

## BAB II

### LANDASAN TEORI

#### 2.1 Sistem

Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan (Kadir, 2003). Sebagai gambaran, jika dalam sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka elemen itu dapat dipastikan bukanlah bagian dari sistem.

##### 2.1.1 Klasifikasi Sistem

Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi yang ada di dalam sistem tersebut (Sutabri, 2012). Oleh karena itu, sistem dapat diklasifikasikan dari beberapa sudut pandang di antaranya:

##### 1. Sistem abstrak dan sistem fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia, yaitu sistem yang berupa pemikiran hubungan antara manusia dengan Tuhan, sedangkan sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem komputer, sistem produksi, sistem penjualan, sistem administrasi personalia dan lain sebagainya.

##### 2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadinya siang malam, pergantian musim. Sedangkan sistem buatan manusia merupakan sistem yang melibatkan interaksi manusia dengan mesin yang disebut *human machine system*. Sistem informasi berbasis komputer merupakan contoh *human machine system* karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

3. Sistem determinasi dan sistem probobalistik  
Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem *deterministic*. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan, sedangkan sistem yang bersifat probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probablistik.
4. Sistem terbuka dan sistem tertutup  
Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa campur tangan pihak luar. Sedangkan sistem terbuka adalah sistem yang berhubungan oleh lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya

## 2.2 Informasi

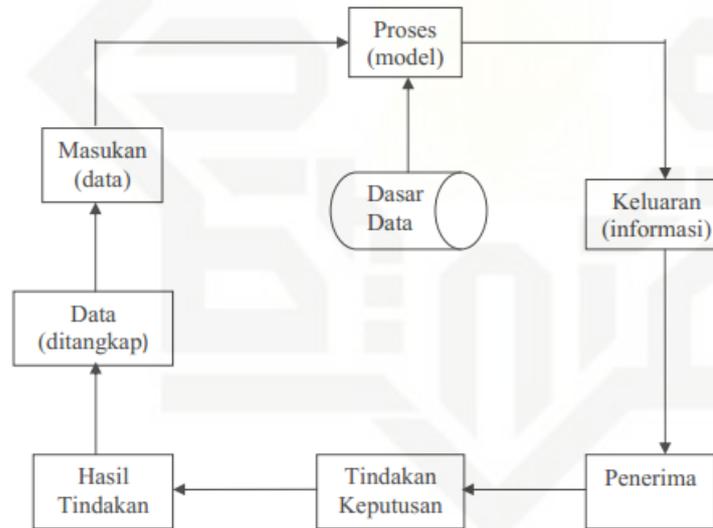
Menurut McFadden, dkk (1999) informasi adalah data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut (Kadir, 2003). Jadi, hal yang terpenting untuk membedakan informasi dengan data, informasi itu mempunyai kandungan makna sedangkan data tidak mempunyai makna. Istilah kualitas informasi terkadang juga dipakai untuk menyatakan informasi yang baik. Dari sekian karakteristik yang telah dibahas, kualitas informasi sering kali diukur berdasarkan relevansi, ketepatan waktu dan keakurasian. Kualitas informasi (*quality of information*) sangat dipengaruhi atau ditentukan oleh tiga hal sebagai berikut:

1. Relevan (*relevancy*)

Seberapa jauh tingkat relevansi informasi tersebut terhadap kenyataan kejadian masa lalu, kejadian hari ini dan kejadian yang akan datang. Informasi yang berkualitas akan mampu menunjukkan benang merah relevansi kejadian masa lalu, hari ini dan masa depan sebagai sebuah bentuk aktivitas yang kongkrit dan mampu dilaksanakan dan dibuktikan oleh siapa saja.

2. Akurat (*accuracy*)  
 Suatu informasi dikatakan berkualitas jika seluruh kebutuhan informasi tersebut tersampaikan (*completensess*), seluruh pesan telah benar atau sesuai (*correctness*), serta pesan yang disampaikan sudah lengkap atau hanya sistem yang diinginkan oleh *user* (*security*).
3. Tepat Waktu (*on time*)  
 Berbagai proses dapat diselesaikan dengan tepat waktu, laporan-laporan yang dibutuhkan dapat disampaikan tepat waktu.

Siklus informasi menggambarkan pengolahan data menjadi informasi dan pemakaian informasi untuk pengambilan keputusan, hingga akhirnya dari tindakan hasil pengambilan keputusan tersebut dihasilkan data kembali. Data tersebut akan ditangkap sebagai *input*. Untuk lebih jelasnya dapat dilihat pada Gambar 2.1 di bawah ini.



Gambar 2.1 Siklus Informasi  
 (Sumber: Kadir, 2003)

### 2.3 Konsep Dasar Sistem Informasi

Menurut Wilkinson dalam buku Abdul Kadir yang berjudul Pengenalan Sistem Informasi, Sistem Informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (*input*) menjadi keluaran (informasi), guna mencapai sasaran perusahaan (Kadir, 2003). Sedangkan

**Hak Cipta Dilindungi Undang-Undang**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

menurut Bodnar dan Hopwood sistem informasi adalah kumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna (Kadir, 2003).

Informasi merupakan proses lebih lanjut dari data yang sudah memiliki nilai tambah. Informasi dapat dikelompokkan menjadi 3 bagian (Sutabri, 2012), yaitu:

1. Informasi Strategis

Informasi ini digunakan untuk mengambil keputusan jangka panjang, yang mencakup informasi eksternal, rencana perluasan perusahaan, dan sebagainya.

2. Informasi Taktis

Informasi ini dibutuhkan untuk mengambil keputusan jangka menengah, seperti informasi tren penjualan yang dapat dimanfaatkan untuk menyusun rencana penjualan.

3. Informasi Teknis

Informasi ini dibutuhkan untuk keperluan operasional sehari-hari, seperti informasi persediaan stok, retur penjualan dan laporan kas harian.

## 2.4 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*), yang terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data, dan blok kendali. Sebagai suatu sistem, keenam blok tersebut saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran (Sutabri, 2012).

1. Blok masukan (*input block*)

*Input* mewakili data yang masuk ke dalam sistem informasi. Yang dimaksud dengan input disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

**Hak Cipta Dilindungi Undang-Undang**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

3. Blok keluaran (*output block*)  
Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
4. Blok teknologi (*technology block*)  
Teknologi merupakan *tool box* dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian sistem secara keseluruhan. Teknologi terdiri dari 3 (tiga) bagian utama, yaitu teknisi (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).
5. Blok basis data (*database block*)  
Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan perangkat lunak digunakan untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut DBMS (*Database Management System*).
6. Blok kendali (*control block*)  
Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan pada sistem itu sendiri, ketidak-efisienan, sabotase, dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dicegah dan bila terlanjur terjadi maka kesalahan-kesalahan dapat dengan cepat diatasi.

## 2.5 Analisis dan Perancangan Sistem

Analisa sistem dapat di definisikan sebagai pengurai dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya (Jogiyanto, 2005).

Sedangkan desain sistem dapat diartikan sebagai berikut:

1. Tahap setelah analisis dari siklus pengembangan sistem.
2. Pendefinisian dari kebutuhan-kebutuhan fungsional.
3. Persiapan untuk rancang bangun implementasi.
4. Menggambarkan bagaimana suatu sistem dibentuk
5. Yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh dan berfungsi.
6. Termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem.

Analisis sistem didefinisikan sebagaimana memahami dan menspesifikasi dengan detail apa yang harus dilakukan oleh sistem. Sementara sistem desain diartikan sebagai menjelaskan detail bagaimana bagian-bagian dari sistem informasi diimplementasikan. Dengan demikian, Analisis dan Desain Sistem Informasi (ANSI) bisa didefinikan sebagai proses organisasional kompleks di mana sistem informasi berbasis komputer diimplementasikan. Atau bisa diringkas, analisis mendefinisikan masalah dan desain memecahkan masalah (Al Fatta, 2008).

## 2.6 Matrikulasi dan Peminatan

Berikut adalah pengertian dari Matrikulasi dan Peminatan di Laboratorium Sistem Informasi UIN Suska Riau.

### 2.6.1 Matrikulasi

Matrikulasi adalah upaya akademik yang dilakukan oleh Program Studi Sistem Informasi untuk mengurangi perbedaan sudut pandang, dengan memberikan bekal akademik untuk memudahkan Mahasiswa dalam mengikuti perkuliahan, yang di asumsikan bahwa Mahasiswa baru mempunyai latar belakang yang kurang relevan dengan Program Studi Sistem Informasi. Dalam hal ini bekal akademik yang diberikan yaitu program *Office* dan *Web*.

## 2.6.2 Peminatan

Adalah kegiatan non pelajaran formal yang ada di Sistem Informasi Uin Suska Riau diluar jam perkuliahan dengan tujuan mengembangkan bakat dan kemampuan.

## 2.7 Metode Berorientasi Objek

Metode berorientasi objek atau *object oriented* merupakan paradigma baru dalam rekayasa perangkat lunak yang memandang sistem sebagai kumpulan objek-objek diskrit yang saling berinteraksi. Yang dimaksud dengan berorientasi objek adalah bahwa mengorganisasikan perangkat lunak sebagai kumpulan objek-objek diskrit yang bekerja sama antara informasi atau struktur data dan perilaku (*behavior*) yang mengaturnya (Sholiq, 2006).

*Object Oriented Programming* (OOP) adalah suatu cara baru dalam berpikir serta berlogika dalam menghadapi masalah-masalah yang akan dicoba-atasi dengan bantuan komputer. Filosofi OOP menciptakan sinergi luar biasa sepanjang siklus pengembangan perangkat lunak (perencanaan, analisis, perancangan, serta implementasi) sehingga dapat diterapkan pada perancangan sistem secara umum menyangkut perangkat lunak, perangkat keras, serta sistem informasi secara keseluruhan (Nugroho, 2002).

Objek adalah konsep abstraksi atau sesuatu yang memiliki arti bagi aplikasi yang akan dikembangkan. Pengertian objek biasanya adalah kata benda, namun objek dalam konteks OOP bukan hanya objek nyata yang bisa dilihat serta diraba dan dilihat secara kasat mata, namun juga menyangkut entitas-entitas konseptual seperti rumus persamaan kuadrat, liberalisme, marxisme, dan sebagainya.

Beberapa istilah berorientasi objek adalah (Sholiq, 2006):

1. Abstraksi (*abstraction*)
2. Pewarisan (*inheritance*)
3. Banyak bentuk (*polymorphism*)
4. Pembungkusan (*encapsulation*)
5. Pengiriman pesan (*message sending*)
6. Assosiasi (*assosiation*)
7. Agregasi (*aggregation*)

## 2.8 *Object Oriented Analysis (OOA)*

*Object Oriented Analysis (OOA)* adalah tahapan perangkat lunak dengan menentukan spesifikasi sistem (sering orang menyebutnya sebagai *SRS/System Requirement Spesifikasi*) dan mengidentifikasi kelas-kelas serta hubungannya satu terhadap yang lainnya (Nugroho, 2005). Untuk memahami spesifikasi sistem, kita perlu mengidentifikasi para pengguna atau yang sering disebut sebagai aktor-aktor. Siapa aktor-aktor yang akan menggunakan sistem dan bagaimana mereka menggunakan sistem.

Mencari objek-objek fisik pada sistem juga memungkinkan kita untuk mendapatkan informasi lebih lengkap objek-objek pada sistem yang bersangkutan. Objek-objek dapat bersifat mandiri, organisasi-organisasi, satuan informasi, gambar-gambar, atau apapun yang menyusun suatu aplikasi dalam konteks representasi dunia nyata dalam sistem yang sedang dikembangkan.

Berikut ini aktifitas utama dari OOA adalah:

1. Menganalisis masalah domain
2. Menjelaskan sistem proses
3. Mengidentifikasi objek
4. Menentukan atribut
5. Mengidentifikasi operasi
6. Komunikasi objek

## 2.9 *Object Oriented Design (OOD)*

*Object Oriented Design (OOD)* adalah merancang kelas-kelas yang teridentifikasi selama tahap analisis dan antarmuka (*user Interface*). Selama tahap ini kita mengidentifikasi dan menambah beberapa objek dan kelas yang mendukung implementasi dari spesifikasi kebutuhan (Nugroho, 2005). Adapun proses pada OOD meliputi:

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

1. Mendefinisikan konteks dan mode dari penggunaan sistem.
2. Mendesai arsitektur sistem.
3. Identifikasi objek sistem utama.
4. Mengembangkan model desain.
5. Menentukan interface objek.

### 2.10 Unified Modeling Language (UML)

Notasi *Unified Modeling Language* (UML) dibuat sebagai kolaborasi dari Grady Booch, DR.James Rumbough, Ivar Jacobson, Rebecca Wirfs-Brock, Peter Yourdon, dan lainnya. Jacobson menulis tentang pendefinisian persyaratan-persyaratan sistem yang disebut *use case*. Juga mengembangkan sebuah metode untuk perancangan sistem yang disebut *Object Oriented Software Engineering* (OOSE) yang berfokus pada analisis. Boch, Rumbough dan Jacobson biasa disebut dengan tiga sekawan (*tree amigod*). Semuanya bekerja di *Rational Software Corporation* dan berfokus pada standarisasi dan perbaikan ulang UML. Simbol UML mirip dengan Boch, notasi OMT, dan juga ada kemiripan dengan notasi lainnya (Sholih, 2006).

Penggabungan beberapa metode menjadi UML dimulai 1993. Pada akhir tahun 1995 *Unified Method* versi 0.8 diperkenalkan. *Unified Method* diperbaiki dan di ubah menjadi UML pada tahun 1996, UML 1.0 disahkan dan diberikan pada *Object Technology Group* (OTG) pada tahun 1997, dan pada tahun itu juga beberapa perusahaan pengembang utama perangkat lunak mulai mengadopsinya.

UML menyediakan beberapa diagram visual yang menunjukkan berbagai aspek dalam sistem. Ada beberapa diagram yang disediakan dalam UML antara lain:

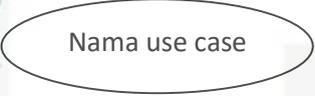
1. Digram *use case* (*use case diagram*)
2. Diagram kelas (*class diagram*)
3. Diagram sekuensial (*sequence diagram*)

### 2.11 Use Case Diagram

Diagram *use case* menyajikan interaksi antara *use case* dan aktor. Dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan

sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai atau pengguna. Simbol-simbol *use case diagram* dapat dilihat pada Tabel 2.1 di bawah ini.

Tabel 2.1 Simbol-simbol *Use Case Diagram*

Simbol	Deskripsi
	Fungsionalisasi yang di sediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya di nyatakan dengan menggunakan kata kerja di awal <i>frase name use case</i> .
	Orang, proses atau sistem yang lain yang berinteraksi dengan sistem informasi yang akan di buat diluar sistem yang akan di buat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu menggunakan orang biasanya di nyatakan menggunakan kata benda di awal <i>frase</i> nama aktor
	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang di tambahkan misal arah panah mengarah pada <i>use case</i> yang di tambahkan

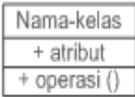
Tabel 2.1 Simbol-simbol *Use Case Diagram* (Lanjutan)

Simbol	Deskripsi
Generalisasi 	Hubungan generalisasi dan spesialisasi antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<i>Include</i> <<include>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> di tambahkan memerlukan <i>use case</i> ini menjalakan fungsinya atau syarat di jalankan <i>use case</i> ini

### 2.12 Class Diagram

Diagram kelas (*Class Diagram*) menunjukkan interaksi antara kelas dalam sistem. Kelas mengandung informasi dan tingkah laku (*behavior*) yang berkaitan dengan informasi tersebut. Sebuah kelas pada diagram kelas dibuat untuk setiap tipe objek pada diagram sekuensial atau diagram kolaborasi. Simbol-simbol diagram kelas dapat dilihat pada Tabel 2.2 di bawah ini.

Tabel 2.2 Simbol-simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
<i>interface</i>  <b>nama_interface</b>	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek
<i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga di sertai dengan <i>multiplicity</i>
<i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

Tabel 2.2 Simbol-simbol *Class Diagram* (Lanjutan)

Simbol	Deskripsi
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesifikasi (umum-khusus)
<i>dependency</i> 	Relasi antar kelas dengan makna ketergantungan antar kelas
<i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> )

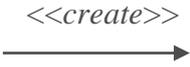
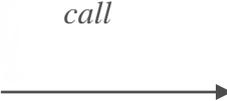
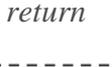
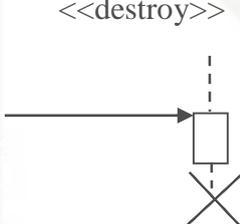
### 2.13 *Sequence Diagram*

Diagram sekuensial (*sequence diagram*) adalah diagram interaksi yang disusun berdasarkan kurun waktu. Diagram sekuensial digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Simbol-simbol *sequence diagram* dapat dilihat pada Tabel 2.3 di bawah ini.

Tabel 2.3 Simbol-simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor  Nama aktor	Orang, proses atau sistem yang lain yang berinteraksi dengan sistem informasi yang akan di buat diluar sistem yang akan di buat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu menggunakan orang biasanya di nyatakan menggunakan kata bemda di awal <i>frase</i> nama actor.
<i>lifeline</i> 	Menyatakan hidup suatu objek
Objek 	Menyatakan objek yang berinteraksi pesan

Tabel 2.3 Simbol-simbol *Sequence Diagram* (Lanjutan)

Simbol	Deskripsi
	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya
	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang di buat
	Menyatakan suatu objek memanggil operasi atau metode yang ada pada objek lain atau dirinya sendiri
	Menyatakan suatu objek mengirim data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
	Menyatakan suatu objek yang telah menjalankan suatu operasi atau metode yang menghasilkan suatu kembalianke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
	Menyatakan suatu objek mengahiri hidup objek yang lain, arah panah mengarah pada objek yang diakiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

## 2.14 PHP

PHP *Hypertext Preprocessor* adalah bahasa pemrograman yang berfungsi untuk membuat *website* dinamis maupun aplikasi *web*. Berbeda dengan HTML yang hanya bias menampilkan konten statis, PHP bisa berinteraksi dengan *database*, *file* dan *folder*, sehingga membuat PHP bisa menampilkan konten yang dinamis dari sebuah *website*. *Blog*, toko online, CMS, forum, dan *website social networking* adalah contoh aplikasi *web* yang bisa dibuat oleh PHP. PHP adalah bahasa *scripting*, bukan bahasa *tag-based* seperti HTML. PHP termasuk Bahasa yang

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

*cross-platform*, ini artinya PHP bisa berjalan pada sistem operasi yang berbeda-beda (Windows, Linux, ataupun Mac). Program PHP ditulis dalam file plain text (teks biasa) dan mempunyai akhiran *.php* (Yuliano, 2007)

PHP ditulis (diciptakan) oleh Rasmus Lerdorf, seorang software engineer asal Greenland sekitar tahun 1995. Pada awalnya PHP digunakan Rasmus hanya sebagai pencatat jumlah pengunjung pada *website* pribadi beliau. Karena itu bahasa tersebut dinamakan *Personal Home Page (PHP) Tools*. Tetapi karena perkembangannya yang cukup disukai oleh komunitasnya, maka beliau pun merilis bahasa PHP tersebut ke publik dengan lisensi open-source. Saat ini, PHP adalah *server-side scripting* yang paling banyak digunakan di *website-website* di seluruh dunia, dengan versi sudah mencapai versi 5 dan statistiknya terus bertambah.

## 2.15 MySQL

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar SQL (*Structured Query Language*). MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *Shareware* (Nugroho, 2005).

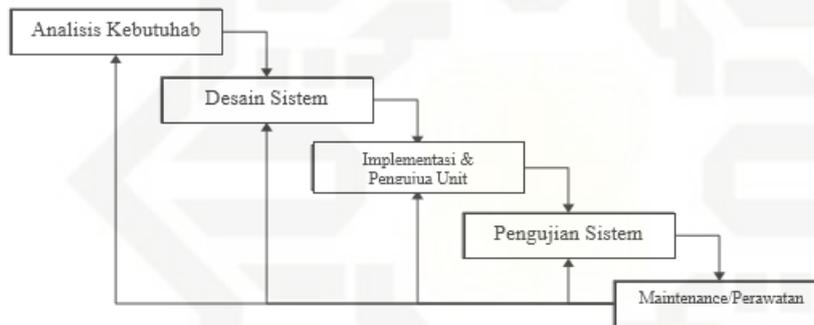
MySQL memiliki beberapa kelebihan dibandingkan database lain diantaranya:

1. MySQL sebagai *database management sistem (DBMS)*.
2. MySQL sebagai *relation database management Sistem (RDBMS)*.
3. MySQL sebuah *software database* yang *opensource* bersifat *free* bebas digunakan oleh siapa saja.
4. MySQL merupakan sebuah *database server*.
5. MySQL merupakan sebuah *database client*.
6. MySQL mampu menerima *query* yang bertumpuk dalam satu permintaan.
7. MySQL sebuah database yang mampu menyimpan data berkapasitas sangat besar.
8. MySQL didukung oleh *driver ODBC* MySQL dapat diakses menggunakan aplikasi apa saja.
9. MySQL *database* menggunakan enkripsi *password*.

10. MySQL *server database* yang multi *user*.
11. MySQL dapat menciptakan lebih dari 16 kunci *portable*.
12. MySQL mendukung *field* yang dijadikan sebagai kunci primer dan kunci unik.
13. MySQL didukung oleh sebuah component C dan perl API.

### 2.16 Model Waterfall

Dalam pembuatan Sistem pendataan dan administrasi Laboratorium ini, digunakan metodologi pengembangan sistem dengan model *waterfall*. Tahapan-tahapan Metode Pengembangan Sistem dalam suatu model *waterfall* berupa perencanaan, analisis sistem, desain sistem, implementasi sistem, perawatan sistem (Jogiyanto, 2005). *Waterfall model* dapat dilihat pada Gambar 2.2 di bawah ini.



Gambar 2.2 *Waterfall Model*  
 (Sumber: Jogiyanto, 2005)

### 2.17 Bootstrap

*Bootstrap* merupakan *framework* CSS yang dapat digunakan untuk mempermudah membangun tampilan *web*. *Bootstrap* pertama kali di kembangkan pada pertengahan 2010 di Twitter oleh @mdo (Mark Otto) dan @fat (Jacob Thornton). *Bootstrap* merupakan *blueprint* dari *Twitter*. Awal dirilis pada Jumat, 19 Agustus 2011. *Bootstrap* sudah memiliki lebih dari dua puluh rilis, termasuk dua penulisan ulang mayor dengan v2 dan v3 (bootstrap, 2017).

*Bootsrap* merupakan sebuah *framework* css yang menyediakan komponen-komponen antarmuka siap pakai dan telah dirancang sedemikian rupa untuk keperluan desain halaman *website* yang artistik.