BAB II
LANDASAN TEORI

2.1 Sistem

Sistem, yaitu sekumpulan objek-objek yang saling berelasi dan berinteraksi serta hubungan antar objek yang dilihat sebagai satu kesatuan yang dirancang untuk mencapai satu tujuan (Hanif, 2007). Sistem mempunyai karakteristik sebagai berikut (Agus, 2009) :

1. Mempunyai komponen sistem (*components system*)

   Suatu sistem tidak berada dalam lingkungan yang kosong, tetapi sebuah sistem berada dan berfungsi di dalam lingkungan yang berisi sistem lainnya. Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Apabila suatu sistem merupakan salah satu dari komponen sistem lain yang lebih besar, maka akan disebut dengan subsistem, sedangkan sistem yang lebih besar tersebut adalah lingkungannya.
2. Mempunyai batasan sistem (*boundary*)

   Batas sistem merupakan pembatas atau pemisah antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.
3. Mempunyai lingkungan (*environment*)

   Lingkungan luar adalah apa pun di luar batas dari sistem yang dapat mempengaruhi operasi sistem, baik pengaruh yang menguntungkan ataupun yang merugikan. Pengaruh yang menguntungkan ini tentunya harus dijaga sehingga akan mendukung kelangsungan operasi sebuah sistem. Sedangkan lingkungan yang merugikan harus ditahan dan dikendalikan agar tidak mengganggu kelangsungan sebuah sistem.
4. Mempunyai penghubung (*interface*) antar komponen

   Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Penghubung inilah yang akan menjadi media yang digunakan data dari masukan (*input*) hingga keluaran (*output*). Dengan

adanya penghubung, suatu subsistem dapat berinteraksi dan berintegrasi dengan subsistem yang lain membentuk satu kesatuan.

5. Mempunyai masukan (*input*)

   Masukan atau *input* merupakan energi yang dimasukan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*), yaitu bahan yang dimasukkan agar sistem tersebut dapat beroperasi dan masukan sinyal (*signal input*), yaitu masukan yang diproses untuk mendapatkan keluaran.

6. Mempunyai pengolahan (*processing*)

   Pengolahan (*process*) merupakan bagian yang melakukan perubahan dari masukan untuk menjadi keluaran yang diinginkan.

7. Mempunyai sasaran (*objective*) dan tujuan

   Suatu sistem pasti memiliki sasaran (*objective*) atau tujuan *(goal*). Apabila sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Tujuan inilah yang mengarahkan suatu sistem. Tanpa adanya tujuan, sistem menajdi tidak terarah dan terkendali.

8. Mempunyai keluaran (*output*)

   Keluaran (*output*) merupakan hasil dari pemrosesan. Keluaran dapat berupa informasi sebagai masukan pada sistem lain atau hanya sebagai sisa pembuangan.

9. Mempunyai umpan balik (*feed back*)

   Umpan balik diperlukan oleh bagian kendali (*control*) sistem untuk mengecek terjadinya penyimpangan proses dalam sistem dan mengembalikannya ke dalam kondisi normal.

2.2 Metodologi Pengembangan Sistem

Metodologi pengembangan sistem yang digunakan dalam penulisan ini, yaitu metodologi *System Development Life Cycle* (SDLC) yang menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, dan pengujian.

1. Analisis

   Proses pengumpulan kebutuhan yang dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.
2. Desain

   Desain perangkat lunak adalah proses multi langkah yang fokus pada desin pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.
3. Pengodean

   Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.
4. Pengujian

   Pengujian hanya fokus pada perangkat lunak secara dari segi lojik dan fungsional dan memastikan bahwa semua bagian sudah di uji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan (Rossa A.S. dan M. Shalahuddin, 2013).

2.3 Alat Pengembangan Sistem

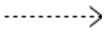
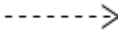
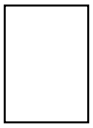
Berikut ini merupakan alat pengembangan sistem yang akan digunakan dalam penelitian.

2.3.1 Diagram *Use-Case (Use Case Diagram)*

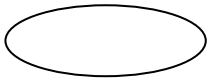
Bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk

mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

Tabel 2.1 Simbol *Use Case Diagram*

GAMBAR	NAMA	KETERANGAN
	*Actor*	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case*
	*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*)
	*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*)
	*Include*	Menspesifikasikan bahwa *use case* sumber secara eksplisit
	*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
	Association	Apa yang menghubungkan anatara objek satu dengan objek lainnya.
	*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

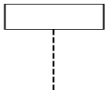
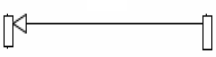
Tabel 2.1 Simbol *Use Case Diagram* (Lanjutan)

GAMBAR	NAMA	KETERANGAN
	*Use case*	Deskripsi dari urutan aksi-aksi yang menampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

2.3.2 Diagram Interaksi dan Urutan (*Sequence*)

Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

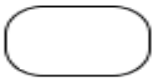
Tabel 2.2 Simbol *Sequence Diagram*

GAMBAR	NAMA	KETERANGAN
	*Life Line*	Objek *entity*, antarmuka yang saling berinteraksi.
	*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
	*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

2.3.3 Diagram Aktivitas (*Activity Diagram*)

Bersifat dinamis. Diagram aktifitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek. Berikut ini merupakan simbol *activity diagram.* Simbol-simbol yang digunakan dalam membuat sebuah *activity diagram* yaitu sebagai berikut.

Tabel 2.3 Simbol *Activity Diagram*

GAMBAR	NAMA	KETERANGAN
	*Actifity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
	*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
	*Initial Node*	Bagaimana objek dibentuk atau diawali.
	*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
	*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

2.3.4 Kamus data

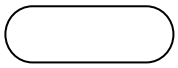
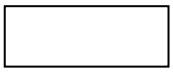
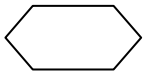
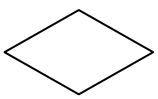
Menurut Rosa dan Shalahudin (2013), kamus data merupakan kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan). Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur. Kamus data biasanya berisi :

1. Nama-nama dari data
2. Digunakan untuk proses-proses yang terkait data
3. Deskripsi merupakan deskripsi data
4. Informasi tambahan, seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data.

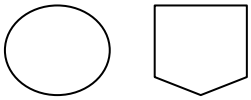
2.3.5 *System flow*

Menurut Sutabri (2012), *system flow* adalah bagan alir sistem yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. *System flow* menunjukkan sistematika dari prosedur yang ada di dalam sistem dan menunjukkan apa yang dilakukan sistem.

Tabel 2.4 Simbol *System Flow*

SIMBOL	KETERANGAN
Terminator 	Digunakan untuk memberikan awal dan akhir suatu proses.
Proses 	Menunjukkan proses dari operasi program komputer.
Preparation 	Proses inisialisasi/pemberian harga awal.
Input/output data 	Proses *input/output* data, parameter, informasi.
Garis alir	Digunakan untuk menunjukkan arus dari proses.
Decision 	Digunakan untuk suatu penyeleksian kondisi di dalam program.

Tabel 2.4 Simbol *System Flow* (Lanjutan)

SIMBOL	KETERANGAN
Proses terdefinisi 	Simbol yang digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan ditempat lain.
Penghubung 	Simbol yang digunakan untuk menunjukkan sambungan dari bagan alir yang terputus dihalaman yang sama maupun dihalaman yang lain.

2.4 Perangkat Lunak Pendukung

Sub bab ini akan menjelaskan tentang perangkat lunak pendukung yang digunakan dalam penelitian.

1. Netbeans IDE 8

Pada pembuatan program aplikasi ini menggunakan netbeans IDE 8. Netbeans adalah salah satu aplikasi IDE yang digunakan oleh *developer software* komputer untuk menulis, meng-*compile*, mencari kesalahan, dan untuk menyebarkan program (Eko K.K., 2012).

2. MySQL

MySQL merupakan *software* RDBMS yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah yang sangat besar, dapat diakses oleh banyak *user (multi-user)*, dan dapat melakukan suatu proses secara sinkron atau berbarengan *(multi-threaded)* (Budi Raharjo, 2011). Pengaksesan data pada mysql dengan SQL (*Structured Query Language*) terdiri dari 4 hal, yaitu :

a. Memasukkan data (*insert*)

   Contoh penggunaan : insert into tabel_mahasiswa (nim, nama, tgl_lahir) values (‘13501058’,’Rosa’,’1986-01-01’);.

b. Mengubah data (*update*)

   Contoh penggunaan : update tabel_mahasiswa set tanggal_lahir = ’1990-03-04’ where nim = ‘13501058’;.

c. Menghapus data (*delete*)

   Contoh penggunaan : delete from tabel_mahasiswa where nim = ‘13501058’;

d. Menampilkan data (*select*)

   Contoh penggunaan : select nim, nama from tabel_mahasiswa where nim = ‘13501058’ (Rossa A.S. dan M. Shalahuddin, 2013).

3. Ireport

Ireport adalah *report designer visual* yang dibangun pada jasperreport. Ireport bersifat intuitif dan mudah digunakan dalam membangun laporan *visual* atau desainer untuk jasperreport. Sebagai alternatif, terdapat *tools* ireport (dengan *library* jasperreport) yang dapat membantu dalam pembuatan laporan. *Library* jasperreport sendiri merupakan java *library* (JAR) yang bersifat *open* dan dirancang untuk menambahkan kemampuan pelaporan *(reporting capabilities)* pada aplikasi java.