



BAB II
LANDASAN TEORI

BAB II

LANDASAN TEORI

2.1 Landasan Teori

2.1.1 Sistem

Sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya (Susanto, 2004). Sistem merupakan suatu rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan (Romney, 2004).

Berdasarkan pendapat para ahli di atas, dapat disimpulkan bahwa sistem adalah sekumpulan unsur atau elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan. Sistem didesain untuk memperbaiki atau meningkatkan pemrosesan informasi[1].

Sistem terdiri dari tiga unsur yaitu *input* (masukkan), proses dan *output* (pengeluaran). *Input* merupakan komponen penggerak atau pemberi tenaga di mana sistem itu dioperasikan, sedangkan *output* adalah hasil operasi. Dalam pengertian sederhana *output* berarti yang menjadi sasaran atau target pengoperasian suatu sistem sedangkan proses merupakan aktivitas yang dapat mentransformasikan *input* menjadi *output*[2].

Menurut La Midjan dan Susanto (2004) sifat dari sistem adalah sebagai berikut yaitu:

- 1) Tujuan Sistem, merupakan target atau sasaran akhir yang ingin dicapai oleh suatu sistem,
- 2) Batas Sistem, merupakan garis abstraksi yang memisahkan antara sistem dan lingkungannya,
- 3) Subsistem, merupakan komponen atau bagian dari suatu sistem, subsistem ini bisa fisik ataupun abstrak,
- 4) Hubungan dan Hirarki Sistem, merupakan hubungan yang terjadi antar subsistem dengan subsistem lainnya yang setingkat atau antara subsistem dengan sistem yang lebih besar,
- 5) *Input-Proses-Output*, yaitu sebagai masukan, diolah untuk menghasilkan berbagai keluaran, dan
- 6) Lingkungan Sistem, merupakan faktor-faktor di luar sistem yang mempengaruhi sistem.

2.1.2 Informasi

Informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi (Kusrini dan Koniyo, 2007). Informasi merupakan hasil dari pengolahan data yang memberikan arti dan manfaat (Susanto, 2004). Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam mengambil keputusan saat ini atau mendatang[1].

Dari pengertian diatas dapat disimpulkan bahwa data harus diolah terlebih dahulu agar dapat menjadi informasi yang berguna untuk pemakai informasi. Yang bersumber dari suatu pengolahan data harus merupakan suatu informasi yang memenuhi kriteria tepat waktu, relevan dan handal[2]. Menurut Susanto (2004) ada tiga hal penting yang harus diperhatikan dari uraian tentang informasi, yaitu:

- 1) Informasi merupakan hasil pengolahan data,
- 2) Memberikan makna atau arti, dan
- 3) Berguna atau bermanfaat.

Sedangkan menurut Susanto (2008) menyatakan bahwa suatu informasi yang berkualitas harus memiliki ciri-ciri sebagai berikut:

- 1) Akurat, artinya informasi harus mencerminkan keadaan yang sebenarnya,

- 2) Tepat Waktu, artinya informasi itu harus tersedia atau ada pada saat informasi tersebut diperlukan, tidak besok atau tidak beberapa jam lagi,
- 3) Relevan, artinya informasi yang diberikan harus sesuai dengan yang dibutuhkan oleh individu yang ada di berbagai tingkatan dan bagian dalam organisasi, dan
- 4) Lengkap, maksudnya informasi harus diberikan secara lengkap.

2.1.3 Database

Database adalah struktur penyimpanan data. Untuk menambah, mengakses dan memproses data yang disimpan dalam sebuah *database* komputer, diperlukan *system* manajemen seperti *MySQL Server*, (Kustiyaningsih, 2011).

Prinsip utama basis data adalah pengaturan data dengan tujuan utama fleksibilitas dan kecepatan dalam pengambilan data kembali. Adapun tujuan basis data diantaranya sebagai efisiensi yang meliputi *speed*, *space* & *accuracy*, menangani data dalam jumlah besar, kebersamaan pemakaian, dan meniadakan duplikasi.

2.1.4 PHP

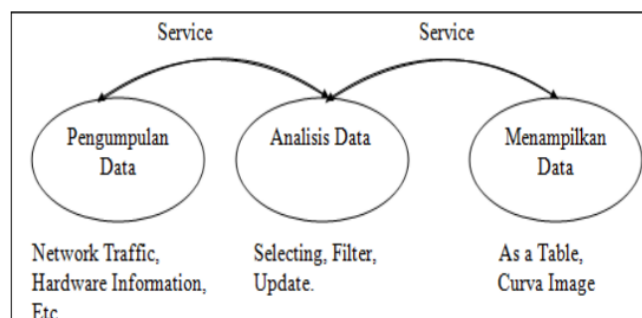
PHP merupakan bahasa pemrograman untuk membuat *website* yang bersifat *server-side scripting*. PHP bersifat dinamis. PHP dapat dijalankan pada berbagai macam sistem operasi seperti *Windows*, *Linux*, dan *Mac Os*. Selain *Apache*, PHP juga mendukung beberapa *web server* lain, seperti *Microsoft ISS*, *Caudium*, dan *PWS*. PHP dapat memanfaatkan *database* untuk menghasilkan halaman *web* yang dinamis. Sistem manajemen *database* yang sering digunakan bersama PHP adalah *MySQL*. Namun, PHP juga mendukung sistem manajemen *Database Oracle*, *Microsoft Access*, *Interbase*, *d-Base*, dan *PostgreSQL*[3].

2.1.5 Sistem Monitoring

Sistem *monitoring* adalah layanan yang melakukan proses pengumpulan data dan melakukan analisis terhadap data-data tersebut dengan tujuan untuk memaksimalkan seluruh sumber daya yang dimiliki. Tujuan sistem *monitoring* untuk mengumpulkan informasi dan data yang berguna dari suatu jaringan sehingga jaringan dapat diatur dan dikontrol[4].

Berdasarkan penjelasan berikut dapat ditarik kesimpulan bahwa sistem *monitoring* merupakan suatu proses untuk mengumpulkan data yang *real time* dari berbagai sumber. Secara garis besar tahapan dalam sebuah sistem *monitoring* terbagi ke dalam tiga proses besar seperti yang terlihat pada Gambar 2.1 dibawah, yaitu:

- 1) Proses di dalam pengumpulan data *monitoring*.
- 2) Proses di dalam analisis data *monitoring*.
- 3) Proses di dalam menampilkan data hasil *monitoring*.



Gambar 2. 1 Tahapan Sistem *Monitoring*

2.1.6 Program Kerja

Program merupakan sekumpulan rencana kegiatan yang disusun dengan tujuan tercapainya sasaran organisasi secara harmonis dan integratif. Dalam sebuah organisasi dibutuhkan program kerja untuk menjadi pedoman bagi setiap kegiatan yang akan dilaksanakan. Program kerja ini menjadi pedoman bagi setiap kegiatan yang harus dijalankan organisasi dalam kurun waktu tertentu. Melalui program kerja ini maka segala bentuk rencana kegiatan akan lebih terorganisir. Program kerja akan membantu organisasi mahasiswa untuk mewujudkan visi dan misi organisasi[5]. Dalam setiap program kerja harus memuat:

- 1) Sasaran atau tujuan kegiatan yang akan dicapai.
- 2) Rangkaian kegiatan yang akan dilaksanakan dalam mencapai tujuan tersebut.
- 3) Memuat aturan-aturan yang harus dijalankan saat melaksanakan kegiatan tersebut.
- 4) Rencana anggaran yang dibutuhkan.
- 5) Strategi pelaksanaan.

2.1.7 Badan Eksekutif Mahasiswa (BEM)

Organisasi mahasiswa yang memiliki kedudukan resmi di lingkungan perguruan tinggi dan mendapat pendanaan kegiatan kemahasiswaan dari pengelola perguruan tinggi dan atau dari Kementerian atau Lembaga. Tugas pokok BEM adalah mewakili mahasiswa, mengkoordinasikan kegiatan organisasi kemahasiswaan dalam bidang ekstrakurikuler di tingkat fakultas dan memberikan pendapat, usul dan saran kepada pimpinan fakultas terutama berkaitan dengan peran, fungsi dan pencapaian tujuan pendidikan nasional[6]. BEM memiliki beberapa fungsi yaitu:

- 1) Sebagai lembaga tinggi mahasiswa yang akan mengakomodir seluruh kepentingan mahasiswa di kampus.
- 2) Sebagai agen perubahan dalam kehidupan berbangsa dan bernegara.
- 3) Sebagai *social control* dalam melihat setiap kebijakan-kebijakan yang ada baik dalam lembaga kampus maupun dalam pemerintahan Republik Indonesia (regional dan nasional).
- 4) Membangun sinergitas dan solidaritas dengan seluruh ormawa di kampus.
- 5) Sebagai fasilitator dalam menjaring aspirasi setiap mahasiswa di kampus.

2.1.8 Rekayasa Perangkat Lunak

Software atau perangkat lunak pada saat ini menjadi salah satu komponen penting dalam kehidupan manusia. Perangkat lunak digunakan untuk mempermudah pekerjaan-pekerjaan yang memerlukan keakuratan, ketepatan dan efisiensi waktu yang tinggi. Perekayasa perangkat lunak yang baik menggunakan pengukuran untuk menilai kualitas model analisis, desain, kode sumber, dan *test case* yang dibuat ketika perangkat lunak direkayasa. Untuk mencapai penilaian kualitas *real-time*, perekayasa harus menggunakan pengukuran untuk mengevaluasi kualitas dalam cara-cara yang obyektif. Pengertian rekayasa perangkat lunak adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, desain, pengkodean, pengujian sampai pemeliharaan sistem setelah dikembangkan (Yasin, 2012).

Berdasarkan penjelasan diatas maka bisa disimpulkan bahwa perangkat lunak adalah kumpulan beberapa perintah yang dieksekusi oleh mesin komputer dalam menjalankan pekerjaannya. Perangkat lunak ini merupakan catatan bagi mesin komputer untuk menyimpan perintah, maupun dokumen serta arsip lainnya. Sedangkan rekayasa perangkat lunak merupakan ilmu yang penting untuk diperdalam karena teknologi ini memberikan stabilitas, kontrol dan

organisasi aktifitas yang jika tidak terkontrol dapat menjadi sangat kacau. Tujuan dilakukan rekayasa perangkat lunak antara lain:

- 1) Untuk membangun *software* yang benar dan benar sebuah *software*.
- 2) Untuk membangun *software* yang tepat.
- 3) Dikelola dengan baik untuk pemeliharaan kebenarannya.
- 4) Memperoleh biaya produksi perangkat lunak yang rendah.
- 5) Menghasilkan perangkat lunak yang kinerjanya tinggi, andal dan tepat waktu.
- 6) Menghasilkan perangkat lunak yang dapat bekerja pada berbagai jenis *platform*.
- 7) Menghasilkan perangkat lunak yang biaya perawatannya rendah.

Rekayasa perangkat lunak memiliki beberapa pokok Bahasa Penting, yaitu:

- 1) *Domain Engineering*
Kemampuan untuk memahami suatu permasalahan yang muncul dan dijadikan suatu proyek perangkat lunak.
- 2) *Requirement Engineering*
Kemampuan untuk memahami kebutuhan seorang pengguna sekaligus dalam memecahkan permasalahan.
- 3) *Software Design*
Kemampuan untuk memahami dan mengimplementasikan dalam perancangan perangkat lunak yang didalamnya terdapat aspek *HCI (Human Computer Interaction)*.
- 4) *Development*
Proses pengembangan yang melibatkan pembelajaran mengenai algoritma, bahasa pemrograman yang di terapkan, dan teknik yang didalamnya berkaitan dengan basis data dan sistem informasi.
- 5) *Maintenance*
Pemeliharaan dan pengembangan suatu sistem yang berguna untuk banyak orang.

2.1.9 Flowchart

Flowchart atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem. seorang analis sistem menggunakan *flowchart* sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun kepada *programmer*. Dengan begitu, *flowchart* dapat membantu untuk memberikan solusi terhadap masalah yang bisa saja terjadi dalam membangun sistem. Pada dasarnya, *flowchart* digambarkan dengan menggunakan simbol-simbol. Setiap simbol mewakili suatu proses tertentu. Sedangkan untuk menghubungkan satu proses ke proses selanjutnya digambarkan dengan menggunakan garis penghubung[7]. Berikut penjelasan mengenai beberapa fungsi *flowchart*:

- 1) Merancang Proyek Baru
Ketika Anda akan merancang suatu proyek, maka hal selanjutnya yang dapat Anda lakukan adalah memetakan proyek tersebut ke bentuk *flowchart*. Itu dapat membantu Anda untuk merancang serangkaian langkah-langkah yang melibatkan keputusan bersama.
- 2) Mengelola Alur Kerja
Untuk mengelola alur kerja, *flowchart* adalah cara yang paling penting dilakukan. Sebab, *flowchart* berperan dalam penentuan integritas dari proses tersebut, yaitu dapat menciptakan hasil yang berkualitas berdasarkan prosedur.
- 3) Memodelkan Proses Bisnis
Proses bisnis yang dimaksud bukan hanya berkaitan dengan keuntungan, melainkan serangkaian tugas baik itu yang sederhana sampai yang rumit juga termasuk kedalam proses bisnis. Tujuan permodelan *flowchart* dapat dilakukan untuk memberikan hasil yang

konsisten dan dapat juga diprediksi.

4) Mendokumentasikan Setiap Proses

Dalam menyelesaikan suatu proyek perlu adanya dokumentasi proses. Dengan begitu, *flowchart* menjadi media yang bagus untuk memenuhi tujuan tersebut. Jika dibandingkan dengan membuat dokumentasi setiap proses melalui narasi dengan memetakannya ke dalam bentuk *flowchart*, maka menggunakan *flowchart* merupakan hal yang lebih efisien dibanding dengan narasi.

5) Merepresentasikan Algoritma

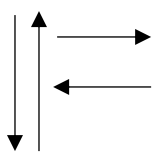





Biasanya, sebelum menuangkan proyek tersebut kedalam bentuk program, para perancang sistem terlebih dahulu menentukan algoritma untuk menyelesaikan proyek tersebut menggunakan *SDL*. *SDL (Specification and Description Language)* merupakan suatu spesifikasi bahasa yang digunakan untuk mendeskripsikan perilaku dari sistem tersebut. *Flowchart* dapat memenuhi kebutuhan tersebut. Sebab, *flowchart* menawarkan berbagai simbol yang unik yang dapat digunakan untuk memetakan sistem yang akan dirancang. Selain itu, *flowchart* juga menawarkan sumber daya yang dapat digunakan untuk menyelesaikan suatu masalah.


6) Mengedit Proses

Secara umum, *flowchart* dapat digunakan untuk mendeteksi kerusakan yang terjadi dalam setiap proses. *Flowchart* dapat membantu Anda menyelesaikan permasalahan dengan cara membagi setiap langkah dari proses itu kedalam segmen-segmen yang lebih kecil, kemudian memeriksa bagian mana yang tidak berfungsi atau perlu diadakan perbaikan.

Flowchart memiliki berbagai macam simbol, yaitu[8]:

Tabel 2. 1 Simbol-simbol *Flowchart*

No.	Simbol	Nama	Keterangan
1.		<i>Flow Direction Symbol/Connecting Line</i>	Befungsi untuk menghubungkan simbol yang satu dengan yang lainnya, menyatakan arus suatu proses.
2.		<i>Terminator</i>	Yaitu simbol untuk permulaan (<i>start</i>) atau akhir (<i>stop</i>) dari suatu kegiatan
3.		<i>Process</i>	Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.
4.		<i>Manual Operation</i>	Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer.
5.		<i>Decision</i>	Simbol pemilihan proses berdasarkan kondisi yang ada.
6.		<i>Input Output</i>	Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.

No.	Simbol	Nama	Keterangan
7.		<i>Document</i>	Simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak ke kertas.




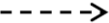

2.1.10 UML (*Unified Modeling Language*)

UML adalah bahasa standar yang digunakan dalam pengembangan perangkat lunak untuk merancang, mendokumentasikan, dan memodelkan sistem perangkat lunak dengan menggunakan notasi grafis. UML membantu dalam komunikasi antara pengembang dan pemangku kepentingan serta memberikan kerangka kerja untuk merancang sistem perangkat lunak yang kompleks. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun[9]. Berikut ini beberapa diagram UML antara lain:

1. *Use Case Diagram*

Use case merupakan deskripsi fungsi dari sebuah sistem dari perspektif atau sudut pandang para pengguna sistem. *Use case* mendefinisikan apa yang akan diproses oleh sistem dan komponen – komponennya. *Use case* bekerja dengan menggunakan *scenario* yang merupakan deskripsi dari urutan atau langkah – langkah yang menjelaskan apa yang dilakukan oleh *user* terhadap sistem maupun sebaliknya[10]. Berikut simbol-simbol yang terdapat pada *use case diagram* dapat dilihat pada Tabel 2.2 berikut[11].






Tabel 2. 2 Simbol-simbol *Use Case Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2.		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>usecase</i> .
3.		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
5.		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

2. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah sekumpulan cara atau peralatan untuk mendeskripsikan data-data atau objek-objek yang dibuat berdasarkan dan berasal dari dunia nyata yang disebut entitas (*entity*) serta hubungan (*relationship*) antar entitas-entitas tersebut dengan menggunakan beberapa notasi[12]. Berikut simbol-simbol yang terdapat pada *entity relationship diagram* yang dapat dilihat pada Tabel 2.3 dibawah:

Tabel 2. 3 Simbol-simbol *Entity Relationship Diagram (ERD)*

No.	Simbol	Nama	Keterangan
1.		Entitas	Merupakan kumpulan dari objek yang dapat diidentifikasi secara unik.
2.		Relasi	Merupakan hubungan yang terjadi antara satu atau lebih entitas.
3.		Atribut	Merupakan karakteristik dari relasi yang merupakan penjelasan detail tentang entitas.
4.		Garis	Merupakan hubungan antara entitas dengan atributnya dan himpunan entitas dengan himpunan relasi.
5.		<i>Input-Output</i>	Merupakan proses <i>input</i> atau <i>output</i> data, parameter, informasi.

-HALAMAN INI SENGAJA DIKOSONGKAN-