

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tinjauan Studi**

Tinjauan studi berguna bagi peneliti untuk dijadikan pedoman dan pegangan penelitian yang selanjutnya akan dibuat, dengan adanya penelitian sebelumnya tersebut memudahkan peneliti sesuai topik pembahasan. Penelitian sebelumnya juga dapat dijadikan perbandingan penelitian sehingga menghasilkan penelitian baru yang lebih bermanfaat. Dalam melaksanakan penelitian ini, diambil beberapa referensi sebagai dasar pelaksanaan penelitian tentang Sistem Informasi Manajemen (SIM) yang diterapkan di organisasi mahasiswa Unisnu Jepara diantaranya :

Menurut Afifah, Sudjalwo dan Fatmawati (2014) yang berjudul Sistem Informasi Keuangan Kemahasiswaan Universitas Muhammadiyah Surakarta. Dalam jurnal tersebut membahas tentang pengelolaan dan rekap keuangan kegiatan kemahasiswaan di Universitas Muhammadiyah Surakarta yang dikelola secara manual dan cenderung masih merepotkan bagi sebagian besar pengurus Organisasi Mahasiswa (ORMAWA). Informasi mengenai dana bagi kemahasiswaan yang harus diakses langsung ke ruangan Bagmawa dinilai menjadi hal yang merepotkan bagi kedua belah pihak. Pada penelitian tersebut menghasilkan sistem informasi berbasis *WebApp* yang lebih fleksibel yaitu sistem informasi keuangan yang memudahkan ORMAWA dalam memperoleh informasi terkait pendanaan mahasiswa termasuk, sisa dana, status masing-masing dana, penyerapan dana, tanggal dan jumlah dana masing-masing yang diperoleh ORMAWA. Metode yang digunakan dalam penelitian ini merupakan *System Development Life Circle* (SDLC). Dengan adanya penelitian tersebut ORMAWA tidak repot dalam mengakses dana kemahasiswaan pada bagmawa yang masih manual dan cenderung menyulitkan sebagian pengurus ORMAWA. Sistem telah diuji dengan pemberian kuisioner penilaian kepada responden dengan rata-rata persentase 82 % validasi dari hasil pengujian [4].

Menurut Hidayat, Listyorini dan Khotimah (2015) yang berjudul Aplikasi Manajemen Unit Kegiatan Mahasiswa pada Universitas Muria Kudus. Dalam jurnal

tersebut membahas tentang kurangnya manajemen pada kegiatan-kegiatan yang dilakukan Unit Kegiatan Mahasiswa (UKM) yang mengakibatkan banyak data yang hilang serta laporan-laporan yang tidak tertata secara rapi sehingga pada pelaporan pertanggungjawaban sering kesusahan dalam mencari data kegiatan yang sudah dilaksanakan. Pada penelitian ini menghasilkan suatu aplikasi manajemen unit kegiatan mahasiswa yang berguna dalam pengolahan data dan pelaporan suatu kegiatan, pengurus, rapat, sponsor, proposal, surat, dokumentasi, inventaris dan keuangan pada unit kegiatan mahasiswa menggunakan metode *waterfall* dengan perancangan berupa *Data Flow Diagram* (DFD) dan *Entity Relationship Diagram* (ERD) dengan Bahasa Pemrograman PHP serta Mysql. Dengan adanya penelitian ini memudahkan Unit Kegiatan Mahasiswa (UKM) untuk mengolah data secara lebih rapi [5].

Menurut Rosyid, Pradana dan Rusdianto (2018) yang berjudul Pembangunan Sistem Informasi Pengajuan Kegiatan dan Pemberian Point bagi Mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya. Dalam jurnal tersebut membahas tentang permasalahan yang terjadi pada pihak kemahasiswaan Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya yaitu pada adanya kesalahan dalam penulisan proposal, tidak mengetahui prosedur pengajuan, berkas yang kurang lengkap dan banyaknya yang menanyakan status proposal yang diajukan. Permasalahan mahasiswa pada alur pengajuan, yang terlalu panjang mengakibatkan menambahnya waktu pengajuan kegiatan. Pada penelitian ini merupakan implementasi pembangunan sistem informasi pengajuan kegiatan dan pemberian poin bagi mahasiswa Fakultas Ilmu Komputer menggunakan *laravel framework*. Dalam perancangan sistem menggunakan *Object Oriented Analysis* (OOA) dan *Unified Modelling Language* (UML). Sistem informasi telah diuji menggunakan diuji menggunakan metode *white box testing* pada pengujian unit dengan hasil kompleksitas logika rata-rata adalah 3. Sedangkan, pada *black box testing* dengan pengujian validasi diperoleh 100% valid untuk seluruh kasus uji. Dengan adanya sistem tersebut dapat membantu dan memudahkan dalam pengajuan kegiatan mahasiswa FIKOM [6].

Dari penelitian tersebut, peneliti ingin merencanakan perancangan dan penerapan sistem informasi manajemen berbasis web yang dapat memudahkan

dalam pengelolaan anggaran organisasi mahasiswa guna mendukung penentuan alokasi anggaran organisasi mahasiswa di Unisnu Jepara. Dengan demikian yang membedakan dalam penelitian ini adalah objek penelitian pada Unisnu Jepara dengan permodelan *Unified Modelling Language* (UML) dalam memudahkan penggambaran sistem yang dirancang dan metode pengembangan sistem menggunakan model prototipe yang menghasilkan produk yang lebih akurat.

## **2.2 Tinjauan Pustaka**

### **2.2.1 Sistem Informasi Manajemen**

Sistem Informasi Manajemen (SIM) merupakan sistem yang dapat mempertemukan suatu kebutuhan pada pengolahan transaksi yang diciptakan untuk melaksanakan pengolahan data, menyajikan data, mendukung fungsi operasi dan manajemen yang menunjang tugas rutin, evaluasi terhadap prestasi serta pengambilan sebuah keputusan yang diperlukan oleh suatu organisasi tertentu [7] [8] [9] [10]. Komponen fisik sistem informasi manajemen yang melengkapi suatu sistem pengoprasianya akan terdiri atas *Software* (perangkat lunak), *hardware* (perangkat keras), *database* (basis data), prosedur dan juga personil [9]. Berikut penjelasan komponen fisik Sistem Informasi Manajemen (SIM), yaitu:

1. *Software* (Perangkat Lunak)

Perangkat lunak yang digunakan dapat dibagi menjadi 3 bagian, yaitu sistem perangkat lunak, aplikasi perangkat lunak umum dan aplikasi perangkat lunak yang termasuk program yang lebih spesifik. Sistem perangkat lunak merupakan suatu sistem pengoprasian dan juga sistem manajemen data yang dapat memungkinkan dalam pengoprasian sistem komputer.

2. *Hardware* (Perangkat Keras)

Perangkat lunak berupa komputer sebagai pusat masukan atau keluaran, unit penyimpanan file dan lainnya, termasuk juga alat penyimpanan data dan terminal masukan atau keluaran.

3. *Database* (Basis Data)

Basis data merupakan suatu file yang berisi sebuah program dan data yang dibuktikan dengan adanya media penyimpanan secara fisik seperti harddisk dan penyimpanan lainnya, file juga merupakan keluaran yang tercetak, berupa catatan diatas kertas, macro film dan sebagainya.

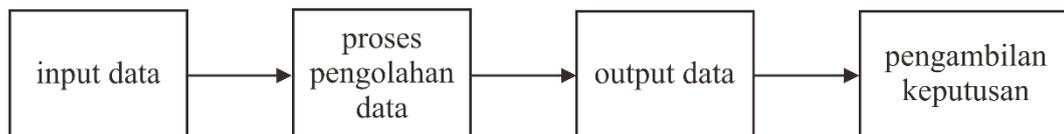
#### 4. Prosedur

Prosedur merupakan suatu komponen fisik karena prosedur yang disediakan dalam bentuk fisik seperti berupa buku panduan dan sebuah intruksi. Jenis prosedur yang dibutuhkan meliputi sebuah instruksi untuk pemakai, instruksi untuk suatu masukkan dan intruksi dalam pengoprasian pusat komputer.

#### 5. Personil

Personil merupakan sebuah operator komputer, seorang analis sistem, programmer atau personil data entry dan seorang manajer sistem informasi.

Sistem informasi manajemen harus terus dijaga untuk menjamin sistem tetap dapat menyiapkan informasi yang dibutuhkan oleh *user* atau penggunanya. Dalam pengembangan sistem, sistem yang baik adalah sistem yang menghasilkan data menjadi informasi dan menjadikan data sebagai fungsi yang berguna bagi penggunanya. Proses kerja dari SIM adalah *input* (masukkan) berupa data yang dibutuhkan dalam suatu organisasi, dimasukkan dalam sebuah sistem informasi yang tersimpan dalam *database*, data akan diproses dan diolah sesuai dengan kebutuhan sistem. Setelah pengolahan data akan menghasilkan sebuah informasi guna mendukung suatu keputusan dalam manajerial berdasarkan kondisi informasi yang ditampilkan.



Gambar 2.1 Proses Kerja SIM

Sumber : Buku Sistem Informasi Manajemen, 2015

## 2.2.2 Codeigniter Framework

### 2.2.2.1 Pengertian Codeigniter Framework

*Framework* merupakan sebuah kumpulan kode atau potongan program yang berupa suatu *library* (pustaka) dan *tool* (alat) yang dipadukan sedemikian rupa sehingga menjadi suatu kerangka kerja yang utuh guna mempermudah dan mempercepat dalam proses pengembangan aplikasi [11] [12]. *Framework* memiliki banyak jenisnya seperti Codeigniter, Yii, Zend, Laravel, Simpony dan lainnya. Keuntungan menggunakan *framework*, selain waktu yang lebih singkat dalam perancangan, *framework* juga mudah dalam pembacaanya sehingga baik dilakukan secara *teamwork* dalam memudahkan perancangan dan pengembangan aplikasi.

Pada penelitian ini, peneliti menggunakan *Codeigniter Framework* dalam mendukung perancangan sistemnya. Karena *codeigniter* merupakan suatu *web framework* untuk bahasa PHP yang membantu memudahkan dan mempercepat pengembang aplikasi web dalam pembuatannya dengan banyak fitur yang memungkinkan pengembang aplikasi web lebih sistematis, cepat, mudah dan lebih fleksibel [11] [12] [13]. *Codeigniter* juga menggunakan metode *Model, View* dan *Controller* (MVC) dalam perancangannya serta *Open Source* (tidak berbayar).

Selain memudahkan dalam perancangan, *codeigniter* juga memiliki kelebihan lain [11] [13], diantaranya :

- a) *Syntax* yang terstruktur dengan rapi memberikan kemudahan dalam penggunaannya.
- b) Ringan pada semua *platform* sistem informasi.
- c) Dokumentasi dan *library* yang lengkap serta berbasis PHP5.
- d) Tidak banyak konfigurasi.
- e) Kinerja lebih bagus dengan jejak yang lebih sedikit.
- f) Kompatibilitas dengan hosting.

#### 2.2.2.2 Metode MVC (Model, View dan Controllers)

Metode *Model, View* dan *Controllers* (MVC) digunakan sebagai pola dasar yang memisahkan antara tiga bagian yang membentuk bagian yang terpisah. Dan mempunyai fungsi masing-masing, diantaranya adalah sebagai berikut :

- a) *Model*

*Model* merupakan representasi dari proses bisnis didalam setiap perangkat lunak. *Model* merupakan suatu bagian yang bertugas dalam mengolah data mentah menjadi data yang mengandung arti yang dibutuhkan oleh pengguna. *Model* biasanya berhubungan dengan sebuah *database* untuk memanipulasi suatu data, untuk menangani validasi dari bagian *controller*, tetapi tidak dapat berhubungan secara langsung dengan *view*. *Model* biasanya mendukung *controller* dalam memecah fungsi dalam pemrosesan aplikasi. *Model* berisi *class* dalam *controller* yang didefinisikan *class* pada objek atau penambahan suatu fungsi.

- b) *View*

*View* merupakan suatu bagian yang menangani suatu logika representasi. Dalam pengembangan aplikasi web bagian ini biasanya berisi sebuah file *template*

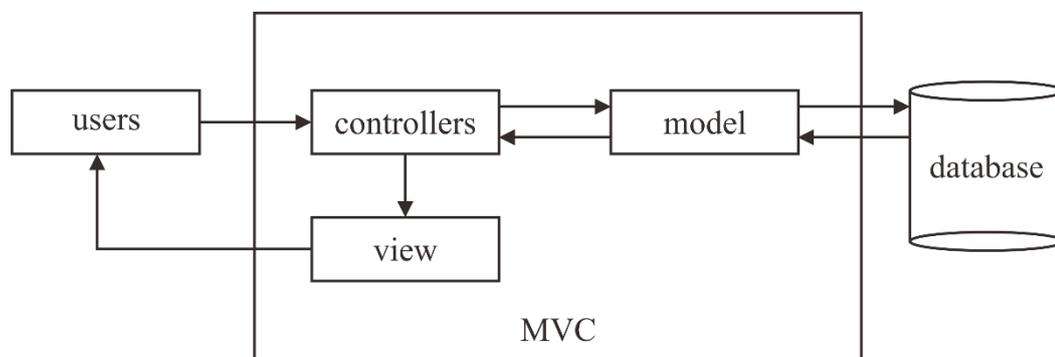
HTML yang diatur pada bagian *controller*. *View* juga berfungsi untuk menerima dan merepresentasikan suatu data kepada *users*. Pada bagian ini tidak memiliki akses langsung terhadap bagian *model*. Biasanya berupa antarmuka (*interface*) yang dimunculkan. *View* menampilkan data yang disiapkan oleh *controller* dan *model*.

c) *Controllers*

*Controller* merupakan bagian yang bertugas untuk mengatur hubungan antara bagian *model* dan bagian *view*, *controller* berfungsi dalam menerima sebuah *request* (permintaan) yang diperoleh dari *users* kemudian diproses oleh suatu aplikasi. *Controller* juga merupakan suatu komponen yang digunakan untuk bekerja bersama dengan *model* dan mengatur *view* mana yang akan digunakan.

### 2.2.2.3 Pola Desain MVC

Dalam implementasi *model*, *view* dan *controller* pada *codeigniter framework*, pola desain dalam proses kerja *Model*, *View* dan *Codeigniter* (MVC) dapat dilihat sebagai berikut :



Gambar 2.2 Pola Model MVC

Sumber : Buku Framework Codeigniter, 2012

Pada gambar diatas, *controller* mengambil masukan dari *users* dan diubah menjadi sebuah perintah, kemudian *model* mempresentasikan struktur data dalam *database*. Hasil perintah *controller* pada *model* dalam pengambilan data dalam *database* dikembalikan pada *controller* yang selanjutnya akan dikirim kepada *view* untuk mempresentasikan keluaran dari *model* melalui *controller* dan ditampilkan kepada *users* sesuai perintah.

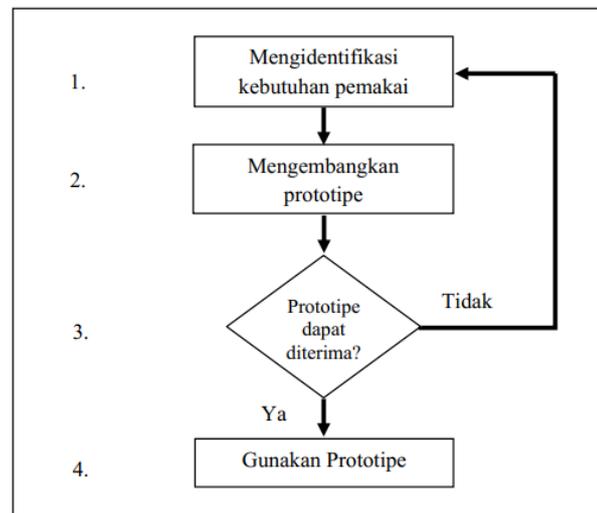
### 2.2.3 Model Prototipe

Prototipe merupakan suatu model pengembangan sistem yang memiliki durasi perancangan yang cukup cepat dan memungkinkan bagi perancang maupun

pemakai dalam mengetahui secara lengkap cara ataupun fungsi-fungsi pada sistem yang telah dirancang, idealnya prototipe digunakan dalam mekanisme pendefinisian spesifikasi-spesifikasi kebutuhan perangkat lunak [14] [15]. Prototipe yang sudah dibuat bersama dengan pengguna belum bisa dikatakan sebagai aplikasi jadi karena prototipe bersifat evolusioner yang dapat dikembangkan sesuai dengan spesifikasi kebutuhan pengguna.

Dalam pembuatan prototipe, dilakukan komunikasi dengan pengguna untuk mendefinisikan spesifikasi kebutuhan, penggambaran area sistem yang dibangun, perancangan antarmuka (*interface*) atau format tampilan. Perancangan secara cepat dengan konstruksi pembuatan prototipe sehingga akan dievaluasi untuk memberikan umpan balik dan memperhalus spesifikasi kebutuhan sebagai evaluasi pengembangan sistem. Dalam pengembangannya, prototipe terdiri dari dua jenis yaitu jenis I (*evolutionary prototype*) dan jenis II (*requirement prototype*) [14]. Tahapan yang terdapat pada prototipe jenis I, langkah-langkahnya adalah sebagai berikut :

1. Mengidentifikasi kebutuhan pemakai. Pada tahap ini peneliti observasi guna menganalisa spesifikasi kebutuhan sistem, menganalisa proses yang telah berjalan, kebutuhan alat dan bahan yang digunakan, kebutuhan fungsional maupun kebutuhan non-fungsional dalam perancangan sistem.
2. Mengembangkan prototipe. Pada tahap ini perancangan menggunakan peralatan penunjang *prototyping*.
3. Persetujuan penggunaan prototipe. Pengembang mendidik pengguna untuk penggunaan sebuah prototipe dan memberikan kesempatan kepada pengguna untuk membiasakan diri dengan sistem yang dibuat. Pengguna memberikan masukan bagi pengembang apakah prototipe memuaskan atau tidak. Jika ya, langkah berikutnya akan diambil dan jika tidak, prototipe akan direvisi atau diperbaiki dengan mengulangi langkah 1 sampai 3 dengan penggunaan yang lebih baik mengenai suatu kebutuhan pengguna sistem.
4. Menggunakan prototipe. Sistem yang ditawarkan kepada pengguna selanjutnya digunakan sebagai sistem operasional dengan kesesuaian spesifikasi kebutuhan yang telah ditentukan pada pertemuan pengembang dengan pengguna.



Gambar 2.3 Prototipe Jenis I

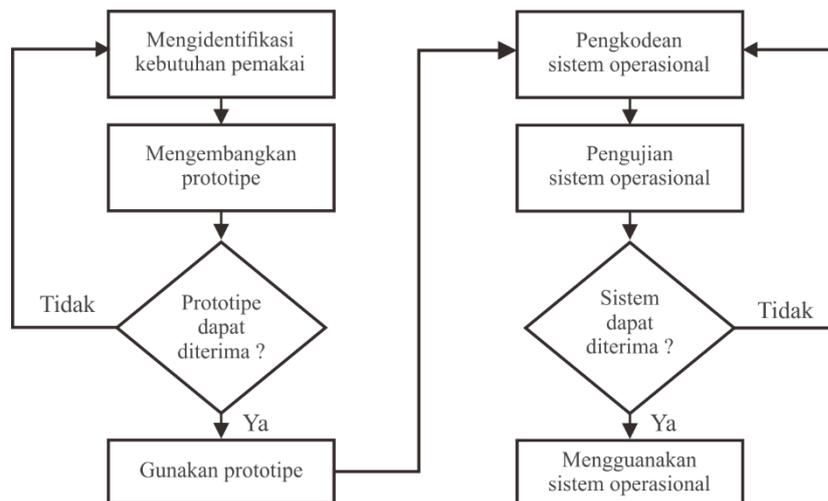
Sumber : Buku Sistem Informasi Manajemen, 2007

Respon pertama pengguna, diawali dengan menampilkan prototipe sistem informasi dan melihat adanya kesesuaian fitur-fitur yang bekerja pada sistem sudah sesuai dengan spesifikasi kebutuhan. Respon tersebut dikumpulkan dalam bentuk lembar observasi, wawancara atau kuesioner.

Pengembangan prototipe jenis II, untuk tiga langkah pertama sama dengan prototipe jenis I, adapun langkah-langkahnya sebagai berikut :

1. Mengidentifikasi kebutuhan pemakai. Pada tahap ini peneliti observasi guna menganalisa spesifikasi kebutuhan sistem yang diinginkan, menganalisa alat dan bahan yang digunakan, kebutuhan fungsional maupun kebutuhan non-fungsional dalam perancangan sistem.
2. Mengembangkan prototipe. Pada tahap ini perancangan menggunakan peralatan penunjang *prototyping*.
3. Persetujuan penggunaan prototipe. Pengembang mendidik pengguna untuk penggunaan sebuah prototipe dan memberikan kesempatan kepada pengguna untuk membiasakan diri dengan sistem yang dibuat. Pengguna memberikan masukan bagi pengembang apakah prototipe memuaskan atau tidak. Jika ya, langkah berikutnya akan diambil dan jika tidak, prototipe akan direvisi atau diperbaiki dengan mengulangi langkah 1 sampai 3 dengan penggunaan yang lebih baik mengenai suatu kebutuhan pengguna sistem.
4. Pengkodean sistem. Pada tahap ini dalam memenuhi pengkodean disesuaikan dengan prototipe yang disetujui.

5. Pengujian sistem. Pengujian sistem dilakukan oleh pengembang dalam sistem yang dibuat, apakah sesuai dengan spesifikasi kebutuhan dan sebagai penanganan *error* atau ketidaksesuaian sehingga perlunya perbaikan.
6. Persetujuan penggunaan sistem. Pengguna memberikan masukan pada pengembang apakah sistem yang dibuat dapat diterima atau tidak. Jika ya, langkah 7 dilakukan dan jika tidak, langkah 4 dan langkah 5 akan diulangi.
7. Penggunaan sistem.



Gambar 2.4 Prototipe Jenis II

Sumber : Buku Sistem Informasi Manajemen, 2007

Penggunaan prototipe dalam mengembangkan sistem informasi memiliki beberapa keuntungan, diantaranya :

- a) Menghasilkan suatu syarat yang lebih baik dari produksi yang dihasilkan.
- b) Sedikit perubahan sistem yang dibuat dengan prototipe.
- c) Menghasilkan produk yang lebih akurat.

#### 2.2.4 WEB

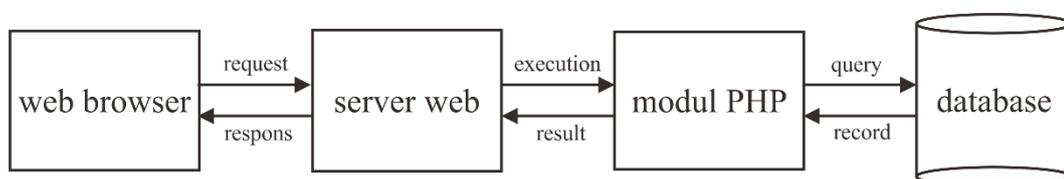
*Web* merupakan singkatan *World Wide Web* (WWW), yang menyajikan ruang informasi yang terdiri dari halaman-halaman web menggunakan konsep multimedia dengan informasi yang disajikan dalam berbentuk teks, gambar, animasi, suara dan gabungan semuanya menjadi lebih statis dengan menggunakan *hyperlink* yang disediakan dalam dokumen yang ditampilkan dalam sebuah *web browser* [8] [16] [17] [18]. Web dapat dikatakan sebagai *server-side* dan *client-server*. Server adalah komputer yang berjalan tanpa henti sebagai tempat

file/program dari aplikasi web diletakkan. Client adalah *users* yang hendak mengakses program ke server dengan menggunakan *address* (alamat) yang unik secara umum menggunakan *web browser*.

Dalam menggunakan aplikasi web dibutuhkan sebuah perangkat lunak atau *software* yang dikodekan dengan bahasa pemrograman seperti HTML, CSS dan Javascript sebagai pembuatan kerangka tampilan web, PHP sebagai pendukung eksekusi kode HTML dan *web server* sebagai integrasi serta menggunakan *web browser* untuk menampilkan hasil perintah atau sesuai dengan perancangan dokumen HTML yang dibuat. Keunggulan aplikasi berbasis Web di antaranya:

1. Dapat dijalankan dimanapun dan kapanpun menggunakan jaringan internet tanpa penginstallan.
2. Tidak memerlukan lisensi dan tidak tergantung platform.
3. Fleksibel dalam penggunaan di sistem operasi.
4. Diakses banyak perangkat seperti komputer, laptop, *smartphone* dll.
5. Tidak memerlukan spesifikasi yang tinggi dalam menggunakan aplikasi ini.
6. Mudah dikembangkan dan konfigurasi *server* lebih mudah.
7. Tidak memerlukan spesifikasi perangkat yang tinggi dalam perancangan.

Dalam perancangan sebuah aplikasi berbasis web memerlukan atau memahami perangkat pendukung seperti sebuah perangkat keras (*hardware*) dan perangkat lunak (*software*) serta perangkat pendukung lainnya [19].



Gambar 2.5 Proses Kerja Web

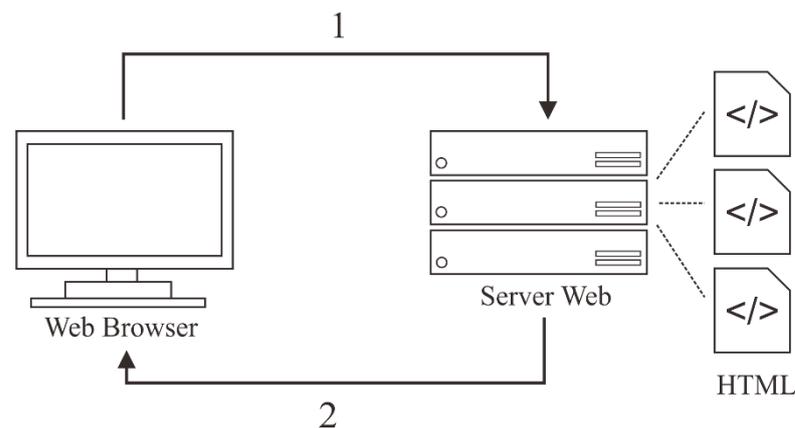
Sumber : Buku Aplikasi Pemrograman Web Dinamis dengan PHP dan MySQL, 2009

Pada gambar diatas, setiap *request* (permintaan) dari *web browser* akan dilayani dan diterima oleh modul PHP melalui *server web*, modul PHP akan melakukan *query* kedalam *database* sesuai permintaan dari *server web*. Selanjutnya *server web* akan memberikan hasil dari proses modul PHP yang berupa dokumen HTML kepada pengunjung. PHP dan *database* merupakan salah satu solusi yang

dapat digunakan dalam pengelolaan suatu situs *web* yang tidak perlu mengubah dokumen HTML akan tetapi cukup dengan menambah atau merubah *database*.

#### 2.2.4.1 Web Server

*Web Server* digunakan dalam menjalankan program PHP yang terintegrasi dengan aplikasi luar dan menjadi bagian dari *web server*. *Web server* merupakan salah satu sebuah mesin dimana terdapat tempat software atau aplikasi beroperasi dalam mendistribusikan suatu *web page* ke *users* (pengguna) dengan menggunakan protokol untuk media *transfer* atau memindahkan berkas yang diminta oleh pengguna melalui sebuah protokol komunikasi tertentu [9] [11] [16] [20]. Secara garis besarnya, *server web* digunakan dalam pemrosesan dan penanganan permintaan web. Cara HTML diproses oleh *server web* yaitu dengan pengaksesan sebuah *website* biasanya mengetikkan sebuah URL atau alamat halaman pada *web browser* setelah berhasil menerima akses dari *web browser* selanjutnya meminta akses kepada *web browser*. Disitulah *web server* akan berjalan yaitu untuk menelusuri data yang diperintahkan *web browser*, lalu mengirimkan data tersebut kepada *web browser* atau menolaknya jika ternyata data yang diminta tidak ditemukan.



Gambar 2.6 Proses HTML pada Server Web

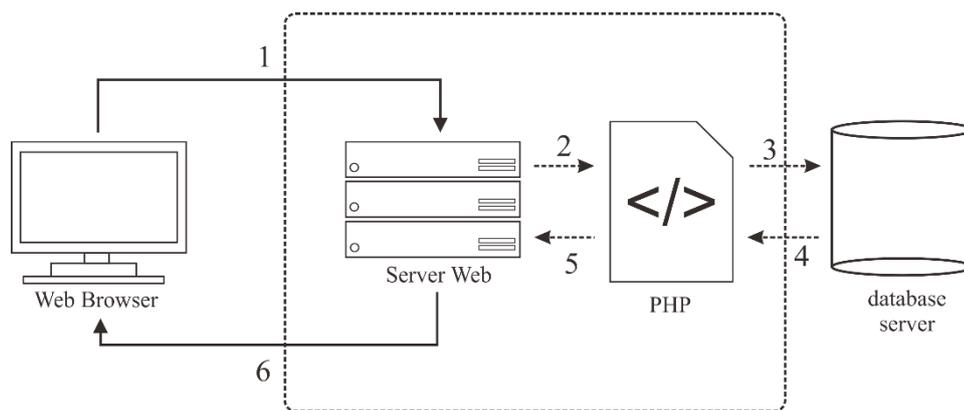
Sumber : Ebook Panduan Belajar PHP Lengkap untuk Pemula, 2015

#### 2.2.4.2 PHP (Hypertext Preprocessor)

*Hypertext Preprocessor* (PHP) merupakan bahasa pemrograman web yang dapat men-*generate* atau membuat kode HTML yang digunakan dalam pemrosesan form, mengakses *database*, management *session* dan *cookie*, membaca file teks, menangani file *upload*, membuat file pdf, membuat file excel, dan lain sebagainya lagi agar dapat dijalankan dalam *web server*. Hal ini karena PHP merupakan sebuah

bahasa pemrograman web *server side* (*server side programming language*) [16] [21] [22]. PHP diartikan sebagai bahasa pemrograman *script server-side* yang didesain menggunakan web atau dikatakan sebagai pemrograman umum. PHP juga diartikan sebagai bahasa pemrograman *script* yang menerjemahkan kode-kode PHP dalam HTML dalam perancangan aplikasi web. Contoh dari framework PHP diantaranya adalah seperti Codeigniter, Zend, Laravel, Symphony, Wordpress dan Drupal.

Dalam perjalanannya, PHP sudah dikembangkan secara berkala dari pengembangan pertama PHP dimunculkan [21], sebagai berikut :



Gambar 2.7 Proses PHP pada Server Web

Sumber : Ebook Panduan Belajar PHP Lengkap untuk Pemula, 2015

Dalam pemrosesan PHP pada *Server Web*, dapat dijelaskan sesuai gambar diatas sebagai berikut :

1. *Web browser* melakukan permintaan kepada *web server* dengan mengetikkan URL atau alamat dengan permintaan file PHP.
2. Ketika *request* (permintaan) berhasil dikirimkan, maka *web server* akan melihat permintaan *web browser*. Karena permintaan adalah file PHP maka *web server* memanggil sesuai dengan permintaan dan melakukan pemrosesan sesuai dengan aturan PHP.
3. Pada saat *web server* memproses kode PHP, file PHP bisa saja memiliki perintah untuk mengambil data dari sebuah *database*. Proses permintaan ini selanjutnya dikirim lagi ke *database server*, misalnya *database MySQL*.
4. *Database server* melakukan pemrosesan terhadap permintaan dari PHP dan mengirimkan hasilnya. Proses ini bisa berupa pencarian data, mengisi

data baru, mengubah data yang sudah ada atau menghapus data yang ada di dalam *database*. Seluruh perintah ini ditulis dengan menggunakan PHP.

5. Modul PHP yang terdapat di dalam *web server* kemudian menyerahkan hasil proses kepada *web server*. *Web server* kemudian men-*generate* kode PHP menjadi kode HTML.
6. Kode HTML hasil dari pemrosesan selanjutnya dikirim ke *web browser* (client) untuk ditampilkan.

### 2.2.4.3 Wampp atau Xampp

Xampp atau Wampp merupakan suatu perangkat lunak yang bebas atau *open source* (gratis) yang banyak mendukung sebuah sistem operasi. Xampp atau Wampp merupakan *tools* yang menyediakan banyak paket perangkat lunak yang dibungkus dalam satu paket (*HTTP Server, Apache, MySQL dan PHP*). Dengan adanya Xampp atau Wampp tidak diperlukan dalam *konfigurasi web server Apache, PHP dan MySQL* secara manual [11] [16] [23]. Dengan adanya Xampp atau Wampp memungkinkan pengembang aplikasi web dapat menjalankan program web secara *offline* atau tanpa adanya bantuan jaringan internet.

Dalam menjalankan Xampp atau Wampp memiliki perbedaan ketika menjalankan program pada operasi sistem yang berbeda (Windows dan Linux), langkah membuka Xampp di sistem operasi Windows, sebagai berikut :

*Pilih menu Start → Pilih Aplikasi Xampp → Jalankan Aplikasi Xampp*

Sedangkan dalam sistem operasi Linux, dalam membuka aplikasi Lampp/Xampp adalah sebagai berikut :

1. Buka terminal pada Linux.
2. Ketikkan *sudo /opt/lampp/lampp start* pada terminal.
3. Klik Enter. Maka aplikasi Lampp secara otomatis telah berjalan/hidup.
4. Jika mematikan Lampp, ketikkan pada terminal *sudo /opt/lampp/lampp stop*. Maka secara otomatis aplikasi Lampp berhenti/mati.

### 2.2.4.4 MySQL

MySQL merupakan sebuah nama *database server* dalam pengorganisasian data yang bertujuan dalam penyimpanan, pengaksesan dan mengelola data dalam *database* yang terintegrasi dengan aplikasi web dengan *database* yang kuat serta stabil sebagai media penyimpanan data [11] [22] [24] [25]. MySQL sebagai

*database server* yang mampu untuk memenejemen *database* dengan baik serta memiliki beberapa keistimewaan selain portabilitas, MySQL juga memiliki keamanan dan performa yang cukup baik. Selain MySQL juga terdapat beberapa jenis *database server* yang memiliki kemampuan sama dan tidak bisa dianggap mudah, contohnya seperti Oracle dan Postgre SQL.

#### **2.2.4.5 PhpMyAdmin**

PhpMyAdmin merupakan perangkat lunak bebas atau bisa dikatakan *open source* (gratis) yang ditulis menggunakan bahasa pemrograman PHP yang berguna dalam menangani suatu administrasi MySQL melalui jejaring WWW (*World Wide Web*) [24]. PhpMyAdmin juga dapat mendukung berbagai operasi MySQL, diantaranya dapat mengolah *database* (basis data), *tables* (tabel-tabel), *field* (bidang), *relations* (relasi), *index*, *user* (pengguna), *permissions* (perizinan) dan yang lainnya.

#### **2.2.4.6 Apache**

Apache merupakan *software* yang digunakan *web server* dalam menjalankan pemrograman PHP yang memungkinkan adanya sedikit konfigurasi didalamnya [16] [17] [24]. Apache juga terbungkus dengan paket *software* Xampp atau Wampp bersama MySQL dalam menjalankan program secara *offline*.

#### **2.2.4.7 Text Editor**

*Text editor* merupakan suatu aplikasi yang dapat digunakan untuk membuat dan mengedit *file text* atau kode HTML yang digunakan dalam pemrograman terutamanya pemrograman berbasis web [17] [26]. *Text editor* banyak ditemukan dalam sebuah sistem operasi salah satunya notepad pada windows. Namun Notepad kurang cocok digunakan untuk pemrograman karena tampilan yang sederhana dengan tidak adanya bantuan sebuah *plugin* sehingga memungkinkan pengembang membuat aplikasi secara manual. Pada penelitian ini menggunakan *Sublime Text* sebagai suatu alat yang digunakan untuk membangun aplikasi. *Sublime Text* merupakan editor teks yang memiliki kemampuan menerjemahkan sebuah bahasa pemrograman termasuk pemrograman PHP. *Sublime Text* merupakan *editor text platform* dengan *python application programming interface* (API) yang tanpa lisensi perangkat. *Sublime text editor* mendukung berbagai bahasa pemrograman dan bahasa *markup* yang fungsinya dapat menambahkan *plugin*.

#### 2.2.4.8 Web Browser

*Web browser* merupakan sebuah aplikasi yang dapat digunakan dalam menampilkan hasil kode-kode HTML atau dalam menampilkan *request* (permintaan) pada pencarian halaman web [17] [26]. *Web browser* memungkinkan dapat menampilkan antarmuka sesuai dengan kode yang dibuatnya. Aplikasi *web browser* banyak dipasang dalam sistem operasi seperti pada windows yaitu Google Chrome, Mozilla Firefox dan lainnya. Sama halnya dengan di operasi linux seperti Google Chrome dan web browser lainnya yang sesuai dengan operasi varian sistem operasi linux.

#### 2.2.5 Permodelan UML

*Unified Modelling Language (UML)* merupakan sebuah pemodelan sistem yang digunakan untuk menspesifikasikan, memvisualisasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak [27] [28]. UML sebagai suatu alat yang digunakan dalam memecahkan sebuah permasalahan dengan mendiskripsikan hasil analisa sehingga menghasilkan visualisasi, perancangan dan dokumentasi perangkat yang dibangun terkait penggambaran sistem *software* dengan suatu objek.

UML memiliki tujuan utama dalam mendesain sebuah kerangka sistem, adapun tujuannya adalah sebagai berikut :

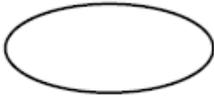
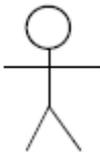
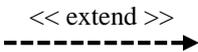
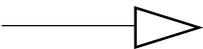
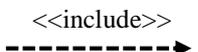
1. Menyediakan pengguna (sebagai analisis dan desain sistem) suatu bahasa permodelan visual yang sangat ekspresif sehingga dapat dikembangkan dan dapat melakukan pertukaran model data yang bermakna.
2. Menyediakan sebuah mekanisme yang digunakan dalam memperluas konsep inti.
3. Memberikan dasar formal untuk sebuah pemahaman bahasa permodelan.
4. Mendukung suatu konsep dalam pembangunan tingkat yang lebih tinggi seperti sebuah kolaborasi, kerangka atau pola dan komponen terhadap suatu sistem.

##### 2.2.5.1 Use Case Diagram

*Use case diagram* adalah suatu permodelan yang digunakan untuk *behavior* (kelakuan) sistem informasi yang dibuat yang mendiskripsikan sebuah interaksi antara satu atau lebih aktor dalam menggambarkan sebuah fungsi-fungsi yang

diharapkan dalam perancangan yang meliputi adanya *actor*, *use case* dan relasi sebagai urutan tindakan pada aktor dengan penilaian terukur [27] [28]. Diagram ini juga dimanfaatkan untuk permodelan atau mengorganisasikan sebuah perilaku suatu sistem yang diperlukan oleh penggunanya.

Tabel 2.1 Simbol Use Case

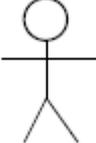
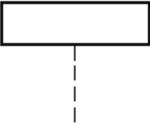
No.	Simbol	Nama	Deskripsi
1.		<i>Case</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit dan aktor. Simbol ini juga Menggambarkan proses atau kegiatan yang dapat dilakukan oleh aktor.
2.		<i>Actor</i>	Orang, proses, atau sistem yang Menggambarkan entitas atau subjek yang dapat melakukan suatu proses.
3.		<i>Association</i>	Komunikasi antara aktor <i>dan use case</i> yang berpartisipasi pada <i>use case</i> .
4.		<i>Extend</i>	Relasi use case dimana use case yang dapat menambahkan use case lain yang berdiri sendiri tanpa use case tambahan.
5.		<i>Generalitation</i>	Hubungan generalisasi (umum - khusus) antara use case satu dan lainnya dimana fungsi yang satu adalah fungsi yang lebih umum dibandingkan dengan lainnya.
6.		<i>Include</i>	Ada dua sudut pandang yang cukup besar mengenai include :

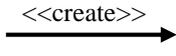
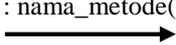
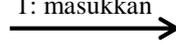
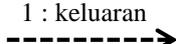
			<p><i>Include</i> berarti <i>use case</i> yang telah ditambah akan selalu dipanggil ketika <i>use case</i> sedang dijalankan. Seperti sebuah validasi <i>username</i> yang include dengan login.</p> <p><i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu melakukan pengecekan apakah <i>use case</i> yang akan ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dapat dijalankan.</p>
--	--	--	---

### 2.2.5.2 Sequence Diagram

*Sequence diagram* adalah suatu diagram yang menggambarkan proses kelakuan objek pada *use case* dengan mendiskrisikan hidup objek dan *message* (pesan) yang akan dikirimkan serta diterima antar objek dengan penggambaran interaksi antar objek yang tersusun berdasarkan waktu [27] [28]. Secara mudah diagram ini adalah gambaran tahap demi tahap termasuk kronologi atau skenario yang dilakukan oleh aktor maupun sistem yang merupakan respons dari sebuah kejadian yang menghasilkan *output* atau keluaran.

Tabel 2.2 Simbol Sequence Diagram

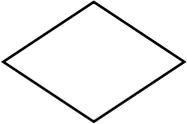
No.	Simbol	Nama	Deskripsi
1.		<i>Actor</i>	Orang, proses atau sistem yang lain yang berinteraksi dengan sistem informasi diluar sistem informasi yang dibuat itu sendiri.
2.		<i>Object</i>	Menggambarkan sebuah pos-pos suatu objek yang pengirim dan penerima pesan ( <i>message</i> ).

3.		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
4.		<i>Waktu Aktif</i>	Menyatakan objek dalam keadaan aktif dan sedang beriteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
5.		<i>Pesan Tipe Create</i>	Menyatakan suatu objek yang membuat suatu objek yang lain, arah panah mengarah pada objek yang dibuat.
6.		<i>Pesan Tipe Call</i>	Menyatakan suatu objek yang memanggil operasi /metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki fungsi operasi.
7.		<i>Pesan Tipe Call</i>	Menyatakan bahwa suatu objek mengirimkan sebuah data, masukan atau informasi ke objek lainnya. Arah panah mengarah pada objek yang akan dikirim.
8.		<i>Pesan Tipe Return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode yang menghasilkan suatu kembalian kepada objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

### 2.2.5.3 Activity Diagram

*Activity diagram* adalah suatu diagram yang menggambarkan sebuah *workflow* (aliran kerja) atau aktivitas dari sebuah sistem yang ada pada perangkat lunak dengan menggambarkan aktifitas atau kegiatan-kegiatan dalam sebuah operasi, diagram ini juga menggambarkan sebuah alur kerja dari suatu perilaku sistem [27] [28].

Tabel 2.3 Simbol Activity Diagram

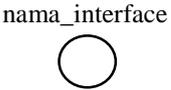
No.	Simbol	Nama	Deskripsi
1.		<i>Activity</i>	Menggambarkan aktivitas yang dilakukan oleh sistem, biasanya diawali kata kerja.
2.		<i>Decision</i>	Menggambarkan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
3.		<i>Join</i>	Menggambarkan asosiasi percabangan dimana jika ada pilihan aktivitas yang digabungkan menjadi satu.
4.		<i>Initial State</i>	Menggambarkan titik awal siklus hidup suatu elemen.
5.		<i>Final State</i>	Menggambarkan titik akhir yang menjadi kondisi akhir suatu elemen.

### 2.2.5.4 Class Diagram

*Class diagram* adalah suatu diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem yang memiliki atribut dari sebuah operasi dengan menggambarkan struktur kelas-kelas dari suatu sistem yang paling banyak dipakai atau digunakan [27] [28]. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas. Sedangkan operasi atau metode merupakan sebuah fungsi-fungsi yang dimiliki oleh suatu kelas. *Class*

*diagram* dibuat agar orang yang mengembangkan sebuah aplikasi dapat menyesuaikan dengan kelas-kelas sesuai dengan perancangan didalam diagram kelas guna adanya sinkronisasi antara perancangan dan aplikasi yang dibuat.

Tabel 2.4 Simbol Class Diagram

No.	Simbol	Nama	Deskripsi
1.		<i>Class</i>	Kelas pada struktur sistem.
2.		<i>Interface</i>	Sama dengan konsep <i>interface</i> dalam suatu pemrograman berorientasi objek.
3.		<i>Association</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya disertai <i>multiplicity</i> .
4.		<i>Direction Association</i>	Relasi antarkelas dengan kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.		<i>Generalitation</i>	Relasi antarkelas yang dengan makna sebuah generalisasi-spesialisasi.
6.		<i>Dependency</i>	Relasi antarkelas yang kebergantungan antarkelas.
7.		<i>Aggregation</i>	Relasi antarkelas dengan makna semua-bagian.

### 2.2.6 Jenis Pengujian Sistem

Pengujian sistem difungsikan sebagai menguji kelayakan suatu sistem dan memeriksa apakah hasil implementasi ada kesesuaian dengan analisis kebutuhan

dan perancangan sistem yang dibuat. Pengujian bisa berupa *black box testing* (pengujian validasi) dan *white box testing* (pengujian unit).

a) Black Bok Testing

*Black Box Testing* merupakan suatu pengujian yang difokuskan pada spesifikasi fungsional dari sebuah perangkat lunak, seorang *tester* dapat mendefinisikan sebuah kumpulan kondisi *input* (masukkan) dan melakukan pengetesan atau pengecekan pada suatu spesifikasi fungsional sebuah program [29]. Secara garis besar *black box testing* menguji pada fungsional dari sistem tanpa memperhatikan struktur dan proses programnya.



Gambar 2.8 Black Box Testing

Sumber : Jurnal Cendekia, 2014

b) White Box Testing

*White Box Testing* merupakan suatu pengujian pada pengecekan detail perancangan dan penggunaan struktur kontrol dari sebuah desain program secara prosedural untuk dapat membagi pengujian ke dalam beberapa kasus pengujian [29]. Tujuan dari pengujian *white box testing* untuk mengetahui bagaimana cara kerja perangkat lunak secara internal dan menjamin operasi-operasi internal sesuai dengan spesifikasi yang telah ditetapkan dengan menggunakan struktur kendali dari prosedur yang dirancang.



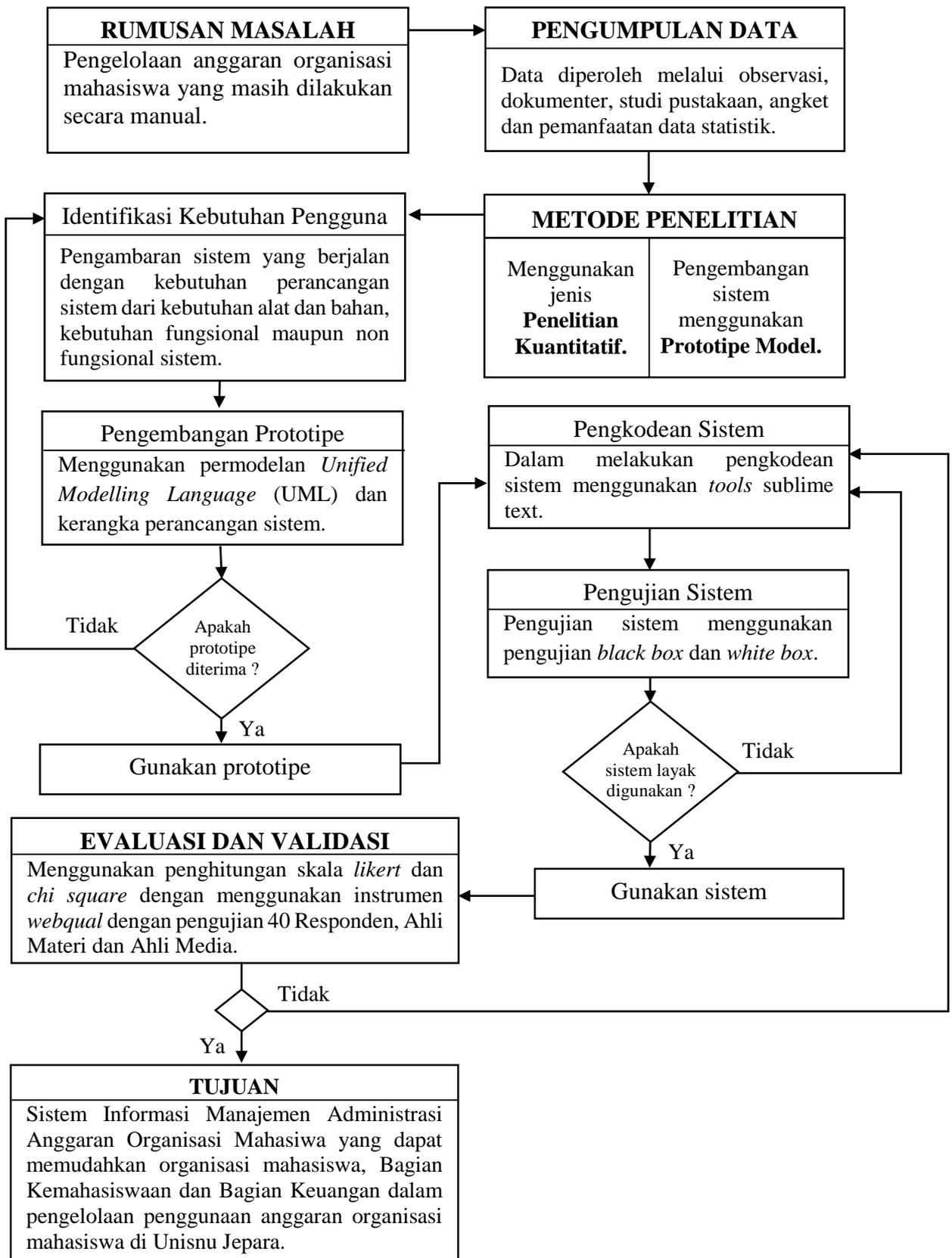
Gambar 2.9 White Box Testing

Sumber : Jurnal Cendekia, 2014

Pada penelitian ini menggunakan *black box testing* dan *white box testing* dalam pengujian sistem informasi manajemen administrasi anggaran organisasi di Unisnu Jepara yang difokuskan pada pengujian validasi dan pengujian unit suatu sistem yang dirancang.

### 2.3 Kerangka Pemikiran

Pada tahapan ini peneliti membuat suatu kerangka pemikiran secara bertahap. Kerangka pemikiran merupakan pola pikir atau konsep dalam melakukan suatu penelitian.



Gambar 2.10 Kerangka Pemikiran