



BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian tentang sistem informasi peminjaman alat sebelumnya dilakukan oleh Sigit Teguh Prakoso, Sugeng Widodo, Yekti Asmoro Kanthi pada Mapala se-kota Malang. Sistem bermanfaat untuk mempermudah peminjaman peralatan antar Organisasi Mapala Se-kota Malang. Sistem ini dirancang untuk memudahkan dalam memberikan informasi tentang peralatan yang dimiliki dan status peminjaman. Selain itu sistem dapat melakukan transaksi peminjaman, pengembalian, dan inventaris barang. [11]

Selanjutnya penelitian dilakukan oleh Lutfiana Amin, Prihastanto, Rina Marlia. Alasan untuk menggunakan sistem informasi diantaranya adalah kecepatan pengolahan yang lebih besar, ketepatan dan konsistensi yang lebih baik, keamanan yang lebih baik, pencapaian informasi lebih cepat dan mudah. Sistem ini menggunakan metode pengembangan sistem model waterfall.[12]

Penelitian lainnya dilakukan oleh Mikael Yurubeli, Ria Arafiyah, Med Irza. Berdasarkan hasil uji coba, sistem informasi administrasi dengan fitur pendaftaran praktikum, penelitian, asisten serta inventaris laboratorium berhasil diimplementasikan dan dapat menjadi alternatif dalam permasalahan administrasi yang terjadi pada laboratorium kimia. Sistem informasi administrasi pada laboratorium kimia Fakultas MIPA Universitas Negeri Jakarta dirancang dan diimplementasikan merujuk pada System Develop Life Cycle (SLDC) model waterfall dengan tahapan analisis kebutuhan, desain sistem, implementasi sistem, uji coba dan pemeliharaan.[5]

Dapat disimpulkan dari penelitian sebelumnya, pada penelitian pertama menggunakan sistem hanya menampilkan informasi data alat dan pencatatan peminjaman. Sedangkan sistem yang akan penulis buat dapat menampilkan bukti peminjaman, bukti pengembalian, denda dan laporan peminjaman kepada kepala program. Pada penelitian kedua menggunakan Model Waterfall, sedangkan pada penelitian penulis menggunakan Model Spiral. Maka dari itu peneliti bermaksud mengembangkan Sistem Informasi Peminjaman Alat Praktikum Multimedia dengan bahasa pemrograman PHP dan *Framework Code*

Igniter dan menggunakan metode pengembangan Spiral dengan fitur tambahan denda dan tersedianya laporan kepada Kepala Program.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Menurut Mulyanto sistem adalah prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sistem mempunyai karakteristik, yaitu :

1. **Komponen sistem**
Komponen terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk sub sistem. Setiap subsistem memiliki sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar, yang disebut dengan supra sistem.
2. **Batasan Sistem (Boundary)**
Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan dan menunjukkan ruang lingkup dari sistem tersebut.
3. **Lingkungan Luar Sistem (Environment)**
Apapun bentuk yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga merugikan sistem tersebut.
4. **Penghubung Sistem (Interface)**
Interface adalah media untuk menghubungkan sistem dengan subsistem lainnya.
5. **Masukan Sistem (Input)**
Masukan (input) sistem adalah segala sesuatu yang masuk kedalam sistem dan menjadi bahan untuk diproses, masukan dapat berupa hal-hal berwujud (tampak secara fisik) maupun yang tidak tampak.

6. Keluaran Sistem (Output)

Keluaran (Output) merupakan hasil pemrosesan yang diolah menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain. Seperti contoh sistem informasi, keluaran yang dihasilkan bisa berupa suatu informasi, saran, cetakan laporan dan keluaran yang dihasilkan ini dapat digunakan sebagai masukan untuk pengambilan keputusan.

7. Pengolah Sistem (Process)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran yang berguna.

8. Sasaran Sistem (Objective)

Suatu sistem pasti memiliki tujuan dan sasaran. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

Informasi adalah data yang telah diproses ke dalam suatu bentuk yang mempunyai makna. Informasi mempunyai kualitas informasi yang digunakan untuk mengukur informasi yang dihasilkan dari suatu sistem informasi dengan kualitas yang dapat memberikan nilai bagi pengguna sistem tertentu dengan karakteristik informasi yang sesuai dengan kebutuhan pengguna [10]. Kualitas informasi mengukur kualitas keluaran dari sistem informasi [11]. Berdasarkan pendapat para ahli dapat disimpulkan bahwa kualitas informasi adalah suatu pengukuran yang berfokus pada keluaran yang di produksi oleh sistem, serta nilai dari keluaran bagi penggunaan. Jogyanto menjelaskan bahwa kualitas informasi terdiri tiga hal, yaitu [11] :

1. Akurat, informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Informasi harus memiliki keakuratan tertentu agar tidak diragukan kebenarannya.
2. Tepat pada waktunya, informasi yang datang pada penerima tidak boleh datang terlambat, karena informasi yang datang tidak tepat waktu, tidak bernilai lagi, sebab informasi digunakan dalam proses pembuatan keputusan.
3. Relevan, informasi yang ada memiliki nilai kemanfaatan sesuai dengan yang dibutuhkan oleh pemakainya. Informasi memiliki tingkat relativitas yang berbeda, tergantung pada tingkat pemakai.

Jadi sistem informasi adalah suatu sistem di dalam organisasi untuk mengelola transaksi harian yang mendukung fungsi dari

organisasi yang bersifat manajerial. Sistem informasi adalah suatu kombinasi dari orang-orang, hardware, software, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi. Komponen sistem informasi antara lain [12] :

1. Komponen input, data yang masuk ke dalam sistem informasi.
2. Komponen output, hasil keluaran dari suatu sistem informasi.
3. Komponen teknologi, alat dalam sistem informasi. Teknologi digunakan untuk menerima output, mengakses data, menghasilkan dan mengirimkan sebuah output.
4. Komponen basis data, kumpulan data yang berhubungan dan tersimpan di komputer dengan menggunakan software database.
5. Komponen kontrol, komponen yang mengendalikan gangguan terhadap sistem informasi.

2.2.2 Peminjaman dan Pengembalian

- a. Peminjaman
Peminjaman alat adalah memberikan alat laboratorium untuk digunakan manfaatnya dalam batas waktu tertentu.
- b. Pengembalian
Pengembalian alat adalah memberikan kembali alat yang telah dipakai kepada petugas laboratorium sesuai dengan kondisi alat sebelumnya yaitu keadaan bersih, baik, rapih dan lengkap.

2.2.3 UML (Unified Modelling Language)





Unified Modelling Language (UML) adalah sebuah bahasa yang menjadi standar untuk merancang dan mendokumentasikan sistem perangkat lunak. UML juga menawarkan standar untuk merancang suatu model dalam sistem. UML digunakan untuk aplikasi perangkat lunak dan aplikasi tersebut dapat berjalan diperangkat keras [4]. Macam-macam UML antara lain :[5]



1. *Use Case Diagram*

Use Case Diagram adalah pemodelan untuk menentukan kelakuan pada sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang terdapat pada sistem dan siapa

saja yang dapat menggunakan fungsi tersebut [6]. Simbol *use case* diagram dapat dilihat pada tabel 2.2.

Tabel 2.1 Simbol Usecase


No	Nama	Simbol	Keterangan
1	<i>Use Case</i>		Fungsi yang disediakan sistem untuk unit yang saling bertukar pesan antar unit dan actor
2	<i>Actor</i>		Orang yang berinteraksi dengan sistem informasi yang dibuat
3	<i>Asosiation</i>		Penghubung komunikasi antar aktor dan <i>use case</i>
4	<i>Ekstend</i>		Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.


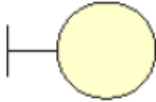



5	<i>InZclude</i>		Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
6	Generalisasi		Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .

2. *Sequence Diagram*

Sequence Diagram adalah gambaran kelakuan objek terhadap *use case* dengan mendefinisikan pesan yang dikirim dan diterima antar objek [6]. Simbol *sequence diagram* dapat dilihat pada tabel 2.3.

Tabel 2. 2 Simbol Sequence Diagram

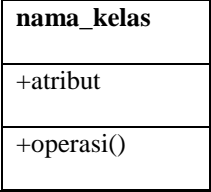
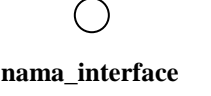

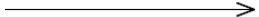
No	Nama	Simbol	Keterangan
1	<i>Actor</i>		Menggambaran orang yang sedang berinteraksi dengan system


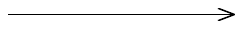
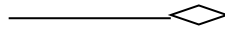
2	<i>Entity Class</i>		Menggambarkan hubungan yang akan dilakukan
3	<i>Boundary Class</i>		Menggambarkan sebuah gambaran dari <i>form</i>
4	<i>Control Class</i>		Menggambarkan hubungan antara <i>boundary</i> dengan <i>table</i>
5	<i>Lifeline</i>		Menggambarkan tempat mulai dan berakhirnya <i>message</i>
6	<i>Line Message</i>		Menggambarkan pengiriman <i>message</i> antar <i>class</i>

3. Class Diagram

Class diagram dibuat agar pembuat program atau programmer membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Berikut Tabel 2.4 adalah simbol-simbol yang ada pada *class diagram* [7].

Tabel 2.3 Simbol-Simbol Class Diagram

No.	Simbol	Nama	Keterangan
1.		Kelas	Kelas pada struktur sistem.
2.		Antarmuka/ <i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.		<i>Asosiasi / association</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.		Asosiasi berarah / <i>directed association</i>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

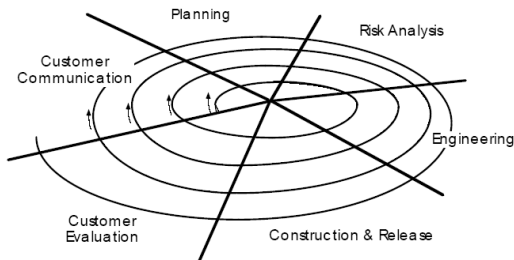
No.	Simbol	Nama	Keterangan
5.		Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
6.		Kebergantungan/dependency	Relasi antarkelas dengan makna kebergantungan antarkelas
7.		Agregasi/aggregation	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

2.2.4 Rekayasa Perangkat Lunak

Perangkat lunak adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, modal desain, dan cara penggunaan (*user manual*). Rekayasa perangkat lunak (*software engineering*) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin.

2.2.5 Metode Pengembangan Sistem

Metodelogi pengembangan sistem yang dipakai adalah Metode Spiral. Spiral merupakan sebuah model proses yang menggabungkan dari model proses prototype dan model waterfall.



Gambar 1.2 Metode Spiral

Tahapan metode spiral adalah :

1. *Customer Communication*
Aktivitas yang dibutuhkan untuk membangun komunikasi yang efektif antara developer dengan user atau customer terutama mengenai kebutuhan dari customer.[2]
2. *Planning*
Aktivitas perencanaan ini dibutuhkan untuk menentukan sumberdaya, perkiraan waktu pengerjaan, dan informasi lainnya yang dibutuhkan untuk pengembangan software.[2]
3. *Risk analysis*
Aktivitas analisis resiko ini dijalankan untuk menganalisis baik resiko secara teknikal maupun secara managerial. Tahap inilah yang mungkin tidak ada pada model proses yang juga menggunakan metode iterasi, tetapi hanya dilakukan pada Spiral Model.[2]
4. *Engineering*
Aktivitas yang dibutuhkan untuk membangun 1 atau lebih representasi dari aplikasi secara teknikal.[2]
5. *Construction & Release*
Aktivitas yang dibutuhkan untuk develop software, testing, instalasi dan penyediaan user atau customer support seperti training penggunaan software serta dokumentasi seperti buku manual penggunaan software.[2]
6. *Customer Evaluation*
Aktivitas yang dibutuhkan untuk mendapatkan feedback dari user atau customer berdasarkan evaluasi mereka

selama representasi software pada tahap engineering maupun pada implementasi selama instalasi software pada tahap construction and release.[2]

3.2.5 Metode Pengujian Sistem

Metode *black box* pengujian sistem bertujuan untuk melihat apakah sistem yang telah dibuat sudah sesuai dengan tujuan awal pembuatan dan layak untuk dipergunakan. Pengujian pada sistem menggunakan metode *black box*, tujuannya untuk mengetahui bahwa bagian-bagian dalam sistem aplikasi telah benar menampilkan pesan-pesan kesalahan jika terjadi kesalahan dalam penginputan data (Sandy 2015). *Black box* Testingsendiri merupakan pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak [1].

Berikut adalah 10 tipe pengujian dari metode *black box* :

1. *Equivalence Partitioning* :
Membagi inputan menjadi kelas data yang dapat digunakan untuk menggenerasi kasus uji.
2. *Boundary Value Analysis / Limit Testing*:
Mengijinkan untuk menyeleksi kasus uji yang menguji batasan nilai input, Merupakan komplemen dari Equivalence Partitioning.
3. *Comparison Testing* :
Uji setiap versi dengan data yang sama untuk memastikan semua versi menghasilkan keluaran yang sama.
4. *Sample Testing* :
Melibatkan beberapa nilai yang terpilih dari sebuah kelas ekivalen.
5. *Robustness Testing* :
Data input dipilih diluar spesifikasi yang telah didefinisikan, Tujuan dari pengujian ini adalah membuktikan bahwa tidak ada kesalahan jika masukan tidak valid.
6. *Behavior Testing* :
Hasil uji tidak dapat dievaluasi jika hanya melakukan pengujian sekali, tapi dapat dievaluasi jika pengujian dilakukan beberapa kali, misalnya pada pengujian struktur data stack.
7. *Performance Testing* :



Mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan misalnya : aliran data, ukuran pemakaian memori, kecepatan eksekusi.

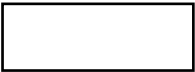

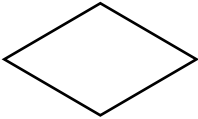
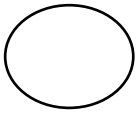
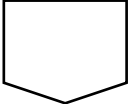

8. *Requirement Testing* :
Spesifikasi kebutuhan yang terasosiasi dengan perangkat lunak diidentifikasi pada tahap spesifikasi kebutuhan dan desain.
9. *Endurance Testing* :
Melibatkan kasus uji yang diulang-ulang dengan jumlah tertentu.
10. *Cause –Effect Relationship Testing* :
Bagi-bagi spesifikasi kebutuhan menjadi bagian yang memiliki kemungkinan kerja.



2.2.6 Flowchart

Flowchart adalah bagan dengan simbol yang menggambarkan suatu proses dengan detail. Simbol *flowchart* juga dipakai sebagai alat bantu untuk mendefinisikan sebuah proses. Proses *flowchart* digambarkan dari halaman atas ke bawah dan dari kiri ke kanan. Aktifitas yang digambarkan harus didefinisikan dengan jelas dan dapat dimengerti oleh pembacanya [3]. Simbol *flowchart* dapat dilihat di Tabel 2.1.

Tabel 2.4 Simbol Flowchart

No	Nama	Simbol	Keterangan
1	Terminal		Menyatakan permulaan atau akhir suatu program
2	<i>Input</i> dan <i>Output</i>		Menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya


3	Proses		Menyatakan proses yang dilakukan oleh computer
4	Simbol arus / <i>flow</i>		Menyatakan jalannya arus suatu proses
5	<i>Decision / keputusan</i>		Menunjukkan suatu kondisi tertentu yang akan menghasilkan data kemungkinan dengan jawaban ya atau tidak
6	<i>Connector (on-page)</i>		Menyederhanakan hubungan antar simbol yang letaknya berjauhan
7	<i>Connector (off-page)</i>		Menghubungkan simbol dalam halman berbeda
8	<i>Manual Operation</i>		Menunjukkan kegiatan yang tidak dilakukan oleh computer

9	Dokumen		Mencetak laporan ke printer
10	Manual Input		Memasukkan data secara manual dengan menggunakan online keyboard

2.2.7 ERD (Entity Relation Diagram)

Entity Relationship Diagram (ERD) adalah model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model. Didalam hubungan ini tersebut dinyatakan yang utama dari ERD adalah menunjukkan objek data dan hubungan yang ada pada Entity berikutnya [8]. Simbol *entity relationship diagram* (ERD) dapat dilihat pada tabel 2.5.

Tabel 2.5 Simbol Erd

No	Nama	Simbol	Keterangan
1	<i>Entity</i>		Menyatakan himpunan entitas yang perlu disimpan dalam database

2	<i>Relationship</i>		Mengetahui jenis hubungan yang ada antara 2 file
3	<i>Atribute</i>		Entitas atau relasi yang menyediakan penjelasan detail tentang entitas atau relasi
4	<i>Multi Valued Atribut</i>		Atribut yang memiliki nilai lebih dari satu setiap baris data
5	<i>Primary Key</i>		Atribut yang menjadi kunci utama dan tidak boleh sama dengan atribut yang lain
6	Alur		Menghubungkan atribut dengan entitas dan entitas dengan relasi

2.2.8 Rekayasa Web

Rekayasa web adalah sub dari rekayasa perangkat lunak yang membantu menyediakan metodologi untuk merancang,

mengembangkan, memelihara dan melibatkan aplikasi web. Tujuan dari rekayasa web untuk kesuksesan dalam mengatur kompleksitas, serta keanekaragaman pengembangan aplikasi web. Menurut Pressman, *web engineering* ini merupakan bagian dari *software engineering* yang dikhususkan dalam pembuatan suatu sistem aplikasi berbasis web berkualitas tinggi [9].

2.2.9 Basis Data

Basis data atau *database* adalah kumpulan dari data-data yang saling terkait dan data-data tersebut dirancang untuk memenuhi kebutuhan informasi. *Database* adalah sebuah hubungan antara tabel satu dengan tabel yang lainnya. Jadi *database* adalah suatu data yang saling terkait dan menggambarkan suatu hubungan tabel satu dengan tabel lainnya yang dirancang untuk memenuhi kebutuhan informasi.[4]

A. DBMS (*Database Management System*)

Database Management System (DBMS) adalah sistem perangkat lunak yang memungkinkan penggunaan untuk mendefinisikan, mengendalikan pembuatan, memelihara dan mengontrol akses ke *database* dalam skala yang besar. Jadi DBMS adalah perangkat lunak yang berinteraksi dengan program aplikasi antara pengguna dan *database* [10]. Adapun bahasa dalam *Database Management System* (DBMS) sebagai berikut :

1. DDL (*Data Definition Language*)

DDL adalah bahasa untuk mendefinisikan sebuah data tipe, struktur serta batasan aturan mengenai data yang akan disimpan ke dalam basis data tersebut [11]. Perintah DDL (*Data Definition Language*) diantaranya :

a. *Create*

Create adalah perintah yang digunakan untuk membuat tabel pada basis data.

Contoh :

```
CREATE DATABASE peminjaman_alat;
```

b. *Drop*

Drop adalah perintah untuk menghapus tabel. Penghapusan tabel menggunakan *drop* maka menghapus semua isi dari tabel yang dihapus.

Contoh :

```
DROP DATABASE peminjaman_alat;
DROP TABLE tb_peminjaman;
```

c. *Alter*

Alter adalah perintah untuk menambahkan kolom baru, memodifikasi kolom yang sudah ada

Contoh :

```
ALTER TABLE tb_peminjaman ADD jumlah_alat integer (2);
```

d. *Truncate*

Truncate adalah perintah untuk menghapus isi tabelnya.

Contoh :

```
TRUNCATE TABLE tb_peminjaman;
```

2. **DML (*Data Manipulation Language*)**

DML adalah perintah yang digunakan di basis data untuk mengolah data informasi antara lain, memasukan, mengubah, menghapus, melihat dan mencari data dari *database* [11]. Perintah DML (*Data Manipulation Language*) diantaranya :

a. *Insert*

Insert adalah perintah untuk menambahkan data pada tabel di *database*.

Contoh :

```
INSERT INTO user(nama) values("Arlinda");
INSERT INTO user(nama,jurusan,jenjang_kls)
values("Arlinda","MM","XIIMM1");
```

b. *Update*

Update adalah perintah untuk pembaruan data pada tabel di *database*.

```
UPDATE user SET kelas="XIIMM1" WHERE id_user=1;
```

c. *Delete*

Delete adalah perintah untuk menghapus data pada tabel di *database*

Contoh :

Delete from user WHERE id_user=1;

d. *Select*

Select adalah perintah untuk memilih satu atau beberapa data pada tabel di *database*.

Contoh :

*SELECT * FROM tb_alat;*

B. MySQL

MySQL adalah sebuah pengoperasian sistem basis data untuk memasukan data yang memungkinkan pengoperasian data dengan mudah dan otomatis [10]