar dapat dimanfaatkan kembali dengan cepat dan mudah.
- b. Kumpulan data yang saling berhubungan dan disimpan secara bersama dan tanpa pengulangan, untuk memenuhi berbagai kebutuhan.
- c. Kumpulan *file* saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Basis data tidak sekedar penyimpanan secara elektronik karena:

- a. Pada penyimpanan dokumen berisi data dalam file teks, file *Spread Sheet*, tidak ada pemilihan dan pengelompokan data sesuai jenis atau fungsi sehingga akan menyulitkan pencarian data.
- b. Keutamaan *basis data* adalah pengaturan, pemilihan, pengelompokan, pengoperasian data yang akan disimpan sesuai fungsi dan jenisnya.

E. MySQL

MySQL adalah sebuah manajemen sistem *database server* yang mampu menangani beberapa *user*, yaitu mampu menangani beberapa instruksi sekaligus dari beberapa *user* dalam satu waktu selain itu merekam semua data *user* di dalam

sistem dalam tabel *user* (Komputer: 2010). Keunggulan *MySQL* dapat menyediakan berbagai fasilitas atau fitur-fitur yang dapat digunakan oleh bermacam-macam *user*. *User* ini sendiri termasuk administrator *database*, *programmer* aplikasi, *manager*, sampai *end user* (pemakai akhir). Pada mulanya *MySQL* bekerja pada *platform unix* dan *linux*. Namun, dengan perkembangannya, sekarang banyak bermunculan beberapa *distro* yang mampu berjalan pada beberapa *platform* yang bersifat *share-ware* dan *corporate*.

F. PHP: *Hypertext Preprocessor*

Menurut Pratama (2010), PHP merupakan suatu bahasa pemrograman sisi *server* yang dapat digunakan untuk membuat halaman *web dinamis*. PHP: *HyperText Preprocessor* dibangun oleh Rasmus Lerdorf pada tahun 1994. Dahulu, pada awal pengembangannya PHP disebut sebagai *Personal Home Page*. PHP merupakan produk *Open Source* sehingga dapat mengakses *source code*, menggunakan, dan mengubahnya tanpa harus membayar. Selain PHP bahasa pemrograman yang biasa digunakan adalah *Microsoft Active Server Page (ASP)* dan *Java Server Page (JSP)*. Dalam suatu halaman HTML dapat menanamkan kode PHP yang akan dieksekusi setiap kali halaman tersebut dikunjungi. Karena kekayaannya akan fitur yang mempermudah perancangan dan pemrograman.

G. *CodeIgniter*

CodeIgniter adalah sebuah *framework* bahasa pemrograman PHP. *CodeIgniter* menawarkan kemudahan serta standarisasi dalam proses

pengembangan *website* menjadi lebih cepat dan berstandar. Standar ini berguna dalam pengembangan *website* yang dikerjakan oleh tim. *CodeIgniter* telah menyediakan *library* dan *helper* yang berguna dan mempermudah proses *development*. Meski telah disediakan beragam komponen dan *library*, juga dapat membuat komponen atau *library* sendiri (Komputer: 2014).

H. XAMPP

Menurut Prasetya (2013), ada banyak pilihan perangkat lunak *web server* yang bisa digunakan. Untuk memudahkan pekerjaan, bisa memanfaatkan paket *bundling* yang praktis, salah satunya adalah XAMPP. Paket aplikasi XAMPP hingga saat ini tersedia untuk empat lingkungan sistem operasi yang berbeda, yaitu *Linux*, *Windows*, *Mac OS X*, dan *Solaris*. Menariknya lagi, distribusi XAMPP ini bisa didapatkan secara gratis dan mudah. Instalasi dan penggunaan paket aplikasi XAMPP sangat mudah, tidak perlu melakukan pengaturan apapun, tinggal *download*, ekstrak, dan jalankan. Hal yang perlu diperhatikan hanya memastikan lokasi instalasi *default* (C:\xampp) untuk memudahkan pengaksesan *file* nantinya.

I. Text Editor

Text editor adalah program yang digunakan membuat dokumen HTML. Contoh HTML *editor* yang digunakan diantaranya: *Ms FrontPage*, *Dreamweaver*, *Notepad*, *Notepad++* (Pamungkas, 2017).

Menurut Abdulloh (2016), *text editor* digunakan untuk menuliskan skrip HTML, CSS, PHP, dan *Javascript*. Ada banyak *text editor* yang dapat digunakan, diantaranya *Notepad*, *Notepad++*, *Adobe Dreamweaver*, *Sublime*, dan sebagainya.

J. Web Browser

Menurut Oktavian (2010), yang berperan penting dalam menampilkan informasi di layar *monitor* adalah *web browser*. Banyak program *web browser* yang tersedia, namun yang sering digunakan adalah: *Mozilla Firefox*, *Internet Explorer*, *Opera*, dan *Google Chrome*. Fungsi dari *web browser* adalah menerjemahkan kode (*script*) HTML yang biasanya juga ditambahkan *JavaScript*, menjadi sebuah informasi yang sudah tertata dan menarik secara visual. Untuk menjalankan *web* dari sebuah *server*, cukup memanggil alamat *web* tersebut, lalu *browser* akan memunculkan halaman *web* yang dimaksud sesuai dengan alamat yang diberikan. Proses yang terjadi, *web browser* meminta kepada *web server*, kemudian *web server* akan membaca *file index* (missal: *index.html*, *index.php*, *index.asp*), kemudian data-data dari *file index* tersebut diterjemahkan dan dikirim kembali ke *web browser*. Selanjutnya *web browser* akan menerjemahkan kode-kode HTML yang menjadi halaman *web*.

K. UML (*Unified Modeling Language*)

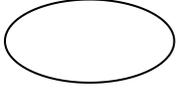
Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah Bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem (Gata, 2013).

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *Use case diagram*

Use case diagram merupakan pemodelan untuk perilaku (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case diagram*, yaitu:

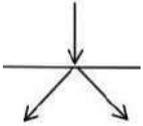
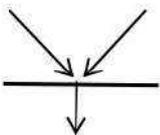
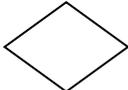
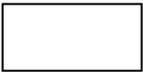
Tabel 2.1. Simbol *Use Case*

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antara unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain dengan mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan tidak mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> digambarkan dengan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

2. Activity diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* yaitu:

Tabel 2.2. Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	<i>End point</i> , akhir aktivitas.
	<i>Activities</i> , menggambarkan suatu proses atau kegiatan bisnis.
	<i>Fork</i> (percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

3. 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 atas entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau *kardinality*.

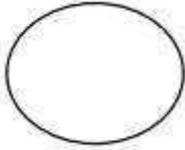
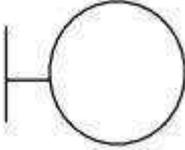
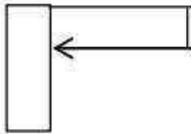
Tabel 2.3. Multiplicity Class Diagram

<i>Multiplicity</i>	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4

4. *Sequence diagram*

Sequence diagram menggambarkan perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* yaitu:

Tabel 2.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i> .
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .