BAB II

LANDASAN TEORI

Landasan teori merupakan suatu hal penting dalam membuat tulisan ilmiah, karena landasan teori digunakan untuk mengaitkan permasalahan dengan teori-teori yang sudah ada untuk mencapai tujuan yang dimaksud.

2.1 Sistem Informasi

2.1.1 Pengertian Sistem

Pada umumnya sistem diartikan sebagai sekumpulan unsur atau elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegitan bersama untuk mencapai satu tujuan. Kata sistem itu sendiri berasala dari kata Latin (*Systema*) dan bahasa Yunani (*sustema*) yang memiliki arti suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran indormasi, materi atau energi untuk mencapai satu tujuan. Dalam mendefinisikan sistem para ahli memiliki pendapat yang berbeda, beberapa diantaranya yaitu :

1. "Sistem adalah suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung satu sama lain." (Fata, 2007) dalam (Fitriyani, 2011)
2. "Sistem merupakan kumpulan elemen yang saling berkaitan dan bekerja sama untuk memproses masukan (*input*) yang ditujukan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (*output*) yang diinginkan" (Kristanto, 2003) dalam (Fitriyani, 2011). Sehingga dapat disimpulkan bahwa sistem merupakan sekumpulan

elemen-elemen yang salaing terkait dan bekerja sama untuk memproses input dan menghasikan output yang diinginkan.

3. Masukan Sistem (*Input*)

   Merupakan energi yang dimasukkan ke dalam sistem. Masukan ini dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

4. Keluaran Sistem (*Output)*

   Merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Artinya keluaran dapat digunakan sebagai masukan untuk subsistem yang lain.

2.1.2 Pengertian Informasi

Sebelum mengetahui apa itu informasi, terlebih dahulu harus memahami apa itu data. Data merupakan nilai, keadaan atau sifat yang berdiri sendiri lepas dari konteks apapun (Fatta, 2007) dalam (Fitriyani, 2011).

Sehingga dapat disimpulkan bahwa data itu adalah angka karakter yang mempunyai arti tertentu. Informasi adalah data yang telah diubah bentuknya dan diorganisir oleh pemrosesan dengan tujuan yang spesifik (Whitten, 2004) dalam (Fitriyani, 2011).

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Hartono, 2005) dalam (Fitriyani, 2011).

Jadi informasi pada awalnya adalah data. Data tersebut diproses, menghasilkan keluaran yang kita sebut sebagai informasi sehingga berguna bagi pihak-pihak tertentu. Pemrosesan data ini dilakukan secara terstruktur dan tersusun sedemikian rupa sehingga dapat digunakan sebagai dasar untuk pengambilan keputusan dalam organisasi. Informasi adalah data yang telah diolah atau diproses sehingga tersusun dari kombinasi-kombinasi data yang diinginkan dan dapat memberikan arti tertentu bagi penerimanya.

2.1.3 Pengertian Sistem Informasi

Istilah sistem informasi biasanya berhubungan dengan sistem yang digunakan dalam suatu organisasi dan mencerminkan rangkaian proses yang dikerjakan untuk mengolah data dan menghasilkan informasi, seperti sistem informasi penjualan, sistem informasi pembelian, sistem informasi penggajian, dan sebagainya. Berikut ini beberapa pengertian sistem informasi menurut para ahli :

1. ”Sistem informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna bagi penerimanya.” (Kristanto, 2003) dalam (Fitriyani, 2011)
2. Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Hartono, 2005) dalam (Fitriyani, 2011).

Dari pengertian di atas, dapat disimpulkan bahwa sistem informasi adalah sistem yang digunakan dalam suatu organisasi atau perusahaan untuk mengolah data sehingga menghasilkan informasi.

2.3 Pengertian *Unified Modeling Language*

UML digunakan untuk menggambarkan perancangan awal dari sistem yang akan dibangun. UML memiliki banyak jenis permodelan. Berikut ini definisi *Unified Modeling Language* (UML) menurut para ahli :

1. "UML (*Unified Modeling Language*) adalah sebuah bahsa untuk menspesifikasikan, memvisualisasikan, serta mengkonsturksi bangunan dasar dari sistem perangkat lunak termasuk didalamnya dengan melibatkan pemodelan aturan- aturan bisnis" (Nugroho, 2004) dalam (Fitriyani, 2011).
2. "*Unified Modeling Language* (UML) adalah bahasa yang telah menjadi standard untuk visualisasi, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak" (Hend, 2006) dalam (Fitriyani, 2011).

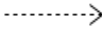
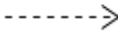
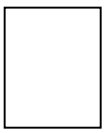
2.4 Pengertian Use Case

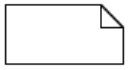
Use Case Diagram merupakan diagram yang mengammbarkan hubungan antara pengguna dengan sistem secara keseluruhan. Berikut definisi *Use Case* menurut para ahli :

1. Suatu *use case* diagram menampilkan sekumpulan *use case* dan aktor (pelaku) dan hubungan diantara *use case* dan aktor tersebut. *Use case* diagram digunakan untuk penggambaran *use case* statik dari suatu sistem. *Use case* diagram penting dalam mengatur dan memodelkan kelakuan dari suatu sistem (Booch, 2005) dalam (Fitriyani, 2011).
2. *Use Case* menurut (Fowler, 2005) dalam (Fitriyani, 2011) adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use Case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. *Use Case* Diagram menampilkan aktor mana yang menggunakan *use case* mana, *uses case* mana yang memasukkan *use case* lain dan hubungan antara aktor dan *use case.*

Sehingga dapat ditarik kesimpulan bahwa *Use case* menjelaskan apa yang dilakukan sistem (atau subsistem) tetapi tidak menspesifikasikan cara kerjanya. *Flow of event* digunakan untuk menspesifikasikan kelakuan dari *use case*. *Flow of event* menjelaskan *use case* dalam bentuk tulisan dengan sejelas-jelasnya, diantaranya bagaimana, kapan *use case* dimulai dan berakhir, ketika *use case* berinteraksi dengan aktor, obyek apa yang digunakan, alur dasar dan alur alternatif.

Tabel 2.1 Simbol *Use Case*

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

NO	GAMBAR	NAMA	KETERANGAN
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2.5 Pengertian *Activity*

Activity diagram digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity diagram* dapat juga digunakan untuk memodelkan *action* yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari *action* tersebut.

Berikut definisi activity diagram menurut para ahli :

1. *Activity diagram* menurut (Fowler, 2005) dalam (Fitriyani, 2011) adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja.
2. Menurut (Satzinger *et al*,2010) dalam (Fitriyani, 2011) *Activity diagram* merupakan sebuah tipe dari diagram *workflow* yang menggambarkan

tentang aktivitas dari pengguna ketika melakukan setiap kegiatan dan aliran sekuensial.

Dapat disimpulkan bahwa *Activity diagram* merupakan teknik yang menggambarkan aliran kerja atau proses.

Tabel 2.2 Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

2.6 Pengertian *Database*

Menurut Connolly dan Begg (2010: 65), *database* adalah sekumpulan data tersebar yang berhubungan secara logis, dan penjelasan dari data ini dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi.

Menurut Inmon (2005: 493), *database* adalah sekumpulan data yang saling berhubungan yang disimpan (biasanya dengan redudansi yang terkontrol dan

terbatas) berdasarkan skema. Sebuah *database* dapat melayani *single* atau *multiple applications*.

Menurut Gottschalk dan Saether dalam jurnal (2010: 41), *database* adalah sekumpulan data yang terorganisir untuk mendukung banyak aplikasi secara efisien dengan memusatkan data dan mengontrol data *redundant.*

Menurut Stephens dan Plew (2000), *database* adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basisdata, pengguna dapat menyimpan data secara terorganisasi. Setelah data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data disimpan dalam basis data menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan ke dalam basisdata, dimodifikasi, dan dihapus.

Menurut Silberschatz, *et al*. (2002), mendefinisikan basisdata sebagai kumpulan data berisi informasi yang sesuai untuk sebuah perusahaan.

Menurut Ramakrishnan dan Gehrke (2003), menyatakan basisdata sebagai kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi atau lebih yang berhubungan.

Menurut McLeod, *et al.* (2001), Basis data adalah kumpulan seluruh sumber daya berbasis komputer milik organisasi.

Menurut Hariyanto (2004), adalah kumpulan data (elementer) yang secara logik berkaitan dalam merepresentasikan fenomena/fakta secara terstruktur dalam domain tertentu untuk mendukung aplikasi pada sistem tertentu. Basisdata adalah

kumpulan data yang saling berhubungan yang merefleksikan fakta-fakta yang terdapat di organisasi.

2.7 Pengertian *Framework Codeigniter*

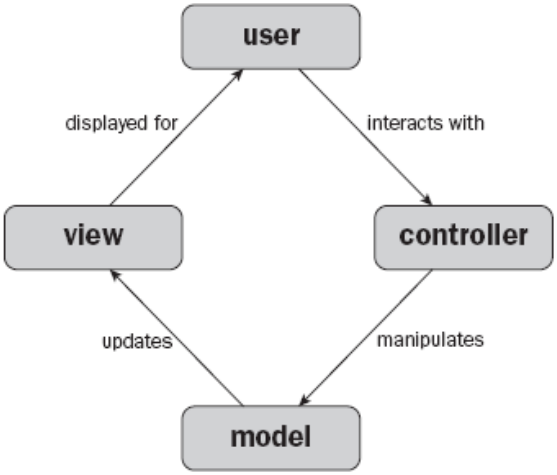
Framework secara sederhana dapat diartikan kumpulan dari fungsi-fungsi/prosedur-prosedur dan *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programer, tanpa harus membuat fungsi atau *class* dari awal.

Ada beberapa alasan mengapa menggunakan *Framework*:

- Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
- Relatif memudahkan dalam proses *maintenance* karena sudah ada pola tertentu dalam sebuah framework (dengan syarat programmer mengikuti pola standar yang ada)
- Umumnya *framework* menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya validasi, ORM, *pagination, multiple database, scaffolding*, pengaturan *session, error handling*, dll
- Lebih bebas dalam pengembangan jika dibandingkan CMS

Codeigniter merupakan aplikasi sumber terbuka yang berupa *framework* PHP dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP. *CodeIgniter* memudahkan developer untuk membuat aplikasi web dengan cepat mudah dibandingkan dengan

membuatnya dari awal. *CodeIgniter* dirilis pertama kali pada 28 Februari 2006. Versi stabil terakhir adalah versi 3.0.4



Gambar 2 1 MVC pada halaman web

***Design Pattern*: MVC (*Model, View, Controller*)**

Model View Controller merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman *Small Talk*, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC *pattern* dalam suatu aplikasi yaitu :

1. *View* merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi web bagian ini biasanya berupa *file template* HTML, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada *use*r. Bagian ini tidak memiliki akses langsung terhadap bagian model.

2. Model biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert, update, delete, search*), menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.
3. *Controller* merupakan bagian yang mengatur hubungan antara bagian model dan bagian *view*, *controller* berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

Dengan menggunakan prinsip MVC suatu aplikasi dapat dikembangkan sesuai dengan kemampuan *developer*-nya, yaitu *programmer* yang menangani bagian model dan *controller*, sedangkan *designer* yang menangani bagian *view*, sehingga penggunaan arsitektur MVC dapat meningkatkan *maintanability* dan organisasi kode. Walaupun demikian dibutuhkan komunikasi yang baik antara *programmer* dan *designer* dalam menangani variabel-variabel yang akan ditampilkan.

Ada beberapa kelebihan *CodeIgniter* (CI) dibandingkan dengan *Framework* PHP lain,

- Performa sangat cepat : salah satu alasan tidak menggunakan *framework* adalah karena eksekusinya yang lebih lambat daripada PHP *from the scracth*, tapi *Codeigniter* sangat cepat bahkan mungkin bisa dibilang *codeigniter* merupakan *framework* yang paling cepat dibanding *framework* yang lain.

- Konfigurasi yang sangat minim *(nearly zero configuration)* : tentu saja untuk menyesuaikan dengan *database* dan keleluasaan *routing* tetap diizinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi seperti *database.php* atau *autoload.php*, namun untuk menggunakan *codeigniter* dengan *setting standard*, anda hanya perlu mengubah sedikit saja *file* pada folder *config*.
- Banyak komunitas: dengan banyaknya komunitas CI ini, memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanya atau teknologi terbaru.
- Dokumentasi yang sangat lengkap : Setiap paket instalasi *codeigniter* sudah disertai *user guide* yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.

2.11 Pengertian PHP

PHP (*Hypertext Preprocessor*) adalah bahasa yang dapat ditanamkan ke dalam HTML. PHP banyak dipakai untuk pemrograman situs web dinamis. PHP dikatakan sebagai sebuah *server-side embedded script language* artinya sintaks-sintaks dan perintah yang user berikan akan sepenuhnya dijalankan oleh *server* tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di *server*. Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung PHP dapat ditemukan dimana-mana dari mulai Apache, IIS, Lighttpd, hingga *Xitami* dengan konfigurasi yang relative mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.