

Rancang Bangun Sistem Informasi Manajemen Aset pada Universitas Pamulang

Fajar Desta Putra¹, Joko Riyanto¹, Ahmad Fikri Zulfikar¹

¹ Program Studi Teknik Informatika, Fakultas Teknik, Universitas Pamulang.

Article History

Received:
09.01.2020

Revised:
08.02.2020

Accepted:
29.03.2020

***Corresponding Author:**
Joko Riyanto
Email:
jokoriyanto@unpam.ac.id

This is an open access article,
licensed under: [CC-BY-SA](#)



Abstrak: Universitas Pamulang merupakan perguruan tinggi swasta yang maju dan berkembang yang terdapat pada Kota Tangerang Selatan. Universitas Pamulang memiliki banyak aset sebagai penunjang kegiatannya. Namun saat ini Universitas Pamulang belum memiliki sistem informasi internal untuk mengontrol aset-aset yang dimiliki. Belum adanya sistem informasi manajemen aset tersebut akan mengalami kesulitan dalam penelusuran data-data aset. Solusi untuk mengatasi permasalahan yang ada, membutuhkan sebuah sistem informasi manajemen aset yang dapat menjalankan pengelolaan aset menjadi lebih efisien dan terstruktur, serta memudahkan dalam pendataan jumlah aset, pendataan aset berdasarkan kondisinya, pengelompokan aset berdasarkan jenisnya dan mempermudah dalam melakukan penelusuran data-data aset. Pengembangan sistem yang digunakan yaitu waterfall. Model ini terdiri dari requirement analysis, system and software design, implementation and unit system, system testing, operation and maintenance. Tools yang digunakan dalam penelitian ini adalah system operating windows 10 pro 64 bit, sublime text 3, dan database yang digunakan sebagai penyimpanan data adalah *xampp*. Hasil akhir penelitian ini menghasilkan sebuah Sistem Informasi Manajemen Aset Pada Universitas Pamulang.

Kata Kunci: Berbasis Web, Sistem Informasi, Manajemen Aset, Waterfall.

Designing the Asset Management Information System of Universitas Pamulang

Abstract: Universitas Pamulang is a successful and growing private university located in South Tangerang City. Universitas Pamulang has many assets to support its activities. But currently the Universitas Pamulang does not yet have an internal information system to control the assets owned. There is no asset management information system that will certainly experience difficulties in tracking asset data. Solution to overcome existing problems, of course, it requires an asset management information system that can run asset management more efficiently and structured, and facilitate the data collection of assets, data collection of assets based on their conditions, grouping assets based on their types and making it easier to search asset data. The system development used is waterfall. This model consists of requirement analysis, system and software design, implementation and unit system, system testing, operation and maintenance. The tools used in this research are Windows 10 pro 64 bit operating system, sublime text 3, and the database used as data storage is *xampp*. The final results of this research resulted in an Asset Management Information System at the Universitas Pamulang.

Keywords: Asset Management, Berbasis Web Bases, Information System, Waterfall.



1. Pendahuluan

Perguruan tinggi merupakan suatu institusi yang sangat kompleks dalam menjalankan kegiatannya. Oleh karena itu, dibutuhkan penerapan *Computer Based Information System* (CBIS). Menurut Pambudi dkk menyatakan bahwa [1] *Computer Based Information System* adalah sistem informasi yang diimplementasikan dalam media teknologi komputer sebagai penunjang seluruh operasional yang berjalan dalam suatu organisasi. Umumnya suatu institusi memiliki sistem yang digunakan untuk menunjang kegiatannya, begitu juga dalam pengelolaan aset. Kebutuhan akan sistem informasi untuk mengelola data dari aset institusi yang ada sangat penting guna menunjang operasional suatu institusi tersebut, tidak adanya sistem informasi dan data yang valid dalam mengelola aset tentu saja membuat pengelolaan serta laporan data aset menjadi tidak seimbang dan menghambat kegiatan operasional.

Universitas Pamulang merupakan perguruan tinggi swasta di kota Tangerang Selatan yang maju dan berkembang. Universitas Pamulang memiliki aset-aset sebagai penunjang kegiatannya seperti meja, kursi, papan tulis, komputer, printer, dan lain-lain. Namun saat ini Universitas Pamulang belum memiliki sistem informasi internal untuk mengontrol aset-aset yang dimiliki. Belum adanya sistem informasi manajemen aset tersebut tentu akan mengalami kesulitan dalam penelusuran data-data aset. Permasalahan tersebut dapat menyebabkan pengelolaan aset kesulitan dalam melakukan pendataan jumlah aset, pendataan kondisi aset itu baik, rusak, ataupun dalam keadaan hilang dan pengelompokan aset berdasarkan jenisnya. Dibutuhkan sebuah sistem informasi manajemen aset yang dapat menjalankan pengelolaan aset menjadi lebih efisien dan terstruktur.

2. Landasan Teori

2.1. Rancang Bangun Aplikasi

Menurut Suyoto & Mudjihartono [2], rancang bangun adalah proses pembangunan sistem untuk menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun hanya sebagian. Sari [3] menyatakan bahwa aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, game, pelayanan masyarakat, periklanan atau semua proses yang hampir dilakukan manusia. Peneliti menyimpulkan bahwa rancang bangun aplikasi adalah proses penciptaan sistem baru dalam bentuk perangkat lunak yang lebih baik untuk melayani kebutuhan baik keseluruhan maupun sebagian.

2.2. Sistem Informasi

Menurut Putra dkk [4] sistem informasi adalah suatu sistem 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 dibutuhkan. Sistem ini menyimpan, mengambil, mengubah, mengolah dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya.

2.3.1. Komponen-Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan yang terdiri dari komponen *input*, komponen *model*, komponen *output*, komponen *teknologi*, komponen *hardware*, komponen *software*, komponen basis data, dan komponen kontrol. Semua komponen tersebut saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran.

2.3.2. Elemen-Elemen Sistem Informasi

Sistem informasi terdiri dari elemen-elemen komponen fisik seperti:

- a. Orang (operator, analis sistem, programmer, personil data *entry* dan manajer sistem EDP)
- b. Prosedur berupa buku panduan.
- c. Perangkat keras terdiri atas komputer, peralatan penyiapan data, dan terminal masukan/keluaran.
- d. Perangkat lunak.
- e. Basis data (*disket*, *harddisk*, *magnetic tape*, dan sebagainya). *File* juga meliputi keluaran tercetak dan catatan lain diatas kertas, *mikro film*, dan lain sebagainya.
- f. Jaringan komputer, *printer* dan lainnya yang terhubung dalam satu kesatuan. Informasi dan data bergerak sehingga memungkinkan pengguna jaringan komputer saling bertukar dokumen dan data.
- g. Komunikasi data merupakan bagian vital dari suatu sistem informasi karena sistem ini menyediakan infrastruktur yang memungkinkan komputer dapat berkomunikasi satu sama lain.

2.4. Sistem Informasi Manajemen Aset

Sistem informasi manajemen sebagai sistem berbasis komputer yang menyediakan informasi bagi para pengguna yang memiliki kebutuhan yang sama. Implementasi Sistem Informasi Manajemen Aset pada hakekatnya adalah upaya untuk tertib dokumen dan tertib administrasi pengelolaan aset. Tertib dokumen aset berkaitan dengan upaya penyediaan dan pendataan dokumen yang menyertai keberadaan aset, sedangkan tertib administrasi lebih dimaksudkan pada upaya membangun prosedur pengelolaan aset mulai saat pengadaan, penerimaan, perubahan data, hingga penghapusan aset [5].

2.5. Website

Website diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa *text*, gambar, *video*, *audio*, dan animasi lainnya yang disediakan melalui jalur koneksi *internet*.

2.5.1. Bahasa Pemrograman Website

Bahasa pemrograman merupakan bahasa yang dapat dipahami oleh komputer. Dalam membangun *website*, ada banyak jenis bahasa pemrograman yang dapat digunakan, diantaranya *HTML*, *CSS*, *Javascript*, dan *PHP* [6].

2.5.2. Web Browser

Web browser digunakan untuk menampilkan hasil *website*. *Web browser* yang sering digunakan adalah *Mozilla Firefox*, *Google Chrome*, dan Safari. Disarankan untuk menggunakan lebih dari satu *web browser* untuk memastikan desain *website* dapat tampil sempurna di berbagai *web browser* [7].

2.6. Aset

Aset adalah barang atau sesuatu yang mempunyai nilai ekonomi, nilai komersial atau nilai tukar yang dimiliki oleh instansi, organisasi, badan usaha ataupun individu [8].

2.6.1. Definisi dari Kasus yang Dianalisa

Pentingnya siklus hidup aset diungkapkan oleh *Publicly Available Specification* mendefinisikan manajemen aset sebagai kegiatan sistematis dan terkoordinasi dan praktik melalui optimasi organisasi dan berkelanjutan mengelola aset dan sistem aset, kinerja yang terkait, risiko dan pengeluaran selama siklus hidup aset untuk tujuan mencapai rencana strategis organisasi. Profesional manajemen harus dapat mengelola semua aspek dari siklus hidup aset untuk memastikan perusahaan mereka mencapai hasil maksimal atas modal, menerapkan efisien dan proses yang efektif dan memberikan tingkat pengendalian untuk mendukung kerja serta menghilangkan proses yang tidak memberikan nilai tambah dan menghabiskan sumber daya dan biaya [9].

2.6.2. Manajemen Aset

Manajemen aset merupakan proses pengorganisasian, perencanaan dan pengawasan terhadap pembelian, penggunaan, perawatan, perbaikan, dan penghapusan aset fisik untuk mengoptimalkan potensi *service delivery* dan meminimalkan resiko yang berkaitan dengan usia hidup aset dengan menggunakan aset *intangible* seperti aplikasi pengambilan keputusan berbasis *knowledge* dan proses bisnis [10].

2.7. Pengembangan Sistem

2.7.1. Model Waterfall

Dalam penelitian ini, pengembangan serta perencanaan sistem perangkat lunak menggunakan model *waterfall* [11]. Diagram model *waterfall* dapat dilihat pada Gambar 1.

- a. Analisis Kebutuhan Perangkat Lunak
Pengumpulan kebutuhan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dipahami yang dibutuhkan oleh user. Spesifikasi kebutuhan tahap ini perlu di dokumentasikan.
- b. Desain
Desain perangkat lunak adalah proses yang fokus pada desain pembuatan program perangkat lunak struktur data, arsitektur perangkat lunak, representasi antar muka dan prosedur kode.
- c. Pembuatan Kode Program

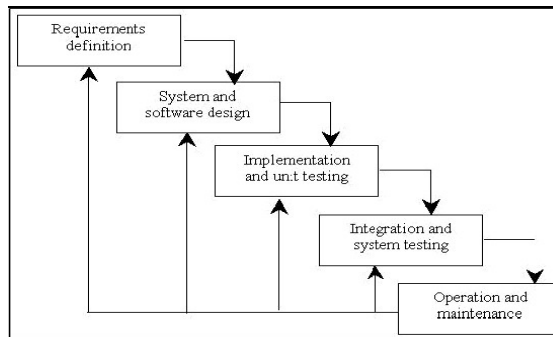
Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

d. Pengujian

Pengujian fokus pada perangkat lunak dari segi logik dan fungsional serta memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

e. Pendukung atau Pemeliharaan

Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak maka dibutuhkan pemeliharaan.



Gambar 1. Model Waterfall

2.7.2. UML

Unified Modeling Language adalah notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek. UML dideskripsikan oleh beberapa diagram.

a. Use Case Diagram

Pembuatan *use case diagram* lebih fokus pada fungsionalitas yang ada pada actor dengan sistem. Simbol pada *use case diagram* dijabarkan pada Tabel 1.

Tabel 1. Simbol-simbol pada Use Case Diagram

Notasi	Nama Elemen	Fungsi
	<i>Actor</i>	<i>Actor</i> dapat berinteraksi dengan <i>specialization</i> atau <i>superclass association</i> . <i>Actor</i> ditempatkan di luar <i>subject boundary</i> .
	<i>Use Case</i>	Mewakili bagian fungsionalitas sistem dalam <i>system boundary</i> .
	<i>Subject Boundary</i>	Menyatakan lingkup dari subjek.
	<i>Association Relationship</i>	Menghubungkan <i>actor</i> untuk berinteraksi dengan <i>use case</i> .
	<i>Include Relationship</i>	Menunjukkan <i>inclusion</i> fungsionalitas sebuah <i>use case</i> dengan <i>use case</i> lainnya. Arah panah dari <i>base use case</i> ke <i>included use case</i> .
	<i>Extend Relationship</i>	Menunjukkan <i>extension</i> dari sebuah <i>use case</i> untuk menambahkan optional <i>behavior</i> . Arah panah <i>extension use case</i> ke <i>base use case</i> .
	<i>Generalization relationship</i>	Menunjukkan generalisasi dari <i>use case</i> khusus ke umum
	<i>Collaboration</i>	Interaksi aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemennya.
	<i>Note</i>	Elemen eksis saat aplikasi dijalankan dan sumber daya komputasi

b. *Class Diagram*

Class adalah spesifikasi yang akan menghasilkan objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut) suatu sistem, serta menawarkan layanan untuk memanipulasi keadaan. Kelas memiliki tiga area pokok yaitu nama, atribut dan metode. Simbol pada *Class Diagram* dapat dilihat pada Tabel 2.

Tabel 2. Simbol-Simbol pada *Class Diagram*

Notasi	Nama Elemen dan Fungsi
	Kelas pada struktur sistem.
	Antarmuka. Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
	Asosiasi adalah relasi antarmuka, asosiasi biasanya disertai dengan <i>multiplicity</i> .
	Asosiasi berarah artinya makna kelas yang satu digunakan oleh kelas yang lain.
	Generalisasi adalah relasi antar kelas dengan makna generalisasi-spesialisasi.
	Agregasi adalah relasi antar kelas dengan makna semua bagian.

c. *Statechart Diagram*

Menggabungkan semua state yang dimiliki dari suatu objek dari suatu *class* dan keadaan yang menyebabkan *state* berubah. *Statechart diagram* tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda. Simbol yang digunakan pada *Statechart Diagram* terdapat pada Tabel 3.

Tabel 3. Simbol-Simbol pada *Statechart Diagram*

Notasi	Nama Elemen	Fungsi
	States	Representasi keadaan sebuah objek selama objek tersebut ada.
	Transition	Suatu anak panah yang tebal adalah jalan antara <i>state</i> yang berlainan dari satu objek. Beri nama transisi dengan kejadian yang menjadi pemicunya dan aksi yang dihasilkan olehnya.
	Initial Pseudo State	Berguna untuk memulai <i>Statechart Diagram</i> .
	Final State	Berguna untuk mengakhiri diagram <i>statechart</i> .

d. *Sequence Diagram*

Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek dan interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. Simbol pada *Sequence Diagram* terdapat pada Tabel 4.

Tabel 4. Simbol-simbol pada *Sequence Diagram*

Notasi	Nama Elemen	Fungsi
	Entity Class	Kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	Boundary Class	Kumpulan kelas menjadi interaksi antar aktor dengan sistem.
	Control Class	Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas.
	Message	Simbol mengirim pesan antar <i>class</i> .
	Recursive	Menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	Activation	Activation mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	Lifeline	Garis terputus dengan objek sepanjang <i>lifeline</i> terdapat <i>activation</i> .

e. *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas dan interaksi beberapa *use case*. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari suatu aktifitas ke aktifitas yang lainnya. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. Simbol yang digunakan pada *Activity Diagram* dapat dilihat pada Tabel 5.

Tabel 5. Simbol-simbol pada *Activity Diagram*

Notasi	Nama Elemen	Fungsi
	<i>Activity</i>	Merepresentasikan sekumpulan aktivitas.
	<i>Control flow</i>	Menunjukkan rangkaian dari suatu eksekusi.
	<i>Initial node</i>	Pertanda dari suatu awal aktivitas.
	<i>Final activity node</i>	Untuk menunjukkan akhir dari suatu aktivitas.
	<i>Decision node</i>	Memastikan bahwa alur objek hanya bergerak dalam satu jalur.
	<i>Merge node</i>	Mengembalikan berbagai <i>decision path</i> menjadi satu.
	<i>Fork node</i>	Membagi perilaku menjadi aktivitas yang berjalan bersama.
	<i>Join node</i>	Menyatukan aktivitas yang berjalan secara bersamaan.

2.8. Database

Basis data terdiri atas dua kata, yaitu basis dan data. Sebagai satu kesatuan istilah, basis data sendiri adalah himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah untuk memenuhi berbagai kebutuhan dan dapat disimpan dalam media elektronis [12].

2.8.1. Entity Relationship Diagram

ERD adalah suatu model jaringan data yang menekankan pada struktur dan *relationship* data [12]. Simbol pada ERD dapat dilihat pada Tabel 6. Penentuan komponen sebagai berikut:

a. Entitas (*Entity*)

Individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Entitas kuat tidak memiliki ketergantungan dengan entitas lainnya. Entitas lemah kemunculannya tergantung pada keberadaan entitas lain dalam relasi.

b. Relasi (*Relationship*)

Menyatukan sekumpulan tipe objek yang dihubungkan panah dan kata menggunakan kata kerja.

c. Atribut

Merupakan suatu sifat entitas atau hubungan dengan maksud memperjelas entitas tersebut. Nilai atribut merupakan suatu informasi tertentu yang disimpan dalam suatu entitas atau *relationship*.

Tabel 6. Simbol-simbol pada *Entity Relationship Diagram*

Simbol	Nama Simbol
	<i>Entity/Entitas</i>
	<i>Relationship/Relasi</i>
	Atribut

2.8.2. Transformasi ERD ke LRS

Tranformasi ERD ke LRS harus diperhatikan karena mempengaruhi yaitu tingkat hubungan *one to one*, *one to many*, atau *many to many*. Setelah transformasi adalah pembentukan LRS [13].

2.8.3. Logical Record Structure

Berikut adalah cara membentuk skema database atau LRS berdasarkan *Entity Relationship Diagram*:

- Relasi 1 to 1, maka *foreign key* diletakan pada salah satu atau menyatukan kedua entitas tersebut.
- Relasi 1 to many, maka *foreign key* diletakan pada entitas *many*.
- Relasi *many to many*, dibuat “file konektor” yang berisi dua *foreign key* berasal dari kedua entitas.

2.8.4. Normalisasi

Normalisasi merupakan proses memecah suatu *file database* yang mengandung permasalahan menjadi dua atau lebih yang sudah tidak mengandung masalah. Setiap *file* terdapat kunci dari *file* berupa *field* yang dapat mewakili *record*. Atribut kunci ini berfungsi sebagai alat untuk mengakses *record* yang diwakilinya. Pada proses normalisasi, terdapat beberapa tahap yang harus ditentukan yaitu:

- a. Bentuk Tidak Normal (*Unnormalize Form*)
- b. Bentuk Normal Pertama (*First Normal Form* atau 1NF)
- c. Bentuk Normal Kedua (*Second Normal Form* atau 2NF)
- d. Bentuk Normal Ketiga (*Third Normal Form* / 3NF)

2.9. Bahasa Pemrograman

2.9.1. HTML

HTML singkatan dari *Hyper Text Markup Language*, yaitu berupa skrip yang berupa tag untuk membuat dan mengatur struktur *website*. Beberapa tugas utama *HTML* dalam membangun *website*, diantaranya: menentukan *layout website*, memformat text, membuat list, membuat table, menyisipkan file (gambar, video dan audio), membuat *link* dan membuat formulir [6].

2.9.2. CSS

CSS (*Cascading Style Sheets*) adalah skrip yang digunakan untuk mengatur desain *website*. Fungsinya memberikan pengaturan yang lengkap agar struktur *website* yang dibuat dengan *HTML* terlihat lebih rapi dan indah [6].

2.9.3. PHP

PHP merupakan secara umum dikenal sebagai bahasa pemrograman *script-script* yang membuat dokumen *HTML* secara *on the fly* yang dieksekusi di *server web*, dokumen *HTML* yang dihasilkan dari suatu aplikasi bukan dokumen *HTML* yang dibuat dengan menggunakan editor teks atau editor *HTML* dikenal juga sebagai bahasa pemrograman *server side* [10].

2.9.4. Javascript

Berbeda dengan *PHP* yang diproses *server*, *javascript* diproses komputer *client*. *Javascript* lebih interaktif dibanding dengan *PHP*. Peran *javascript* dalam membuat *website* adalah memberikan efek animasi yang menarik dan interaktif dalam penanganan *event* oleh pengguna *website* [11].

2.9.5. JQuery

Jquery merupakan salah satu teknik atau kumpulan library *javascript* yang sangat terkenal dengan animasinya. Dengan sedikit sentuhan, animasi dalam *website* mudah kita ciptakan. *Jquery* dapat dianggap sebagai frameworknya *javascript* [12].

2.10. Perangkat Lunak yang Digunakan

2.10.1. Sublime Text Editor

Sublime Text adalah editor teks yang dirancang untuk mengolah potongan kode, *plugin*, dan *markup* serta menulis artikel dan mengetik dalam prosa normal. Keunggulan *Sublime Text* terletak pada kualitas dan kuantitas fiturnya seperti blok multitempat, kursor banyak dan pengolahan split [13].

2.10.2. XAMPP

XAMPP merupakan merupakan paket *php* berbasis *open source* yang dikembangkan oleh sebuah komunitas *open source*. Program yang lain tidak perlu diinstal karena telah disediakan oleh *XAMPP*.

2.10.3. Paket Apache dan Phpmyadmin

Apache berfungsi sebagai web server, yaitu tempat menyimpan file *php* dan lainnya yang diperlukan di dalam *website*, sedangkan *Phpmyadmin* merupakan aplikasi berbasis web yang digunakan untuk membuat *database MySQL* sebagai tempat untuk menyimpan data-data *website*. Keduanya, biasanya sudah disediakan dalam satu paket aplikasi seperti *Appserv* dan *Xampp* [6].

2.10.4. MySQL

MySQL server mengatur akses ke data kita untuk memastikan bahwa sekelompok *user* dapat bekerja dengannya secara bersamaan, untuk menyediakan akses yang cepat ke *database* dan untuk memastikan bahwa hanya *user* yang memiliki otoritas yang dapat memperoleh akses. *MySQL* merupakan salah satu *software* untuk *database server* yang banyak digunakan, *MySQL* bersifat *open source* dan menggunkan *SQL* [12].

2.11. Pengujian Sistem

2.11.1. Pengujian *Black Box*

Black-Box testing adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya, sehingga para tester memandang perangkat lunak seperti layaknya sebuah kotak hitam yang tidak penting dilihat isinya, tapi cukup dikenali proses testing di bagian luar [5].

2.11.2. Pengujian *White Box*

Teknik pengujian yang digunakan yakni *basis path testing*, *control structure testing*, *data flow testing*, *loop testing*. Teknik pengujian dalam *white box* yang sering digunakan adalah *basis path testing*.

3. Analisa dan Perancangan

3.1. Analisa Sistem

Analisa sistem adalah penguraian dari suatu yang sistem utuh ke dalam bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan sehingga dapat diusulkan perbaikan.

3.1.1. Analisa Sistem Saat Ini

Universitas Pamulang belum memiliki sistem informasi internal untuk mengontrol aset yang dimiliki. Permasalahan tersebut dapat menyebabkan pengelolaan aset kesulitan dalam melakukan pendataan jumlah aset, pendataan kondisi aset dan pengelompokan aset berdasarkan jenisnya.

3.1.2. Analisa Sistem Usulan

Sistem manajemen aset yang dibuat berfungsi untuk menyimpan data dengan jumlah banyak dan memberikan kemudahan dalam pendataan berdasarkan kondisi aset serta pengelompokan berdasarkan jenisnya. Sistem manajemen aset ini dirancang untuk melakukan laporan terhadap kondisi aset, sehingga diharapkan dapat memberikan kemudahan terhadap pengelola aset dalam melakukan penelusuran data aset. Secara umum fitur akan dibuat dalam sistem usulan meliputi:

- a) Admin dan Teknisi mempunyai hak akses *account* yang berbeda untuk *login* ke dalam aplikasi.
- b) Admin dan Teknisi dapat melakukan pengelolaan gedung, lantai, kelas, aset, dan jenis aset.
- c) Aplikasi ini dilengkapi fitur untuk pelapor dalam membuat laporan terhadap kondisi aset yang ada.
- d) Laporan yang telah dibuat oleh pelapor dikirimkan kepada admin dan teknisi.
- e) Fitur rekap yang dapat dilihat dan dicetak oleh admin berdasarkan pengalokasian dan kondisi aset.

3.2. Perancangan Basis Data

3.2.1. Entity Relationship Diagram

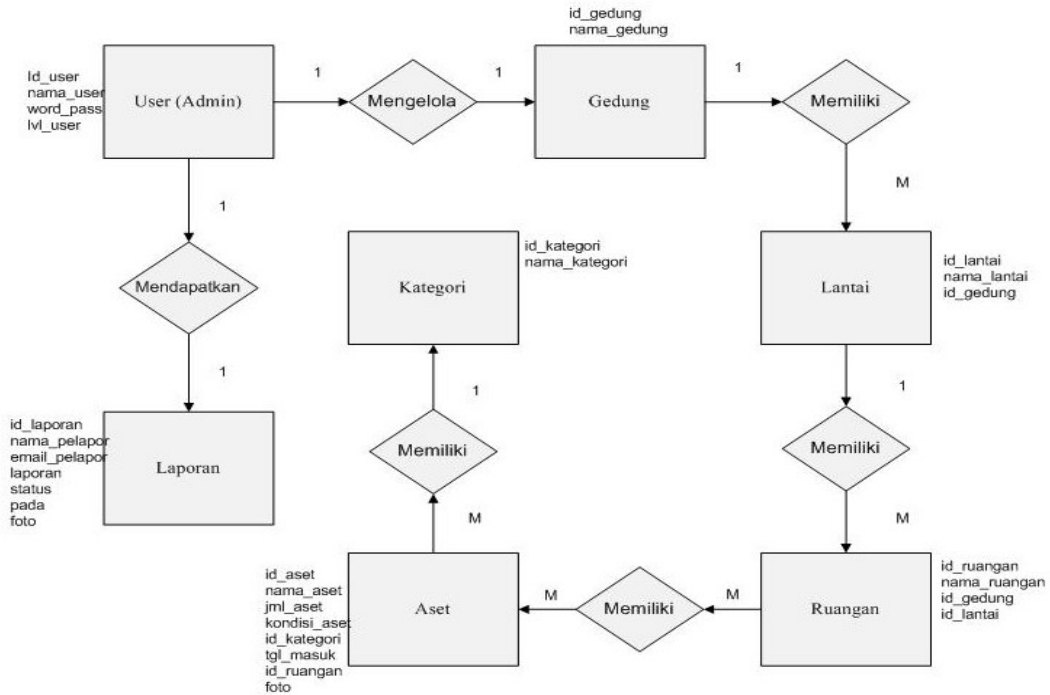
ERD adalah suatu model jaringan data yang menekankan pada struktur dan *relationship* [10], dapat dilihat pada Gambar 2.

3.2.2. Transformasi ERD ke LRS

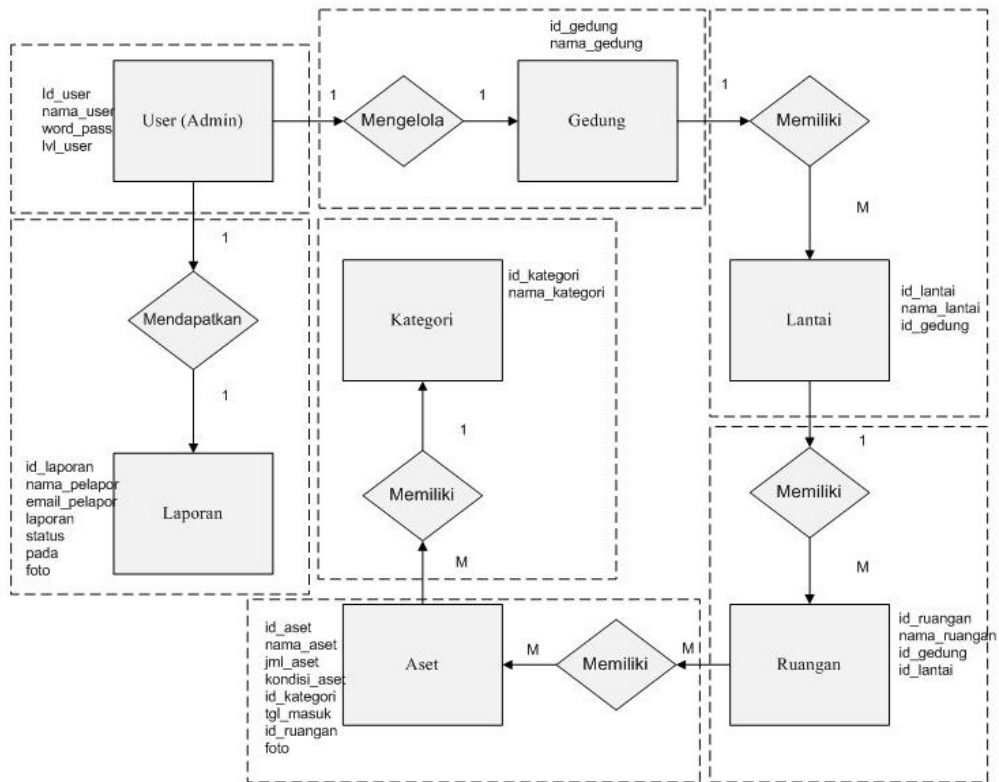
Gambar 3 menunjukkan model transformasi ERD ke LRS.

3.2.3. Logical Record Structure

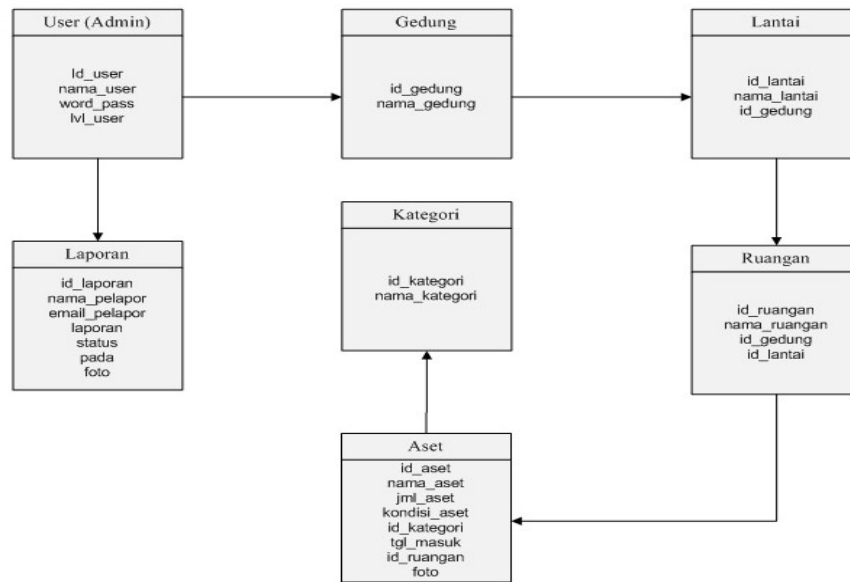
Gambar 4 menunjukkan model jaringan LRS.



Gambar 2. Entity Relationship Diagram



Gambar 3. Transformasi ERD ke LRS



Gambar 4. Logical Record Structure

3.2.4. Spesifikasi Basis Data yang Digunakan

Basis data yang digunakan antara lain basis data untuk User, Gedung, Lantai, Ruang, Kategori, Aset, Laporan.

3.3. Perancangan Aplikasi

Perancangan aplikasi manajemen aset yang dibangun ini bersifat *object orientasi* dengan menggunakan *Unified Modeling Language (UML)* sebagai bahasa pemodelan.

3.3.1. Use Case Diagram

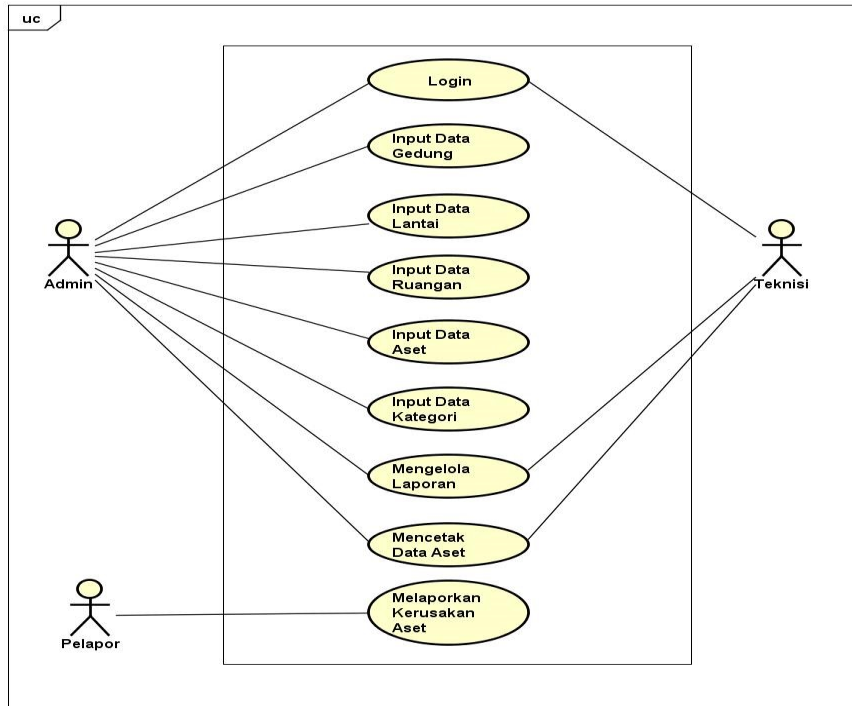
Use Case Diagram adalah gambaran interaksi antara aktor dengan sistem tersebut dalam Gambar 5.

3.3.2. Activity Diagram

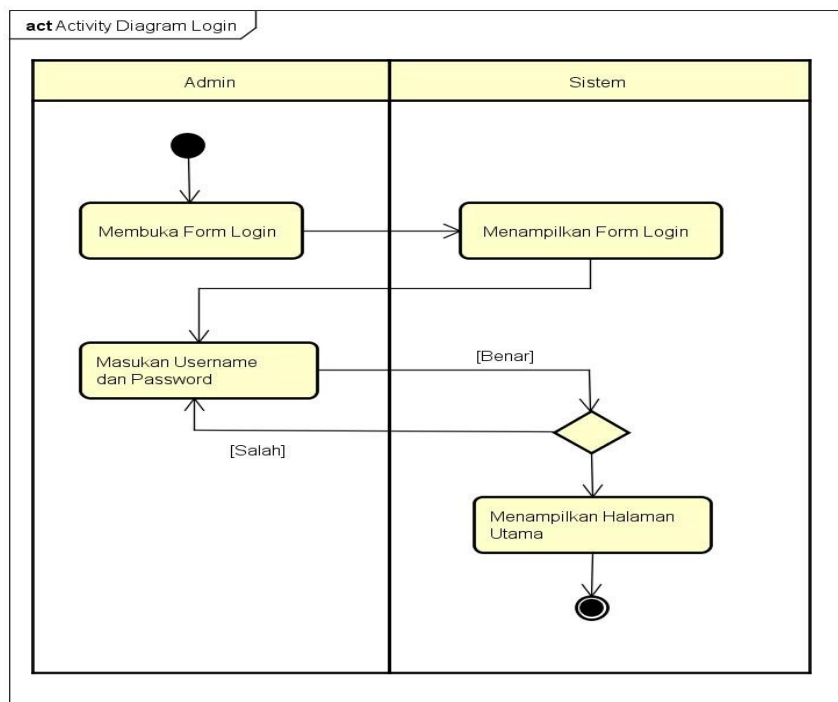
Activity Diagram Login dapat dilihat pada Gambar 6. Beberapa *Activity Diagram* lainnya antara lain:

- *Activity Diagram Input Data Gedung*. Yang mendeskripsikan admin menginput data gedung ke form *input* kemudian disimpan ke *database*. Data akan tampil dalam form data Gedung. Data dapat diedit dan dihapus.
- *Activity Diagram Input Data Lantai*. Yang mendeskripsikan admin menginput data lantai ke form *input*, lalu disimpan ke dalam *database* yang akan tampil dalam form daftar lantai. Data dapat diedit dan dihapus.
- *Activity Diagram Input Data Ruang*. Yang mendeskripsikan admin menginput data ruangan ke form kemudian disimpan ke *database* yang akan tampil dalam form daftar ruangan. Data dapat diedit dan dihapus.
- *Activity Diagram Input Data Kategori*. Yang mendeskripsikan admin menginput data kategori aset ke form kemudian disimpan ke dalam *database*, lalu tampil dalam form daftar kategori. Data dapat diedit, dan dihapus.
- *Activity Diagram Input Data Aset*. Yang mendeskripsikan admin menginput data aset ke form lalu disimpan ke dalam *database*. Sehingga data akan tampil dalam form daftar aset. Data dapat diedit, dan dihapus.
- *Activity Diagram Cetak Data Aset*. Yang mendeskripsikan teknisi dalam mencetak data aset.
- *Activity Diagram Olah Data Kondisi Aset*. Yang mendeskripsikan teknisi dalam mengolah kondisi aset, dimulai dengan login kedalam sistem lalu memilih menu daftar aset. Teknisi mengolah status kondisi aset setelah berhasil sistem akan menampilkan perubahan pada kondisi aset.

- *Activity Diagram* Input Laporan. Yang mendeskripsikan aktifitas pelapor dalam melaporkan kerusakan aset. Selanjutnya data akan tersimpan ke dalam database akan ditampilkan ke dalam menu laporan admin/teknisi.



Gambar 5. Use Case Diagram



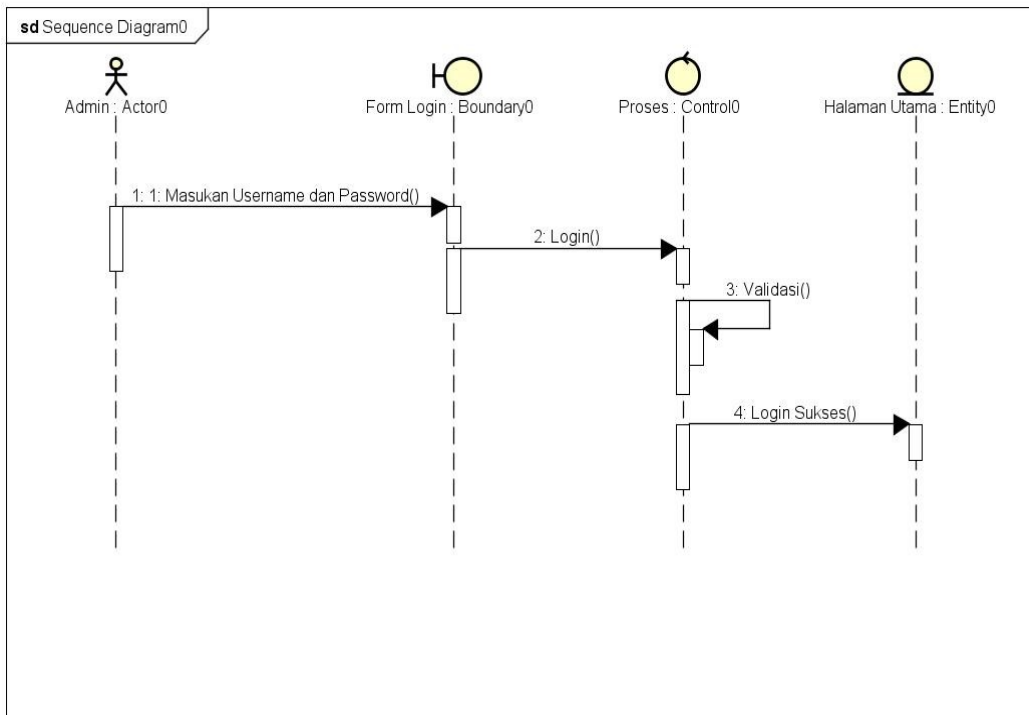
Gambar 6. Activity Diagram Login

3.3.3. Sequence Diagram

Sequence Diagram Login dilihat pada Gambar 7.

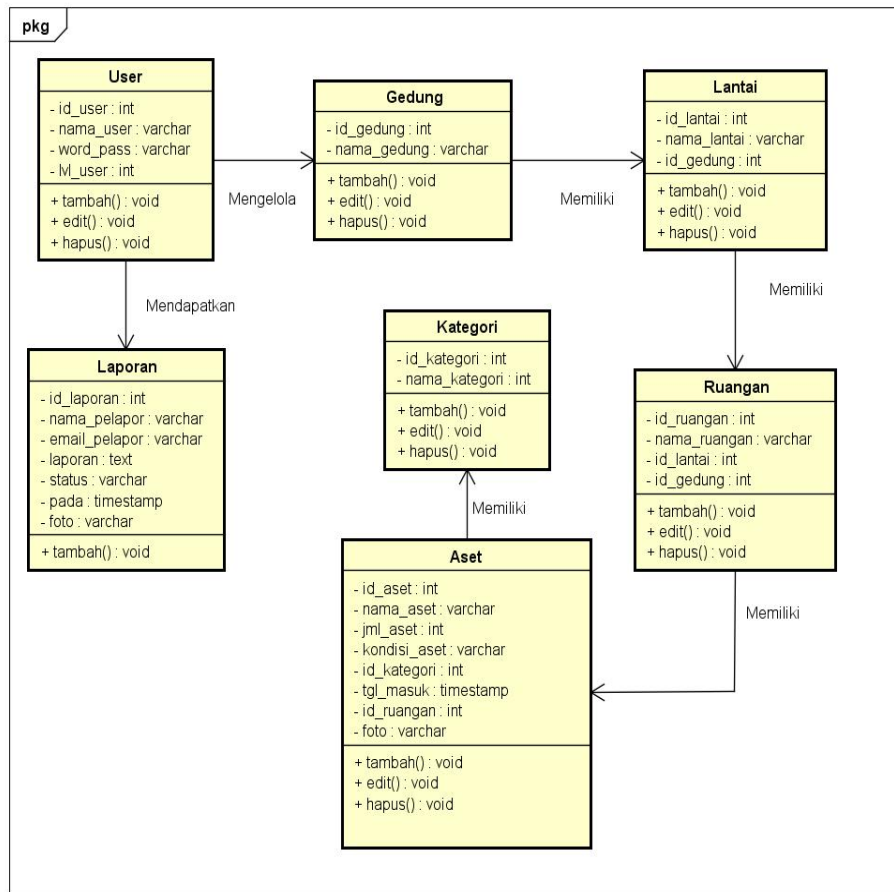
Beberapa *Sequence Diagram* lainnya antara lain:

- *Sequence Diagram Input Data Gedung*. Yang mendeskripsikan admin memasukan data gedung lalu menyimpannya, kemudian sistem memvalidasi jika berhasil maka data akan tersimpan ke dalam *database* dan tampil ke form daftar gedung. Data dapat diedit dan dihapus.
- *Sequence Diagram Input Data Lantai*. Yang mendeskripsikan admin memasukan data lantai valid lalu menyimpannya ke database dan tampil ke form daftar lantai. Dalam form daftar lantai juga data dapat diedit dan dihapus.
- *Sequence Diagram Input Data Ruangan*. Yang mendeskripsikan memasukan data ruangan valid lalu menyimpannya ke dalam database dan tampil ke form daftar ruangan. Dalam form daftar ruangan juga data dapat diedit dan dihapus.
- *Sequence Diagram Input Data Aset*. Yang mendeskripsikan admin *input* data aset yang *valid* lalu menyimpannya lalu tampil ke form daftar aset. Dalam form daftar aset juga data dapat diedit dan dihapus.
- *Activity Diagram Input Data Kategori*. Yang mendeskripsikan admin *input* data kategori aset valid lalu menyimpannya ke dalam database dan tampil ke form daftar kategori. Dalam form daftar kategori juga data dapat diedit dan dihapus.
- *Sequence Diagram Mencetak Data Aset*. Yang mendeskripsikan admin melakukan cetak data aset, membuka form rekap data aset lalu sistem akan menampilkan data aset pada form rekap data asset lalu mencetak data aset.
- *Sequence Diagram Input Laporan*. Yang mendeskripsikan aktifitas oleh pelapor dengan membuka aplikasi maka akan tampil form input laporan selanjutnya memasukan lalu kirim laporan.



Gambar 7. Sequence Diagram Login

3.3.4. Class Diagram



powered by Astah

Gambar 8. Class Diagram

3.4. Perancangan Antar Muka

3.4.1. Perancangan Halaman Membuat Laporan

————— Buat Laporan —————

Nama Pelapor

Alamat Email

Silahkan isi laporan dengan detail

No file chosen

Admin ? [Login disini](#)

SIMASET!

©copyright 2018 Universitas Pamulang | Fajar Desta

Gambar 9. Perancangan Halaman Membuat Laporan

Beberapa perancangan halaman yang lain, yaitu:

- Perancangan Halaman Login Admin dan Teknisi
- Perancangan Halaman Utama
- Perancangan Halaman Tambah Data Gedung
- Perancangan Halaman Daftar Gedung
- Perancangan Halaman Tambah Data Lantai
- Perancangan Halaman Daftar Lantai
- Perancangan Halaman Tambah Data Ruangan
- Perancangan Halaman Daftar Ruangan
- Perancangan Halaman Tambah Data Aset
- Perancangan Halaman Daftar Aset
- Perancangan Halaman Tambah Data Kategori
- Perancangan Halaman Daftar Kategori
- Perancangan Halaman Lihat Laporan
- Perancangan Halaman Rekap Aset. Perancangan halaman rekap aset berisi data-data aset yang dapat dicari berdasarkan lokasi aset atau kondisi aset, serta terdapat button cetak.
- Perancangan Halaman Daftar Pengguna
- Perancangan Halaman Tambah Pengguna

4. Implementasi dan Pengujian

4.1. Implementasi Sistem

Tahapan implementasi sistem merupakan tahapan yang dilakukan setelah perancangan selesai dilaksanakan. Adapun tujuan dari implementasi adalah untuk menerapkan dan mengkaji rangkaian sistem sehingga pengguna dapat memberi masukan kepada pengembang sistem.

4.1.1. Spesifikasi Perangkat Lunak yang Digunakan

Spesifikasi perangkat lunak yang digunakan dapat dilihat pada Tabel 7.

Tabel 7. Spesifikasi Perangkat Lunak

No	Jenis Perangkat Lunak	Keterangan
1	Sistem Operasi	<i>Windows 10 Pro 64-bit</i>
2	<i>Web Server</i>	<i>XAMPP v3.2.2</i>
3	<i>Web Browser</i>	<i>Google Chrome</i>
4	<i>Software Aplikasi</i>	<i>Sublime Text 3, Notepad++</i>

4.1.2. Spesifikasi Perangkat Keras yang Digunakan

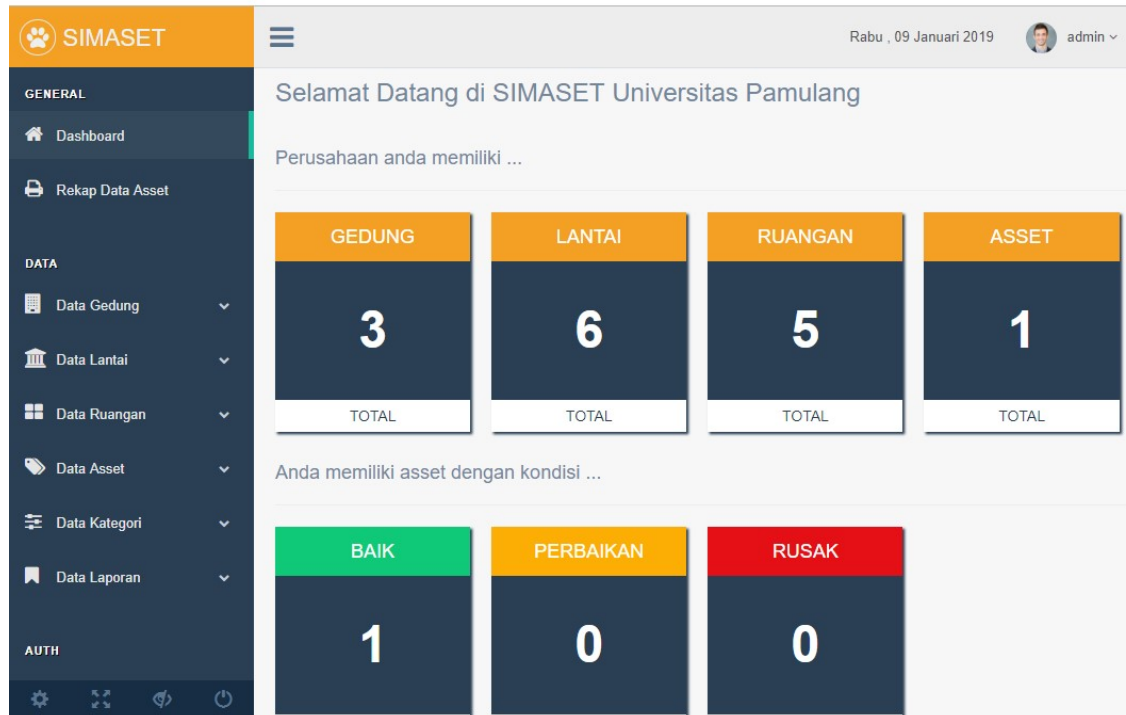
Spesifikasi perangkat keras yang digunakan dapat dilihat pada Tabel 8.

Tabel 8. Spesifikasi Perangkat Keras

No	Nama Hardware	Keterangan
1	<i>Processor</i>	Intel(R) Core(TM) i3-36006U CPU @2.00GHz (4 CPUs), ~2.0GHz
2	<i>Memory RAM</i>	4GB
3	<i>VGA</i>	Nvidia Geforce 920MX
4	<i>Harddisk</i>	500GB
5	<i>Monitor</i>	14 inch

4.2. Implementasi Aplikasi

4.2.1. Tampilan Halaman Utama



Gambar 10. Tampilan Halaman Utama

Beberapa tampilan lain, antara lain:

- Tampilan Form Membuat Laporan
- Tampilan Form Login
- Tampilan Tambah Data Gedung
- Tampilan Daftar Gedung
- Tampilan Tambah Data Lantai
- Tampilan Daftar Lantai
- Tampilan Tambah Data Ruangan
- Tampilan Daftar Ruangan
- Tampilan Tambah Data Aset
- Tampilan Daftar Aset
- Tampilan Tambah Data Kategori
- Tampilan Daftar Kategori
- Tampilan Data Laporan
- Tampilan Rekap Data Aset
- Tampilan Tambah Pengguna
- Tampilan Daftar Pengguna

4.3. Pengujian Sistem

Pengujian sistem merupakan suatu proses pengujian terhadap kelayakan dan kualitas aplikasi yang telah dibuat. Dalam tahapan ini terdapat dua jenis pengujian yaitu pengujian *black box* dan *white box*.


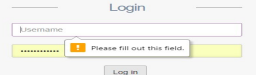
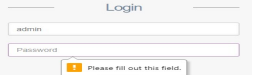
4.3.1. Pengujian *Black Box*

a. Pengujian *Black Box Login*

Tabel 9. Pengujian Black Box Login Data Benar

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Admin input username dan password.		Menampilkan halaman utama.	(√) Diterima () Ditolak

Tabel 10. Pengujian Black Box Login Data Salah

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Admin input username dan password.		Sistem akan menampilkan halaman "Gagal!"	(√) Diterima () Ditolak
Admin input email tidak diisi		Sistem akan menampilkan pesan error "please fill out this field"	(√) Diterima () Ditolak
Admin input password tidak diisi		Sistem akan menampilkan pesan error "please fill out this field"	(√) Diterima () Ditolak

Beberapa pengujian lain, yaitu:

- Pengujian *Black Box* Tambah Data Gedung Normal
- Pengujian *Black Box* Tambah Data Gedung Inputan Kosong
- Pengujian *Black Box* Tambah Data Gedung Inputan Ganda
- Pengujian *Black Box* Tambah Data Lantai Normal
- Pengujian *Black Box* Tambah Data Lantai Inputan Kosong
- Pengujian *Black Box* Tambah Data Lantai Inputan Ganda
- Pengujian *Black Box* Tambah Data Ruangan Normal
- Pengujian *Black Box* Tambah Data Ruangan Inputan Kosong
- Pengujian *Black Box* Tambah Data Ruangan Inputan Ganda
- Pengujian *Black Box* Tambah Aset Normal
- Pengujian *Black Box* Tambah Data Aset Inputan Kosong
- Pengujian *Black Box* Tambah Data Aset Inputan Ganda
- Pengujian *Black Box* Tambah Data Kategori Normal
- Pengujian *Black Box* Tambah Data Kategori Inputan Kosong
- Pengujian *Black Box* Tambah Data Kategori Inputan Ganda
- Pengujian *Black Box* Daftar Gedung
- Pengujian *Black Box* Daftar Lantai
- Pengujian *Black Box* Membuat Laporan
- Pengujian *Black Box* Daftar Ruangan
- Pengujian *Black Box* Daftar Aset
- Pengujian *Black Box* Daftar Kategori
- Pengujian *Black Box* Rekap Data Aset
- Pengujian *Black Box* Membuat Laporan Lanjutan

4.3.2. Tampilan Daftar Pengguna

Adapun jenis pengujian *white box* yang digunakan adalah notasi diagram alir dan kompleksitas siklomatis. Dalam pengujian *white box* ini peneliti hanya melakukan pengujian ke beberapa *form*.

a. Pengujian Login

Tabel 11. Pengujian White Box Login

No	Fungsi	Basic Path
1	<pre><?php class Auth extends MY_Controller { function __construct() { parent::__construct(); \$this->load- >model('auth_model','auth'); } }</pre>	
2	<pre>function index() { \$this->load->view('login'); }</pre>	
3	<pre>function login() { \$post = \$this->input->post(); if(!empty(\$post)) { \$this->db->select('*'); \$this->db- >where('name_user',\$post['name_user']); \$data = \$this->db- >get('user')->result_array(); } }</pre>	<pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 3 --> 4((4)) 3 --> 5((5)) 5 --> 6((6)) 6 --> 8((8)) 7((7)) --> 8 4 --> 8 </pre>
4	<pre>if(empty(\$data)) { \$data['notice'] = 0; \$this->load- >view('login',\$data); return ; }</pre>	<p>Berdasarkan flowgraph diatas, kompleksitas cyclomaticnya dapat dihitung.</p> <ol style="list-style-type: none"> Jalur Independent: <ul style="list-style-type: none"> 1-2-3-4-8 1-2-3-5-6-8 1-2-3-5-7-8 $CC = E - N + 2$ $9 - 8 + 2 = 3$ $CC = P + 1$ $2 + 1 = 3$ Region = 3 <p>Keterangan:</p> <ol style="list-style-type: none"> Jalur Independent: beberapa cara penyelesaian grafik alir bisa sampai tujuan E: Jumlah Edge/garis panah N: Jumlah Node/Simpul P: Jumlah predikat node yang memiliki cabang <p>Region: Jumlah wilayah kompleksitas siklomatik</p>
5	<pre>Else { if(password_verify(\$post['pass_user'], \$data[0]['word_pass'])) { set_cookie('user', \$value = \$data[0]['name_user'], \$expire='7200'); set_cookie('lvl', \$value = \$data[0]['lvl_user'], \$expire='7200'); redirect('dashboard'); } }</pre>	
6	<pre>else { \$data['notice'] = 1; \$this->load->view('login',\$data); return ; }</pre>	
7	<pre>}</pre>	
8	<pre>?></pre>	

b. Pengujian Tambah Data Gedung

Tabel 12. Pengujian White Box Tambah Data Gedung

No	Fungsi	Basic Path
1	<pre><?php class Gedung extends MY_Controller { function __construct() { parent::__construct(); \$this->load- >model('gedung_model','gedung'); } }</pre>	<pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 2 --> 4((4)) 3 --> 5((5)) 4 --> 5 </pre>
2	<pre>function input() { \$post = \$this->input->post(); \$get = \$this->input->get(); if(!empty(\$post)) { \$ret = \$this->gedung- >add(\$post); \$data['notice'] = \$ret; } }</pre>	<p>Berdasarkan flowgraph diatas maka kompleksitas cyclomaticnya dapat dihitung sebaga berikut:</p> <ol style="list-style-type: none"> Jalur Independent: 1-2-3-5 1-2-4-5 $CC = E - N + 2$ $5 - 5 + 2 = 2$ $CC = P + 1$ $1 + 1 = 2$ Region = 2
3	<pre>if(!empty(\$get)) { \$data['data'] = \$this- >gedung->get(\$get['data'])->result_array()[0]; \$this->load- >view('input',\$data); return; }</pre>	<p>Keterangan:</p> <ol style="list-style-type: none"> Jalur Independent: beberapa cara penyelesaian grafik alir bisa sampai tujuan E: Jumlah Edge/garis panah N: Jumlah Node/Simpul P: Jumlah predikat node yang memiliki cabang
4	<pre>\$data['title'] = 'Tambah Data Gedung'; \$data['page'] = 'gedung/input'; \$this->load- >view('theme/layout',\$data); }</pre>	<p>Region: Jumlah wilayah kompleksitas siklomatik</p>
5	}	
6	?>	

4.3.3. Kesimpulan Hasil Pengujian

Sistem manajemen aset yang dibangun ini bebas dari kesalahan sintaks dan secara fungsional memberikan hasil yang sesuai.

5. Penutup

Berdasarkan uraian yang telah ada, maka dapat ditarik kesimpulan sebagai berikut:

- Sistem informasi manajemen aset ini dibangun dengan bahasa pemrograman *php*, dan basis data *mysql* serta model yang digunakan adalah model pengembangan *waterfall*.

- b. Dengan adanya sistem informasi manajemen aset ini dapat memberikan kemudahan dalam pendataan jumlah aset, pendataan kondisi aset, dan pengelompokan aset berdasarkan jenisnya serta memudahkan dalam melakukan penelusuran data-data aset.

Beberapa saran yang dapat peneliti berikan antara lain:

- a. Diharapkan pada penelitian selanjutnya aplikasi ini dikembangkan menjadi sistem pengadaan aset yang dapat menunjang kebutuhan instansi.
- b. Peneliti mengharapkan untuk pengembangan selanjutnya sistem manajemen aset ini dapat dikembangkan ke dalam sistem *mobile* seperti untuk *platform android* ataupun *ios*.

Daftar Pustaka

- [1] G. S. Pambudi, Sriyanto, and A. Arvianto, "Rancang Bangun Sistem Informasi Manajemen Aset Berbasis Web Untuk Optimalisasi Penelusuran Aset Di Teknik Industri Undip Semarang," *Jurnal Teknik Industri*, vol. 11, no. 3, 2016.
- [2] Y. Maryono, Suyoto, and P. Mudjihartono, "Analisis Dan Perancangan Sistem Informasi Manajemen Aset TIK Studi Kasus: Asmi Santa Maria Yogyakarta," *Jurnal Buana Informatika*, vol. 1, no. 2, 2010.
- [3] Y. P. Sari, "Rancang Bangun Aplikasi Penjualan Dan Persediaan Obat pada Apotek Merben di Kota Prabumulih," *Jurnal JSK*, vol. 1, no. 1, 2017.
- [4] R. P. Putra, A. Riyadi, and S. Wardani, "Rancang Bangun Sistem Informasi Persewaan Dan Penjualan Properti Di Diy Berbasis Web," *Seminar Nasional Dinamika Informatika*, vol. 1, no. 2, pp. 302-311, 2017.
- [5] R. A. Sagita, and Hari Sugiarto, "Penerapan Metode Waterfall Pada Sistem Informasi Penjualan Furniture Berbasis Web," *Jurnal Keamanan dan Jaringan*, vol. 5, no. 4, 2016.
- [6] R. Abdulloh, *Membuat Toko Online dengan Teknik Oop, Mvc, dan Ajax*. Jakarta: PT. Elex Media Komputindo, 2017.
- [7] F. Nugraha, B. Sunarso, and B. Noranita, "Sistem Pendukung Keputusan Evaluasi Pemilihan Pemenang Pengadaan Aset dengan Metode Simple Additive Weighting (SAW)," *Jurnal Sistem Informasi Bisnis*, 2012.
- [8] A. Aira, "Peran Manajemen Aset Dalam Pembangunan Daerah," *Jurnal Penelitian Social Keagamaan*, vol. 17, 2014.
- [9] F. Purwaningtias, "Rancang Bangun Pengolahan Data Penyewaan Alat Berat Pada PT. Sumatra Unggul Palembang," *Jurnal Informatika*, vol. 1, 2015.
- [10] E. S. Munthe, "Sistem Penyewaan Kontainer Pada Pt. Putra Guna Jaya Mulia Jakarta," *Jurnal Komputer*, vol. 7, no. 2, 2011.
- [11] D. Hendrianto, "Pembuatan Sistem Informasi Perpustakaan Berbasis Website pada Sekolah Menengah Pertama Negeri 1 Donorojo Kabupaten Pacitan," *Jurnal Keamanan dan Jaringan*, vol. 3, no. 4, 2014.
- [12] F. Iskandar, L. Y. Astri, and D. Kisbianty, "Perancangan Aplikasi Penyewaan Alat Berat berbasis Web pada PT. Indotruck Citra Pramata Jambi," *Jurnal Processor*, vol. 12, no. 2, 2017.
- [13] A. T. Soelistio, T. A. Wibowo, and A. G. Permana, "Aplikasi Sistem Informasi Geografis (SIG) Untuk Pengelolaan Padi Di Pulau Jawa Berbasis Web," *Prosiding Science Applied*, vol. 1, no. 1, 2015.