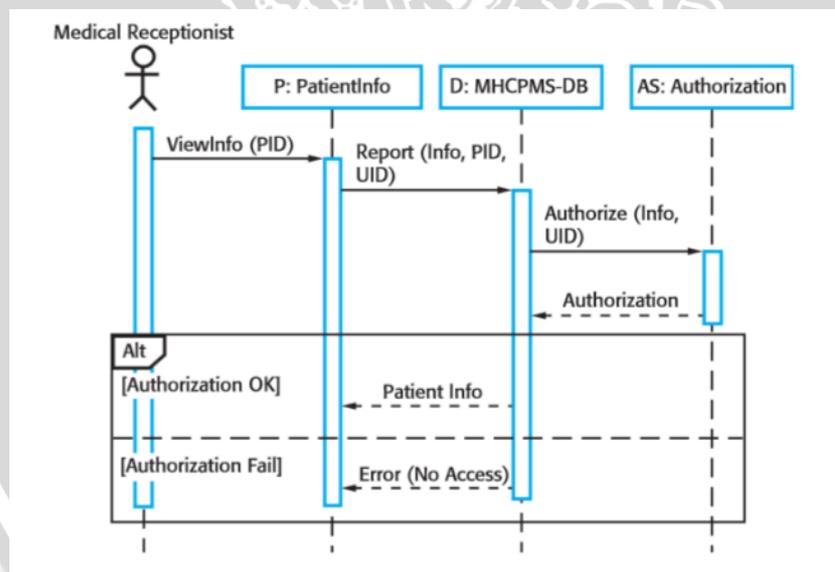


Gambar 2.5 adalah sebuah *activity diagram* yang ditujukan untuk menunjukkan aktifitas yang membuat sebuah proses sistem dan aliran kontrol dari satu aktifitas ke aktifitas lain. Untuk memulai proses dimulai dengan *filled circle*, dan diakhiri dengan lambing lingkaran dalam lingkaran. Persegi dengan segi tumpul menggambarkan aktifitas.

Dalam UML *activity diagram*, garis panah menggambarkan alur kerja dari aktifitas satu ke aktifitas yang lain. Garis tebal digunakan untuk mengindikasikan kordinasi aktifitas. Saat alur mulai lebih dari satu aktifitas mengarah ke garis tebal maka semua aktifitas harus dilakukan sebelum melanjutkan ke tahap berikutnya. Saat alur dari garis tebal menuju aktifitas tertentu, maka aktifitas tersebut boleh dikerjakan secara paralel [SOI-11: 123].

2.3.3 Sequence Diagram



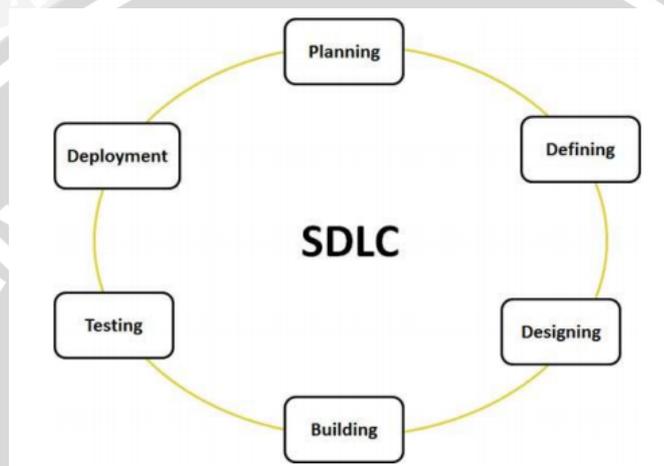
Gambar 2.6 Sequence Diagram

Sumber : [SOI-11: 127]

Sequence diagram dalam UML digunakan untuk memodelkan interaksi antara aktor dan objek dalam sistem dan interaksi dengan dirinya sendiri. UML mempunyai banyak *syntax* untuk *sequence diagram* yang memungkinkan berbagai macam interaksi untuk dimodelkan. Seperti namanya, sebuah *sequence*

diagram menunjukkan urutan dari interaksi yang digunakan selama *use case* tertentu. Gambar 2.6 adalah sebuah contoh *sequence diagram* yang menggambarkan notasi sederhana. Diagram ini memodelkan interaksi yang terjadi dalam tampilan *use case* informasi pasien, dimana resepsionis dapat melihat informasi pasien.

2.4 Software Development Life Cycle



Gambar 2.7 SDLC Cycle

Sumber : [SDL-13: 2]

Sebuah Siklus Pengembangan Perangkat Lunak dalam gambar 2.7 terdiri dari tahapan sebagai berikut:

Tahap 1: Perencanaan dan Analisis Kebutuhan

Analisis Kebutuhan merupakan tahap yang paling penting dan mendasar dalam SDLC. Hal ini dilakukan oleh anggota senior dari tim dengan masukan dari pelanggan, departemen penjualan, survei pasar dan pakar domain dalam industri. Informasi ini kemudian digunakan untuk merencanakan pendekatan proyek dasar dan untuk melakukan studi kelayakan produk dalam ekonomi, operasional, dan teknis daerah. Perencanaan untuk persyaratan jaminan kualitas dan identifikasi risiko yang terkait dengan proyek ini juga dilakukan dalam tahap perencanaan. Hasil dari studi kelayakan teknis adalah untuk menentukan berbagai pendekatan teknis yang dapat diikuti untuk melaksanakan proyek tersebut berhasil dengan resiko minimal.

Tahap 2: Mendefinisikan Persyaratan

Setelah analisis kebutuhan dilakukan langkah berikutnya adalah untuk secara jelas mendefinisikan dan mendokumentasikan persyaratan produk dan membuat mereka disetujui dari pelanggan atau analis pasar. Hal ini dilakukan melalui ' SRS ' - Dokumen Spesifikasi Kebutuhan Software yang terdiri dari semua persyaratan produk yang akan dirancang dan dikembangkan selama siklus hidup proyek.

Tahap 3: Merancang arsitektur produk

SRS adalah referensi bagi arsitek produk untuk keluar dengan arsitektur terbaik bagi produk yang akan dikembangkan. Berdasarkan persyaratan yang ditentukan dalam SRS, biasanya lebih dari satu pendekatan desain untuk arsitektur produk diusulkan dan didokumentasikan dalam DDS - Dokumen Desain Spesifikasi. DDS ini ditinjau oleh semua pemangku kepentingan terkait dan berdasarkan berbagai parameter sebagai risiko.

Tahap 4: Membangun atau Mengembangkan Produk

Pada tahap ini SDLC perkembangan sebenarnya dimulai dan produk dibangun. Kode pemrograman yang dihasilkan sesuai DDS selama tahap ini. Jika desain dilakukan dengan cara yang rinci dan terorganisir, generasi kode dapat dicapai tanpa banyak kerumitan. Pengembang harus mengikuti pedoman pengkodean yang ditetapkan oleh organisasi mereka dan tool pemrograman seperti kompiler, interpreter, debugger dll digunakan untuk menghasilkan kode. Bahasa pemrograman tingkat tinggi yang berbeda seperti C, C ++, Pascal, Java, dan PHP digunakan untuk pemrograman. Bahasa pemrograman yang dipilih sehubungan dengan jenis perangkat lunak yang dikembangkan.

Tahap 5: Pengujian Produk

Tahap ini biasanya subset dari semua tahapan seperti dalam model SDLC modern, kegiatan pengujian sebagian besar terlibat dalam semua tahapan SDLC. Namun tahap ini mengacu pada tahap -satunya pengujian produk di mana cacat

produk dilaporkan, dilacak, tetap dan diuji ulang, sampai produk mencapai standar kualitas yang ditetapkan di SRS.

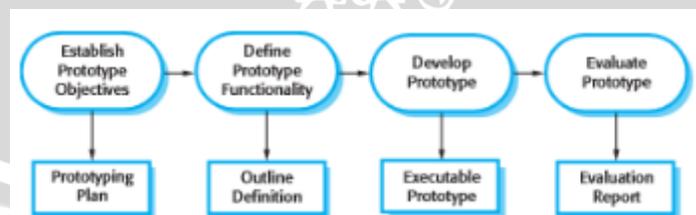
Tahap 6: Penyebaran di Pasar dan Pemeliharaan

Setelah produk diuji dan siap untuk digunakan dilepaskan secara resmi di pasar yang tepat. Kadang penyebaran produk terjadi secara bertahap sesuai strategi bisnis organisasi. Produk ini mungkin pertama kali dirilis dalam segmen terbatas dan diuji dalam lingkungan bisnis yang nyata. Kemudian berdasarkan umpan balik, produk mungkin akan dirilis seperti itu atau dengan perangkat tambahan yang disarankan dalam segmen pasar sasaran. Setelah produk ini dirilis di pasar, pemeliharaan dilakukan untuk basis pelanggan yang sudah ada.

Model SDLC

Ada pengembangan perangkat lunak berbagai model siklus hidup didefinisikan dan dirancang yang diikuti selama proses pengembangan perangkat lunak. Model ini juga disebut sebagai "Proses Pengembangan Perangkat Lunak Model". Setiap model proses mengikuti Seri langkah unik untuk jenisnya, untuk memastikan keberhasilan dalam proses pengembangan perangkat lunak. Ada banyak model yang termasuk SDLC, salah satunya adalah *Software Prototyping* yang akan digunakan dalam penelitian ini.

2.5 Prototyping



Gambar 2.8 Prototyping

Sumber : [SOI-11]

Prototipe adalah model kerja perangkat lunak dengan fungsi terbatas [SDL-13] yang tercantum dalam gambar 2.8. Prototipe tidak selalu memegang

logika yang tepat digunakan dalam aplikasi perangkat lunak yang sebenarnya dan upaya ekstra untuk dipertimbangkan di bawah estimasi usaha. *Prototyping* digunakan untuk memungkinkan pengguna mengevaluasi proposal pengembang dan mencobanya sebelum pelaksanaan. Hal ini juga membantu memahami persyaratan yang pengguna tertentu dan mungkin belum dipertimbangkan oleh pengembang selama desain produk.

2.6 CodeIgniter

CodeIgniter (CI), sebuah alat untuk membuat PHP mudah digunakan [DCI-07: 7]. CI dapat membantu untuk membuat website besar dengan struktur yang koheren. Ini akan mendisiplinkan *coding* dan membuatnya lebih kuat dengan hanya memerlukan dasar PHP tanpa perlu terlalu ahli untuk menggunakan CI. Manfaat CI di antaranya adalah:

- Efisiensi waktu

Berikut adalah contoh *coding* tanpa CI :

```
$connection = mysql_connect("localhost","fred","12345");
mysql_select_db("websites",$connection);
$result = mysql_query("select * from sites", $connection);
while($row = mysql_fetch_array($result, MYSQL_NUM)){
    foreach($row as $attribute)
        print "{$attribute[1]}";}
```

Sedangkan bila dibandingkan dengan Code Igniter, maka akan lebih efisien:

```
$this->load->database('websites');
$query = $this->db->get('sites');
    foreach($query->result() as $row){
        print $row->url;
    }
```

- Membuat web yang dibuat lebih kuat

Hal ini dimasukkan agar mencegah pengguna memasuki halaman terlarang, sehingga keamanan yang dibuat lebih kuat.

- Menjaga *Link* tetap *up to date* secara otomatis

Link yang terdapat dalam web seringkali mengalami perubahan alamat yang dituju, oleh karena itu dengan menggunakan CI alamat web akan berubah secara otomatis karena script yang dibangun saling terintegrasi.

2.7 Javascript

Harapan pengguna telah menuntun ke dalam perkembangan aplikasi sehingga menghasilkan *Ajax*, animasi, *drag and drop*, dan konsep UI lainnya yang menambah kecanggihan pencarian web [MTE-11]. Sampai beberapa tahun lalu, memenuhi hal tersebut menggunakan *JavaScript* sangatlah menyusahkan. Sehingga muncullah *JavaScript frameworks* yang dapat menjadi kerangka kode bagi pengguna yang dapat langsung dieksekusi. Hal ini membantu pengembang mengatasi hal yang paling sulit dilakukan dalam *JavaScript*, karena lingkungan browser adalah salah satu hal yang pengembang tidak dapat kontrol. Berikut penjelasan mengenai salah satu kerangka *JavaScript*.

2.7.1 MooTools

MooTools adalah kerangka *JavaScript* yang menyediakan tiga hal dasar yaitu menyediakan jalan pintas dan kelas dasar, mengembangkan *native object* untuk menambah fungsi dan juga library mengenai *JavaScript* dan *MooTools* [MTE-11]. Keunikan *MooTools* adalah gaya pemrograman yang well-rounded offering, mempunyai pendekatan filosofi sederhana untuk pengembangan selanjutnya.

Penjelasan mengenai karakteristik *MooTools* adalah sebagai berikut:

1. Tidak menduplikasi kode
2. Menambah fungsionalitas yang sesuai dengan dasar filosofi *JavaScript*
3. Mengembangkan *native objects* (*String*, *Function*, *Array*, *Element*, *Event*, dan *Number*) seperti *JavaScript*.

2.8 Enterprise Resource Planning

Menjadi organisasi yang kompetitif harus menghasilkan produk secara cepat, berkualitas dan dengan harga yang bersaing. Kumpulan dari bisnis proses yaitu *enterprise resource planning* atau ERP telah terbukti menjadi alat yang

efektif untuk mencapai tujuan tersebut. ERP adalah kumpulan alat dan proses yang mengintegrasikan departemen dan fungsi dalam perusahaan menjadi satu sistem komputer. ERP berjalan dalam database tunggal, memungkinkan bermacam – macam departemen untuk berbagi informasi dan berkomunikasi satu sama lain.

Sistem *enterprise resource planning* (ERP) adalah sistem manajemen bisnis yang meliputi kumpulan software yang terintegrasi, dapat digunakan, dapat diimplementasikan, dikelola dan diintegrasikan ke semua fungsi bisnis dalam organisasi [SSS-04].

ERP sangat dibutuhkan oleh perusahaan karena dapat memperbaiki layanan perusahaan terhadap permintaan konsumen hingga memprosesnya ke dalam *invoice* dan *revenue* yang disebut sebagai proses pemenuhan pemesanan. Itulah salah satu fungsi ERP yang bermanfaat dalam segala sisi di dalam perusahaan. Sebuah alat yang sederhana namun memberikan dampak luar biasa baik terhadap bisnis dan dunia informasi. Berikut merupakan hal yang dapat dilakukan oleh sistem ERP:

- a. Mempengaruhi hampir semua organisasi
- b. Memaksa kompetisi untuk merubah strategi dan proses
- c. Mempengaruhi rekan bisnis untuk menjadi lebih kompetitif
- d. Meningkatkan keuntungan organisasi konsultan
- e. Menjadi alat paling penting dalam bisnis proses *engineering*
- f. Memaksa mengimplementasikan bisnis proses dalam suatu organisasi
- g. Dapat memanfaatkan secara keseluruhan potensi dari *client – server computing* untuk menghasilkan produk *enterprise*.
- h. Merubah kebiasaan dari fungsi sistem informasi dan IT professional
- i. Merubah kebiasaan pekerjaan dalam seluruh area fungsional bisnis
- j. Implementasi ERP sangat mahal

ERP dapat diterapkan dalam segala bidang pekerjaan dan usaha, karena dapat disesuaikan modul sesuai kebutuhan. Seperti dalam bidang bisnis, usaha, maupun pendidikan.

2.8.1 ERP Education

Enterprise Resource Planning Education (ERP) adalah sistem perangkat lunak sistem yang mengintegrasikan data kelembagaan dan proses melalui satu [AET-08].

Yang dilakukan ERP untuk lembaga pendidikan antara lain [AET-08]:

1. Mengintegrasikan informasi di semua fungsi (contoh termasuk pendaftaran, bantuan keuangan, sumber daya manusia).
2. Memperlancar arus informasi di antara fungsi lembaga.
3. Melacak berbagai kegiatan-kegiatan kelembagaan secara terpadu, dan memfasilitasi perencanaan kegiatan di masa depan berdasarkan peristiwa sebelumnya.
4. Mendukung analisis kecenderungan dan dengan demikian meningkatkan kinerja lembaga.
5. Memungkinkan pengguna untuk:
 - a. Input data ke dalam satu sistem untuk memungkinkan untuk diproses dengan data lain.
 - b. Akses data sebagai laporan informasi dalam lingkungan *real-time*.
 - c. Berbagi data dan praktek di seluruh institusi
 - d. Mengubah Praktek bisnis

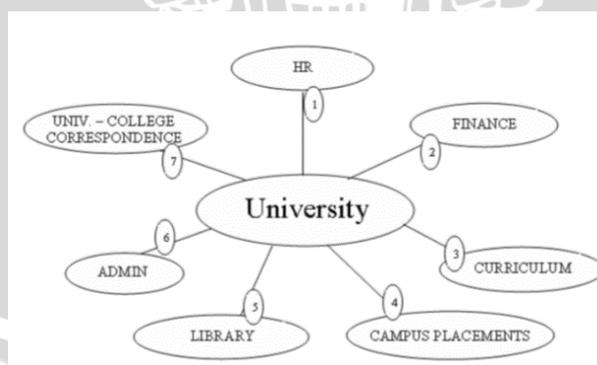
Modul utama dari sistem ERP, dalam lingkungan bisnis, adalah sumber daya manusia, administrasi, keuangan, perencanaan produksi, penjualan dan pemasaran, inventarisasi dan strategi manajemen. Modul ini saling terkait dan ketika terintegrasi, memberikan informasi yang diperlukan digunakan secara luas untuk proses pengambilan keputusan.



Gambar 2.9 Modul ERP

Sumber : [AGS-09]

Gambar 2.9 menunjukkan modul yang terintegrasi dalam sistem ERP. Mengingat universitas sebagai sistem ERP, modul utama dapat dibandingkan dengan fungsi umum seperti yang diamati dalam lingkungan universitas. Gambar 2.10 menunjukkan kegiatan masing-masing berubah sebagai modul di setup ERP. Dalam sebuah organisasi bisnis dimana sistem ERP diimplementasikan, penerima manfaat (beneficiary) adalah 'pelanggan', sedangkan pada universitas, penerima manfaat adalah 'mahasiswa'. Dengan penyesuaian seperti ini, perhatian utama untuk karir masa depan siswa akan dibentuk dengan baik. Dalam perjalanan waktu, perangkat tambahan lebih lanjut dan fitur nilai tambah dapat dimasukkan ke dalam sebuah sistem Perencanaan Sumber Daya Perusahaan berbasis web.



Gambar 2.10 Modul ERP *University*

Sumber : [AGS-09]

Berikut merupakan perbandingan modul dari ERP perusahaan dengan ERP Universitas seperti yang terdapat dalam tabel 2.2.

Table 2.2: Perbandingan Modul

<i>Subject of Comparison</i>	<i>ERP</i>	<i>Univ./Institution</i>
<i>Beneficiary</i>	<i>Customers</i>	<i>Students & Staff</i>
<i>Workers</i>	<i>Employee</i>	<i>Staff & Administration</i>
<i>Modules</i>	<i>HR</i>	<i>Administration</i>
	<i>Sales & Marketing</i>	<i>University Correspondence</i>
	<i>Inventory</i>	<i>Library</i>
	<i>Costing</i>	<i>Academic Schedule</i>
	<i>Manufacturing</i>	<i>Curriculum</i>
	<i>Sales & Distribution</i>	<i>On-Campus & Off- Campus</i>
	<i>Finance</i>	<i>Finance</i>

Sumber : [AGS-09]

Dari perbandingan modul tersebut, dapat disimpulkan bahwa ERP dapat diterapkan dalam universitas yang notabene merupakan lembaga pendidikan.

2.9 Pengujian Perangkat Lunak

2.9.1 Usability Testing

Proses pembelajaran mengenai pengguna dengan mengobservasi mereka menggunakan produk untuk mencapai tujuan tertentu yang membuat mereka tertarik [JDU-08]. Usability testing dapat meliputi:

1. Mudah dipelajari
2. Mudah Digunakan
3. Menyenangkan
4. Intuitif

Pengujian ini berfokus dalam mengobservasi pengguna dengan menguji menggunakan tugas yang sebenarnya dan dapat dilakukan dengan dua cara yaitu:

1. Pengujian formatif : dilakukan pada saat program sedang dibuat.
2. Pengujian summatif: dilakukan pada saat program selesai dibuat.

2.9.2 *Functional Testing*

Pengujian fungsional memastikan bahwa fungsi individu bekerja dengan baik. Pengujian bagaimana fungsi aplikasi, atau, dengan kata lain, kaitannya dengan pengguna dan terutama ke seluruh sistem. Secara tradisional, pengujian fungsional dilaksanakan oleh tim penguji, independen dari pengembang [AGS-09].

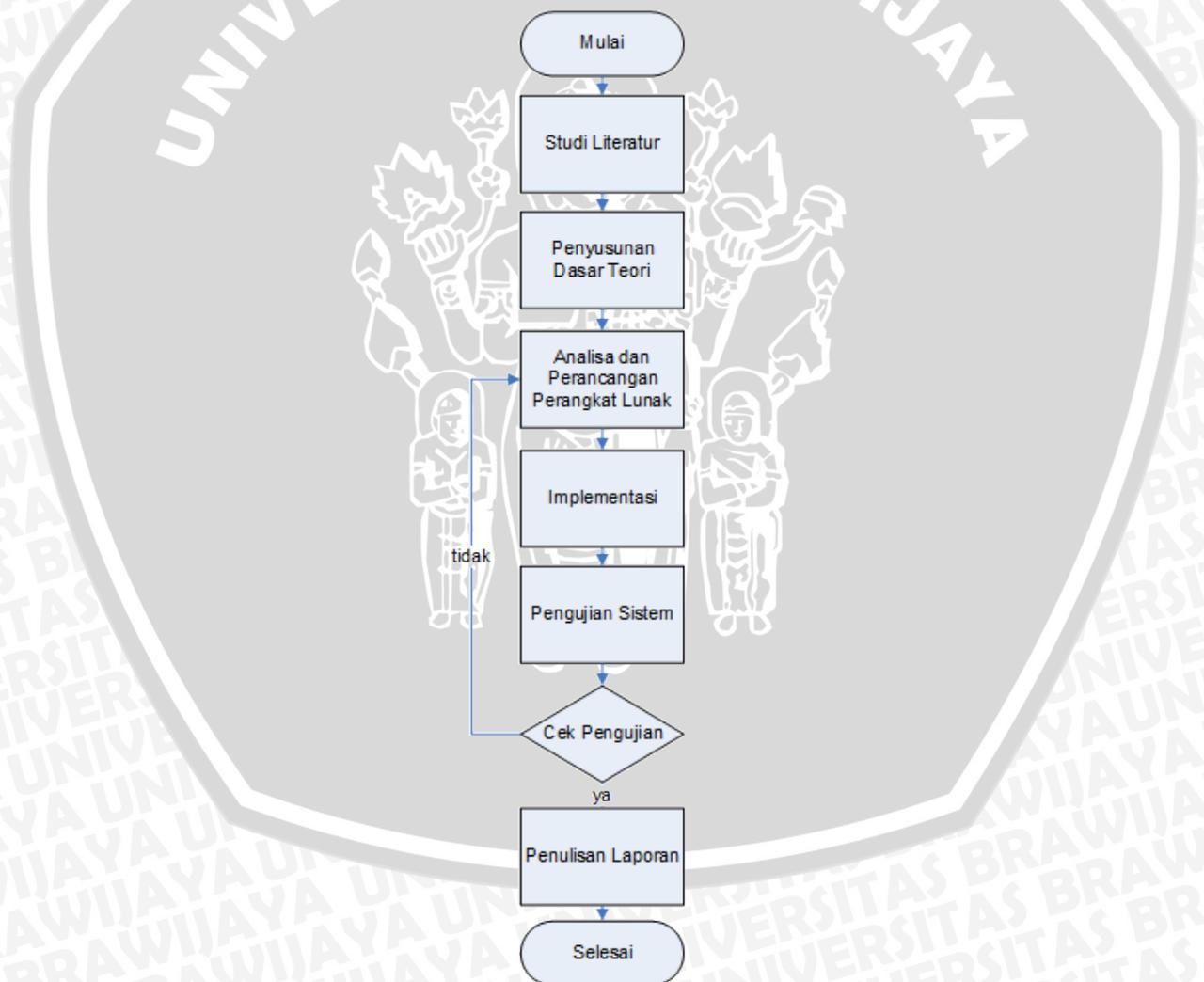
Sebuah tes fungsional tidak peduli dengan rincian internal sebuah aplikasi. Cara lain untuk menyatakan ini adalah bahwa pengujian fungsional adalah "tes pelanggan" [FTT-13]. Pengembang perlu patokan selama semua tahap pembangunan untuk memberitahu mereka apa yang mereka miliki dan belum dicapai. Pengujian fungsional dimulai segera setelah ada fungsi untuk menguji dan terus melalui penyelesaian aplikasi dan kontak pelanggan pertama.

Beberapa orang berpikir pengujian fungsional sebagai meniru pengguna dan memeriksa output yang diharapkan. Tapi pelanggan nyata tidak hanya memberi perintah sederhana ke dalam aplikasi. Mereka menjalankan aplikasi pada sistem, bersamaan dengan aplikasi lain, dengan fluktuasi konstan dalam beban pengguna.

Aplikasi harus tahan benturan dalam menghadapi kondisi ini. Semakin banyak, user interface aplikasi kita diberikan oleh komponen pra-diuji, yang menghasilkan beberapa kejutan setelah mereka terintegrasi dengan benar. Di sisi lain, kita terus-menerus diminta untuk menulis aplikasi kustom untuk sistem yang terus meningkat kompleksitasnya. Oleh karena itu, fungsi utama diuji melalui uji fungsional yang berfokus kepada sistem yang terkait daripada pengguna terkait.

BAB III METODE PENELITIAN

Bagian ini menjelaskan langkah – langkah yang akan ditempuh dalam penyusunan skripsi, yaitu studi literatur dan penyusunan dasar teori, analisis dan perancangan perangkat lunak, implementasi, pengujian dan evaluasi sistem, serta pengambilan kesimpulan dan saran. Untuk lebih memudahkan dalam pengerjaan aplikasi maka dibuat sebuah alur kerja yang dapat dipresentasikan oleh *flowchart* berikut ini:



Gambar 3.1. Diagram Blok Metode Penelitian
Sumber : Perancangan

3.1 Studi Literatur dan Penyusunan Dasar Teori

Studi literatur menjelaskan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori – teori pendukung tersebut meliputi :

1. Kajian Pustaka
2. Rekayasa Perangkat Lunak
3. *Unified Modelling Language*
 - a. *Use Case*
 - b. *Activity Diagram*
 - c. *Sequence Diagram*
4. *Software Development Life Cycle*
5. *Prototyping*
6. *Code Igniter*
7. *JavaScript*
 - a. *MooTools*
8. *Enterprise Resource Planning*
 - a. *ERP Education*
9. Pengujian Perangkat Lunak
 - a. *Usability Testing*
 - b. *Functional Testing*

3.2 Analisis dan Perancangan Perangkat Lunak

3.2.1 Analisis Kebutuhan

Analisis kebutuhan dilakukan dengan menentukan kebutuhan apa saja yang dibutuhkan untuk menganalisis bisnis proses SMKN 2 Buduran dan merancang bentuk *ERP Education*. Proses analisis ini meliputi :

1. Pemodelan use case untuk proses bisnis (*as-is*) dan proses sistem(*to-be*)
2. Pemodelan aktifitas bisnis dan aktifitas sistem

Metode analisis yang digunakan adalah *Object Oriented analysis* dengan menggunakan bahasa pemodelan UML (Unified Modelling Language). Kebutuhan yang digunakan dalam pembuatan aplikasi ini

meliputi kebutuhan fungsional seperti dalam table 3.1, kebutuhan non-fungsional seperti dalam table 3.2 dan kebutuhan lingkungan seperti dalam table 3.3.

Tabel 3.1 Kebutuhan fungsional

Kebutuhan Fungsional	
Kurikulum	<i>Create, Read, Update</i> dan <i>Delete</i> data mata pelajaran, tahun kurikulum, kelas, jurusan dan nilai siswa.
Kesiswaan	<i>Create, Read, Update</i> dan <i>Delete</i> data siswa, orang tua, ekstrakurikuler dan osis.
Humas	<i>Create, Read, Update</i> dan <i>Delete</i> data DUDI, status prakerin dan prakerin.
Sarana	<i>Create, Read, Update</i> dan <i>Delete</i> data barang, peminjaman
Kepegawaian	<i>Create, Read, Update</i> dan <i>Delete</i> data pegawai, jenis pegawai

Sumber : Perancangan

Tabel 3.2 Kebutuhan non-fungsional

Kebutuhan Non Fungsional [BWH-07]	
<i>Information</i>	Tampilan form dalam aplikasi mudah dimengerti dan nyaman digunakan oleh pengguna.
<i>Safety</i>	Keselamatan data harus terjamin pada sistem ini dengan selalu memberikan konfirmasi saat pengguna ingin melakukan operasi seperti edit dan hapus.
<i>Security</i>	Sistem informasi ini harus dapat mem- <i>filter user</i> yang berhak untuk mengakses sistem atau tidak. Oleh karena itu dibuat halaman <i>login</i> agar tidak sembarang orang dapat masuk menjadi admin ke dalam sistem. Yang dapat masuk ke dalam sistem hanyalah admin bukan masyarakat umum.

Sumber : Perancangan

Tabel 3.3 Kebutuhan lingkungan

Kebutuhan Lingkungan	
Hardware	<ul style="list-style-type: none"> - Personal Computer minimal processor Dual Core - RAM minimal 1GB
Software	<ul style="list-style-type: none"> - Microsoft Windows 8 sebagai sistem operasi. - Adobe Dreamweaver CS5 sebagai pendukung pembuatan website. - AstahUML sebagai pembuat model perancangan aplikasi. - Microsoft Visio sebagai pembuat model perancangan aplikasi. - XAMPP sebagai alat pendukung basis data. - Code Igniter sebagai framework pembuatan web. - MooTools sebagai framework JavaScript.

Sumber : Perancangan

3.3 Perancangan Perangkat Lunak

Setelah menentukan jenis kebutuhan yang diperlukan tahap selanjutnya adalah desain atau perancangan sistem. Perancangan terdiri dari beberapa tahap yang sesuai dengan model perangkat lunak *Prototyping*, antara lain [SDL-13]:

1. **Basic Requirement Identification.** Langkah ini melibatkan pemahaman persyaratan produk yang sangat dasar terutama dalam hal *user interface*. Kerumitan internal desain dan aspek eksternal seperti kinerja dan keamanan dapat diabaikan pada tahap ini.
2. **Developing the initial Prototype.** Prototipe awal dikembangkan di tahap ini, di mana persyaratan yang sangat dasar yang ditunjukkan dan *user interface* disediakan. Fitur-fitur ini mungkin tidak benar-benar bekerja dengan cara yang sama seperti dalam perangkat lunak sebenarnya sehingga *workarounds* digunakan untuk memberikan tampilan yang sama dan merasa untuk pelanggan di prototipe yang dikembangkan.
3. **Review of the Prototype.** Prototipe yang dikembangkan kemudian disajikan kepada pelanggan dan *stakeholder* penting lainnya dalam proyek.

Umpan balik yang dikumpulkan secara teratur dan digunakan untuk peningkatan lebih lanjut produk dalam pengembangan.

4. **Revise and enhance the Prototype.** Umpan balik dan *review* komentar dibahas selama tahap ini dan beberapa negosiasi terjadi dengan pelanggan berdasarkan faktor-faktor seperti, waktu dan keterbatasan anggaran dan kelayakan teknis pelaksanaannya. Perubahan diterima ini lagi tergabung dalam. Prototipe baru yang dikembangkan dan siklus berulang sampai harapan pelanggan terpenuhi.

3.2.1 Implementasi

Implementasi aplikasi dilakukan dengan mengacu pada perancangan perangkat lunak yang sudah didesain pada sub bab analisis kebutuhan dan perancangan perangkat lunak. Implementasi perangkat lunak dilakukan dengan menggunakan bahasa pemrograman PHP, *Code Igniter* untuk *framework* dan *MySQL-Xampp* untuk basis data.

3.2.2 Pengujian dan Evaluasi Sistem

Pengujian perangkat lunak dilakukan untuk mengetahui apakah kinerja sistem perangkat lunak *Enterprise Resource Planning Education* telah memenuhi spesifikasi kebutuhan yang melandasi. Pengujian dilakukan dengan dua cara, yaitu:

1. *Functional Testing*

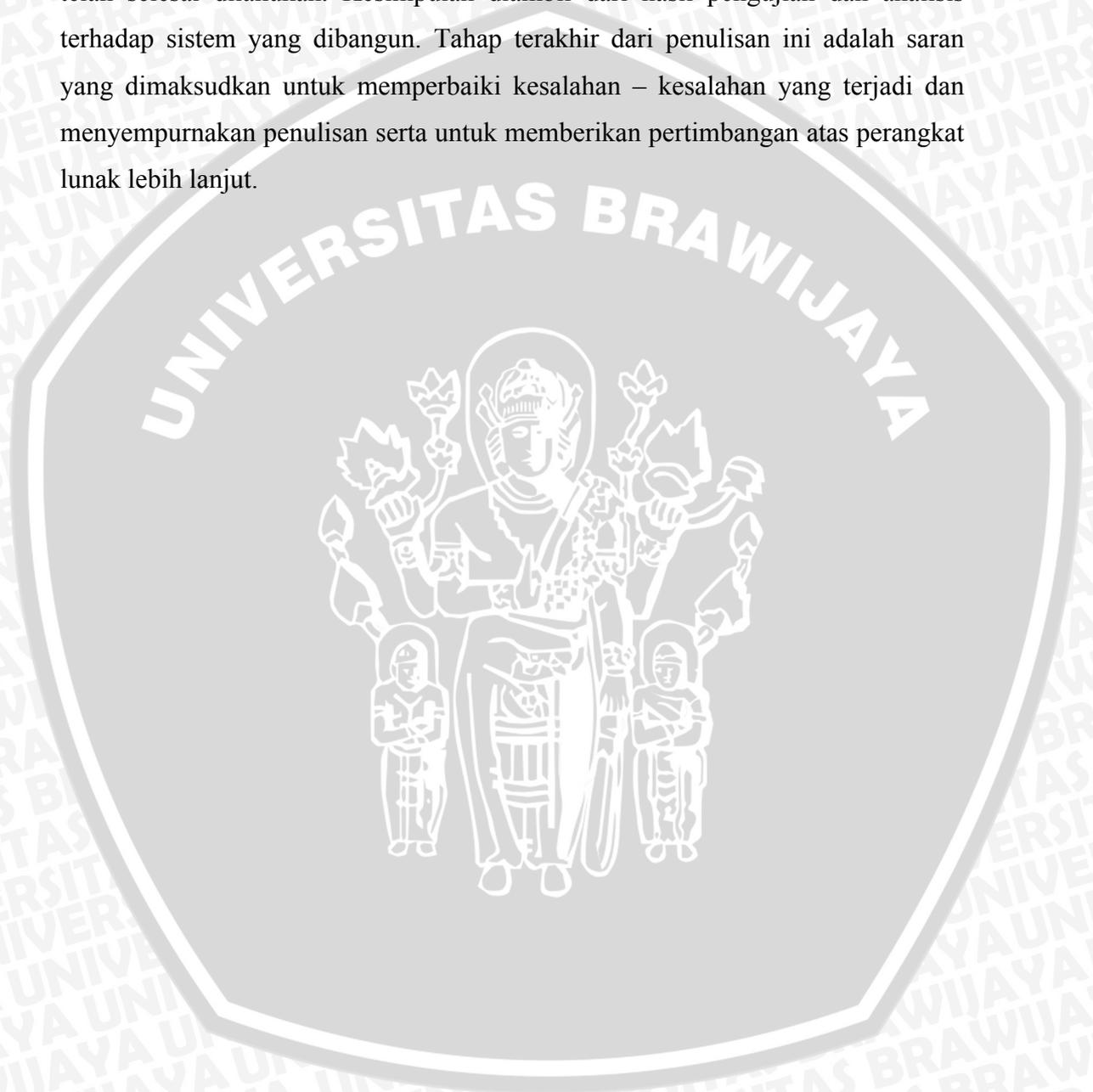
Dalam aplikasi web, pengujian fungsional dapat berkisar dari pengujian apakah link bekerja untuk memeriksa apakah perubahan yang dilakukan oleh pengguna di halaman web tercermin dalam database.

2. *Non-functional Testing*

Pengujian ini berfokus kepada *information*, *safety* dan *security*. Dimana dalam menguji aspek *information*, penulis menggunakan *Usability Testing* dimana pengujian ini memastikan bahwa aplikasi nyaman digunakan oleh pengguna.

3.3 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan perangkat lunak, implementasi perangkat lunak, dan pengujian perangkat lunak telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap terakhir dari penulisan ini adalah saran yang dimaksudkan untuk memperbaiki kesalahan – kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas perangkat lunak lebih lanjut.



BAB IV

PERANCANGAN DAN IMPLEMENTASI

Perancangan perangkat lunak digunakan sebagai tahap awal pengimplementasian desain sistem dan perancangan antarmuka yang digunakan. Perancangan perangkat lunak dibangun berdasarkan hasil studi literature dan analisis kebutuhan sistem. Tahap perancangan sistem meliputi pemodelan proses bisnis (*as-is*), pemodelan sistem (*to-be*), rekapitulasi kebutuhan sistem dan perancangan.

4.1. Analisis Kebutuhan

Pada tahap ini proses analisis kebutuhan dan perancangan sistem dibagi menjadi 2 yakni, pemodelan untuk proses bisnis (*as-is*) dan pemodelan proses sistem (*to-be*).

4.1.1. Pemodelan Proses Bisnis (*as-is*)

Proses bisnis *as – is* merupakan proses bisnis awal yang diterapkan dalam SMKN 2 Buduran. Proses ini belum melibatkan keterkaitan aplikasi perangkat lunak dalam kegiatannya. Oleh karena itu, pada bagian ini akan dibahas mengenai proses bisnis yang dilakukan setiap *actor* / objek pelaku dari divisi dalam di SMKN 2 Buduran sebelum menggunakan aplikasi *Enterprise Resource Planning Education* dari setiap divisi yang akan dibahas. Berikut merupakan keseluruhan alur proses bisnis yang digambarkan dalam *flowchart* gambar 4.1. *Flowchart* tersebut masih tergambar secara umum mengenai bagian keseluruhan sekolah, sehingga untuk mengetahui tugas setiap bidang dalam organisasi sekolah, dibutuhkan *Use Case Diagram* yang mengacu kepada struktur organisasi sekolah dalam gambar 4.2.