

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

##### **2.1.1 Sistem Informasi Perencanaan Anggaran dan Biaya Universitas Respati Yogyakarta (Nuraini, 2015).**

Menurut Nuraini (2015), kegiatan perencanaan anggaran dan belanja tahunan merupakan kegiatan rutin yang harus disiapkan masing-masing unit di Universitas Respati Yogyakarta. Kegiatan ini bertujuan untuk menyelaraskan kegiatan-kegiatan yang tidak terlepas penggunaan anggaran. Penyusunan anggaran direncanakan di setiap awal tahun berdasarkan rencana kegiatan yang akan direncanakan.

Dalam proses pengajuan anggaran yang diajukan oleh masing-masing unit di Universitas Respati Yogyakarta sering sekali tidak sesuai seperti perencanaan anggaran di awal tahun kegiatan. Hal ini menyulitkan Bagian Keuangan untuk mempersiapkan pos anggaran rutin yang harus diberikan kepada masing-masing unit dikarenakan adanya kenaikan anggaran dan control penggunaan anggaran. Selain itu masing-masing unit juga merasa kesulitan dalam melakukan monitoring anggaran yang sudah digunakan. Hal ini dikarenakan belum adanya sistem yang sistematis dalam hal ini perangkat lunak khusus dalam proses monitoring dan control terhadap penggunaan anggaran.

Tujuan Penelitian adalah mengembangkan Sistem Informasi Rencana Anggaran Belanja di Universitas Respati Yogyakarta. Pengembangan perangkat lunak berbasis web dengan bahasa pengembangan PHP dengan DBMS Mysql sebagai basis data. Hasil penelitian diharapkan dapat mempermudah bagian keuangan atau masing-masing unit dalam melakukan monitoring dan kontroling penggunaan anggaran. Tampilan Sistem Informasi Perencanaan Anggaran dan Biaya Universitas Respati Yogyakarta terdapat pada Gambar 2.1. dan Gambar 2.2.



Sistem Informasi Rencana & Anggaran Belanja  
**UNIVERSITAS RESPATI YOGYAKARTA**

Login

Home Login

User Name:

Password:

Auto login until I logout explicitly

Save my user name

Always ask for my user name and password

Login

©2014. Biro Keuangan UNRPTO

Gambar 2.1 *Login* Sistem Informasi Perencanaan Anggaran dan Biaya Universitas Respati Yogyakarta



Sistem Informasi Rencana & Anggaran Belanja  
**UNIVERSITAS RESPATI YOGYAKARTA**

DATA-INDUK

TRANSAKSI

LAPORAN

Logout

Home > anggaranunit > Add

id anggaranunit:

Kode Tahun Anggaran:

Kode Anggaran:

Kode Unit Kerja:

Kuantitas Anggaran:

Rencana (Bulan-Tahun):

Realisasi (Bulan-Tahun):

Gambar 2.2 Tampilan Antar Muka Anggaran Unit

### **2.1.2 Sistem Informasi Rancangan Anggaran Biaya (RAB) Pada PT.PLN Distribusi Area Tegal (Prayitno, 2015).**

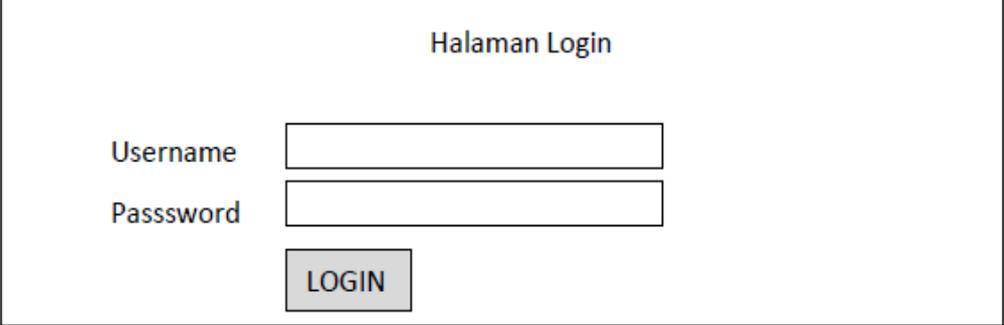
Menurut Prayitno (2015), penganggaran pemasangan baru atau penambahan daya merupakan kegiatan yang ditugaskan pada bagian perencanaan pada PT.PLN Distribusi Area Tegal. Namun fungsi media perhitungan yang digunakan masih sangat terbatas, hal ini dilihat dari penggunaan MS.Excel yang belum menggunakan sistem *online*. Permasalahan yang diangkat dalam penelitian ini adalah bagaimanakah pembuatan Sistem Informasi Rancangan Anggaran Biaya berbasis *online* untuk menunjang kualitas pekerjaan dalam segi efisiensi waktu, biaya, dan sumber daya manusia.

Metode yang digunakan dalam penelitian ini yaitu *Prototyping* yaitu dengan mengidentifikasi kebutuhan pemakai, mengembangkan *prototype*, menentukan apakah *prototype* dapat diterima, mengkodekan sistem operasional, menguji sistem operasional, menentukan jika sistem operasional dapat diterima, dan menggunakan sistem operasional. Pengujian dilakukan menggunakan *Blackbox testing* dengan tenaga ahli dari PT.PLN Distribusi sebagai *acceptance person* serta uji produk pada pegawai. Pengambilan data penelitian ini dilakukan dengan kegiatan *field research*, wawancara, dan pengamatan sehingga data yang diperoleh berupa data harga per satuan (HPS) 2014, dan blangko pembuatan RAB menggunakan MS.Excel.

Pengembangan sistem informasi RAB menggunakan *PHP* dan *javascript* sebagai bahasa pemrograman dalam pembuatan sistem. Pada uji kelayakan menggunakan *Black-box testing*, 100% sistem yang diusulkan kepada PT.PLN Distribusi Area Tegal masuk dalam kategori diterima. Pada uji kelayakan dari nilai teknis, operasional, dan ekonomis menunjukkan bahwa sistem yang diimplementasikan berhasil. Sehingga hasil penelitian menunjukkan bahwa Sistem Informasi Rancangan Anggaran Biaya pada PT.PLN Distribusi Tegal dapat diterima dengan baik.

Simpulan yang dapat dipaparkan oleh peneliti yaitu penerapan sistem informasi rancangan anggaran biaya dapat memudahkan pengguna dalam biaya, waktu, dan tenaga. Serta saran yang dapat diberikan oleh peneliti adalah dengan

kepelatihan khusus untuk pemeliharaan sistem yang berjalan. Tampilan sistem informasi rancangan anggaran biaya pada PT.PLN distribusi area Tegal, terdapat pada Gambar 2.3. dan Gambar 2.4.

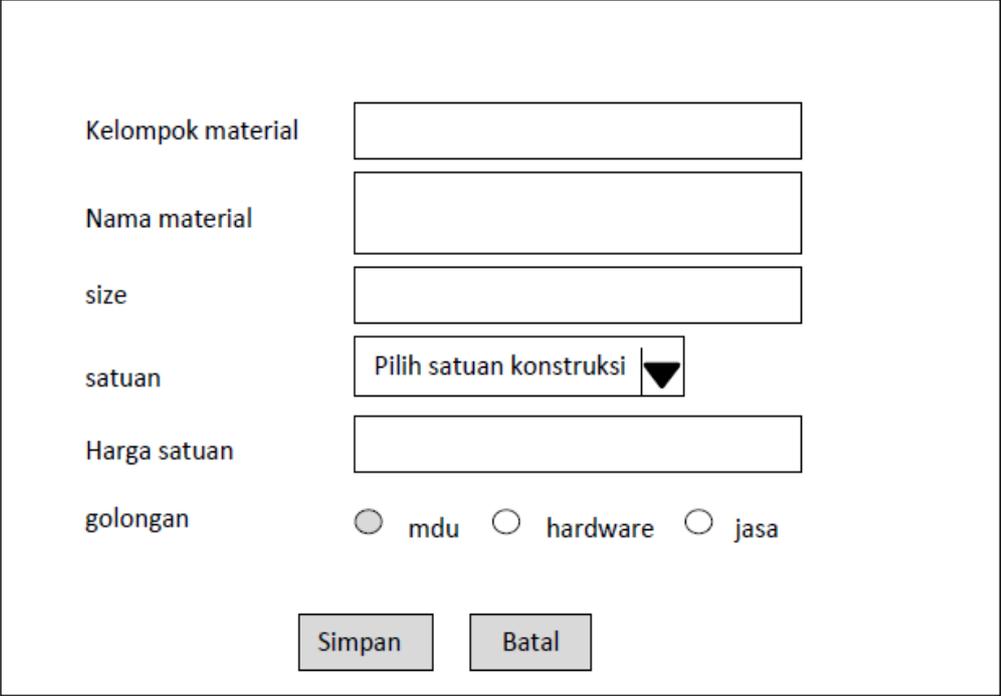


Halaman Login

Username

Passsword

Gambar 2.3 Halaman *Login*



Kelompok material

Nama material

size

satuan

Harga satuan

golongan  mdu  hardware  jasa

Gambar 2.4 Halaman *Input*

### **2.1.3 Sistem Informasi Manajemen Anggaran (Simangga) Perguruan Tinggi Berbasis Web (Rahmatullah, 2017).**

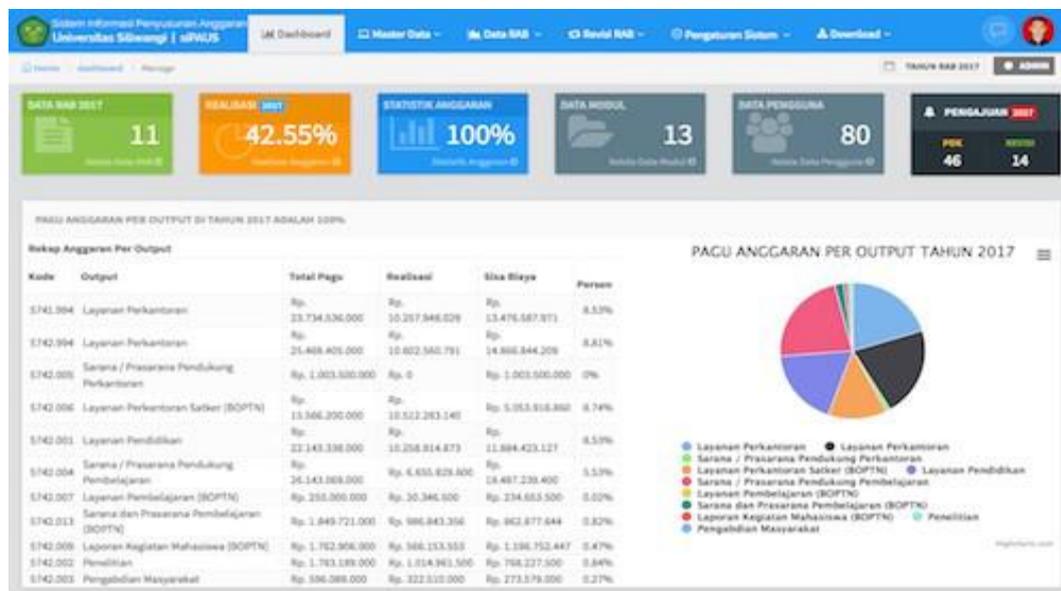
Menurut Rahmatullah (2017), pengelolaan anggaran merupakan suatu permasalahan yang umum terjadi di berbagai bidang ilmu. Permasalahan terjadi akibat tidak adanya manajemen anggaran yang baik, terencana, terukur dan realisasi dapat dipantau. Hampir di semua jenjang pendidikan, lembaga, sekolah ataupun perguruan tinggi permasalahan tersebut terjadi. Samahalnya dengan Universitas Siliwangi (UNSIL), sebuah universitas di Priangan Timur tepatnya di Kota Tasikmalaya yang beralih status dari Perguruan Tinggi Swasta menjadi Perguruan Tinggi Negeri Baru (PTNB). Pengelolaan anggaran di UNSIL masih manual ditambah lagi dengan statusnya sebagai PTNB yang jika dilihat cara pengelolaan anggarannya berbeda dengan swasta. Hal tersebut menjadi permasalahan yang cukup serius karena dapat berdampak pada semua aspek dalam penyelenggaraan kegiatan Perguruan Tinggi.

Sehingga dari status PTNB ini, Unsil merupakan salah satu Satker. Satuan kerja (satker) adalah instansi pemerintah yang merupakan bagian dari struktur kementerian di pemerintahan. Setiap satu satker mempunyai kepala satker (kepala dinas, kepala kantor dsb.), bendahara satker, pejabat pembuat komitmen, pejabat penanda tangan Surat Perintah Membayar (SPM), pembuat daftar gaji (PDG), dan pegawai yang bekerja di instansi tersebut. Satker mendapatkan dana dari Anggaran Pendapatan dan Belanja Negara (APBN) untuk melaksanakan kegiatannya. Satker sebagai organisasi pemerintah menjalankan fungsi akuntansi dan pengelolaan keuangan atas kegiatan yang dilakukannya.

Berdasarkan hasil penelitian yang telah dilakukan, dari permasalahan-permasalahan yang terjadi dapat diatasi dengan dibuatnya sistem manajemen anggaran sehingga menjadikan solusi yang tepat agar proses pengelolaan anggaran di Universitas Siliwangi berjalan lancar. Dimulai dari proses pembuatan rencana anggaran masing-masing unit, pengajuan realisasi, revisi sampai pelaporan.

Sistem manajemen anggaran ini dapat membantu pengelolaan anggaran internal sebelum digabungkan dengan sistem yang sudah disediakan oleh

pemerintah. Tampilan sistem informasi manajemen anggaran yang dimaksud disajikan pada Gambar 2.5.



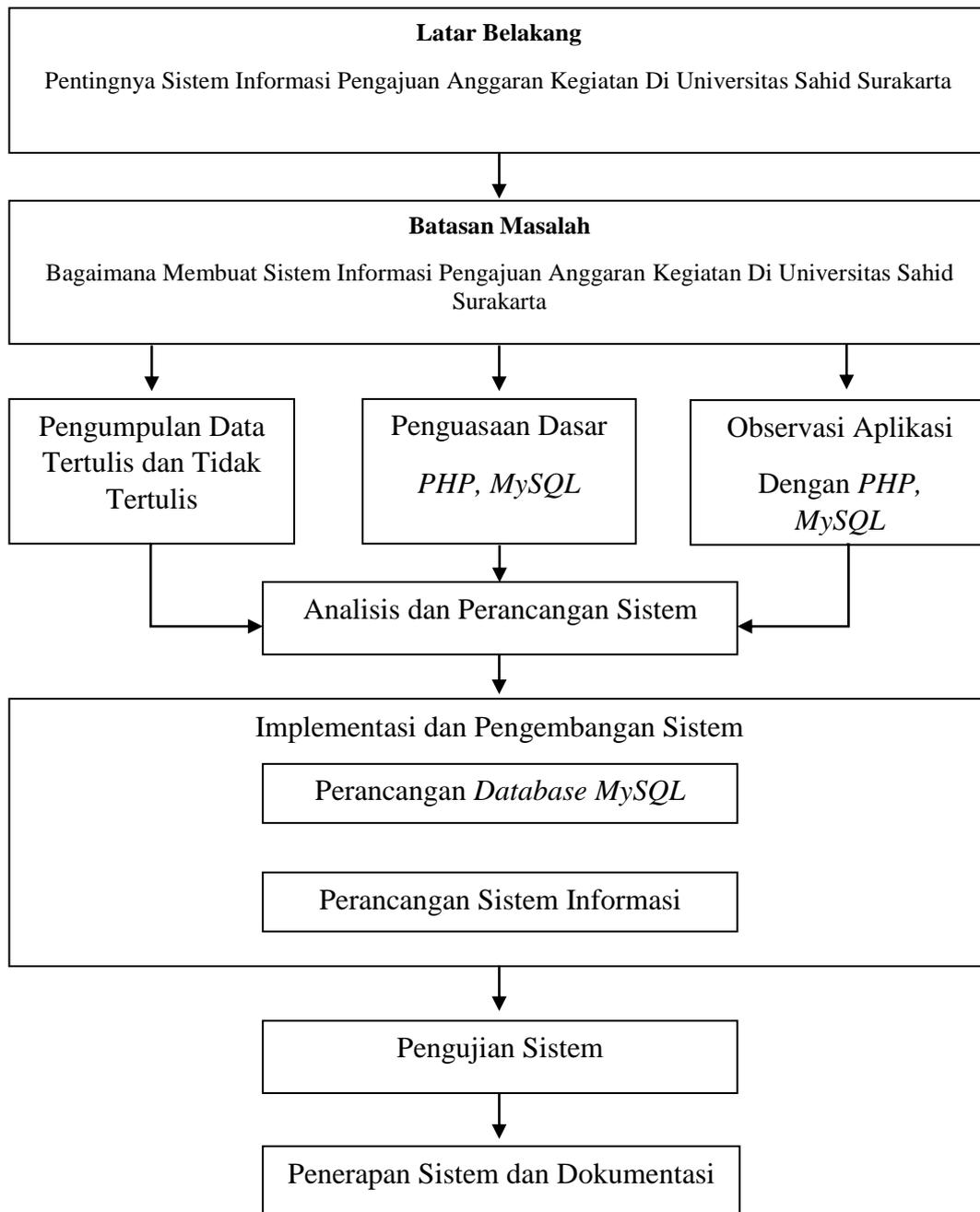
Gambar 2.5 Tampilan *Dashboard* antar muka

## 2.2 Kerangka Pemikiran

Pengelolaan anggaran kegiatan merupakan suatu permasalahan yang umum terjadi diberbagai bidang ilmu. Permasalahan terjadi akibat tidak adanya manajemen anggaran yang baik, terencana, terukur dan realisasi dapat dipantau. Hampir di semua jenjang pendidikan, lembaga, sekolah ataupun perguruan tinggi permasalahan tersebut terjadi. Bagian Keuangan merupakan komponen penting dalam pengelolaan keluar dan masuknya dana yang digunakan untuk kegiatan.

Di Universitas Sahid Surakarta pengelolaan anggaran kegiatan masih dilakukan secara manual, sehingga proses pengajuan anggaran kegiatan terbilang masih cukup rumit dan kurang efisien. Hal tersebut dapat berdampak pada semua aspek dalam penyelenggaraan kegiatan di Universitas Sahid Surakarta.

Mahasiswa Universitas Sahid Surakarta harus memahai secara detail tentang alur-alur pengajuan anggaran kegiatan, dengan adanya sistem informasi pengajuan anggaran kegiatan dapat menjadi fasilitas untuk memudahkan mahasiswa dalam mengajukan anggaran kegiatan secara lebih efektif dan efisien. Diagram kerangka pemikiran dalam penyusunan tugas akhir dapat dilihat pada Gambar 2.6.



Gambar 2.6 Kerangka Pemikiran

Uraian dari kerangka berfikir sebagai berikut :

#### 1. Latar Belakang Masalah

Latar belakang masalah pada tugas akhir ini adalah belum adanya sistem pengajuan anggaran kegiatan di Universitas Sahid Surakarta menyebabkan kurangnya efisiensi waktu dalam pengajuan anggaran dana untuk kegiatan.

## 2. Rumusan Masalah

Rumusan masalah pada tugas akhir ini adalah bagaimana cara membuat sistem informasi pengajuan anggaran kegiatan di Universitas Sahid Surakarta berbasis *website* yang dapat memberikan informasi dan pelayanan di bidang keuangan.

## 3. Pengumpulan Data

Pengumpulan data dalam penelitian ini ada dua yaitu pengumpulan data tertulis dan pengumpulan data tidak tertulis. Penelitian ini mengumpulkan semua data yang dibutuhkan dengan melakukan *interview* dengan bidang keuangan di Universitas Sahid Surakarta.

## 4. Penguasaan Dasar

Penguasaan dasar dalam penelitian ini adalah penguasaan yang telah diketahui atau dikuasai penulis mengenai *PHP* dan *MySQL*.

## 5. Observasi Aplikasi

Observasi alat dalam penelitian ini adalah penguasaan terhadap *software-software* yang akan digunakan untuk membuat sistem informasi pengajuan anggaran kegiatan di Universitas Sahid Surakarta.

## 6. Analisis dan Perancangan Sistem

Analisis dan perancangan sistem dalam penelitian ini adalah memahami dengan detail, apa yang harus dilakukan untuk sistem yang akan dibuat serta merancang sistem yang baru.

## 7. Implementasi dan Pengembangan Sistem

Implementasi sistem dalam penelitian ini adalah proses pelaksanaan penelitian, setelah dilakukan analisis dan perancangan kemudian dilakukan pembuatan sistem.

## 8. Pengujian Sistem

Pengujian sistem dalam penelitian ini adalah dilakukan untuk mengetahui jika ternyata masih ada kesalahan atau kekurangan pada sistem yang dibuat.

## 9. Penerapan Sistem dan Dokumentasi

Penerapan sistem dan dokumentasi dalam penelitian ini adalah proses menerapkan aplikasi di Universitas Sahid Surakarta dan proses pengambilan dokumentasi setelah sistem selesai dibuat dari keseluruhan kegiatan Tugas Akhir.

## **2.3 Teori Pendukung**

### **2.3.1 Pengertian Sistem**

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Informasi juga berarti kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Nuraini, 2015).

### **2.3.2 Pengertian Informasi**

Informasi adalah data yang telah diolah ke dalam suatu bentuk yang lebih memiliki arti dan dapat digunakan untuk pengambilan keputusan. Data sendiri merupakan fakta-fakta yang mewakili suatu keadaan, kondisi, atau peristiwa yang terjadi atau ada di dalam atau di lingkungan fisik organisasi (Prayitno, 2015).

### **2.3.3 Pengertian Sistem Informasi**

Sistem informasi adalah suatu sistem di 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 diperlukan (Prayitno, 2015).

### **2.3.4 HTML**

*HTML* atau *Hypertext Markup Language* merupakan bahasa pemrograman yang biasa digunakan untuk membuat aplikasi berbasis *web*. Bahasa pemrograman ini ditulis dalam berkas format *ASCII*, supaya dapat menghasilkan tampilan wujud yang terintergrasi. *HTML* adalah bahasa pemrograman web yang umum dan biasa digunakan serta mudah dipelajari (Maya, dkk, 2015).

### **2.3.5 Website**

*Website* merupakan sebuah kumpulan halaman yang diawali dengan halaman muka yang berisikan informasi, iklan, serta program aplikasi (Septiyanto, 2017).

### 2.3.6 Database

*Database* merupakan sebuah tempat untuk menyimpan data yang jenisnya beraneka ragam. Keuntungan menyimpan data di *database* adalah kemudahannya dalam menyimpan dan menampilkan data dalam bentuk table (Winarno, dkk, 2014).

### 2.3.7 MySQL

*MySQL* adalah nama *database server*. *Database server* adalah *server* yang berfungsi untuk menangani *database*. *Database* adalah suatu pengorganisasian data dengan tujuan memudahkan penyimpanan dan pengaksesan data. Dengan menggunakan *MySQL*, kita bisa menyimpan data dan kemudian data bisa diakses dengan cara yang mudah dan cepat (Septiyanto, 2017).

### 2.3.8 PHP (PHP Hypertext Processor)

PHP merupakan singkatan dari PHP Hypertext Processor. Ini merupakan bahasa interpreter yang mempunyai kemiripan dengan C dan Perl. PHP dapat digunakan bersama dengan HTML sehingga memudahkan dalam membuat aplikasi web dengan cepat. PHP berbentuk skrip yang ditempatkan dalam server dan di proses di *server*. Hasilnya dikirimkan ke *client* tempat *user* menggunakan *browser*. Dapat digunakan untuk menciptakan *website* dinamis baik itu yang memerlukan penggunaan *database* ataupun tidak (Nuraini, 2015).

Untuk *web*, *PHP* adalah bahasa *scripting* yang bisa dipakai untuk tujuan apapun. Di antaranya cocok untuk pengembangan aplikasi *web* berbasis *server* (*server-side*) dimana *PHP* nantinya dijalankan di *server web* (Winarno, dkk, 2014).

### 2.3.9 XAMPP

*XAMPP* merupakan *tool* yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan meng-*instal XAMPP* maka tidak perlu lagi melakukan instalasi dan konfigurasi *web server Apache*, *PHP* dan *MYSQL* secara manual (Hendra, 2018).

*XAMPP* merupakan paket *PHP* berbasis *open source* yang dikembangkan oleh sebuah komunitas *Open Source*. Dengan menggunakan *XAMPP* tidak perlu lagi melakukan penginstalan program yang lain karena semua kebutuhan telah

disediakan oleh *XAMPP*. Beberapa paket yang telah disediakan adalah *Apache*, *MySQL*, *Php*, *Filezilla*, dan *Phpmyadmin* (Hendra, 2018).

### **2.3.10 Bootstrap**

*Bootstrap* sebuah alat bantu untuk membuat sebuah tampilan halaman *website* yang dapat mempercepat pekerjaan seorang pengembang *website* ataupun pendesain halaman *website*. Sesuai namanya, *website* yang dibuat oleh alat bantu ini memiliki tampilan halaman yang sama atau mirip dengan tampilan halaman *twitter* atau *desainer* juga dapat mengubah tampilan halaman *website* sesuai dengan kebutuhan. Tampilan *website* yang dibuat *bootstrap* akan menyesuaikan ukuran layar dari *browser* yang kita gunakan baik *dekstop*, *tablet* ataupun *mobile device*. Fitur ini bisa diaktifkan ataupun di-*non*-aktifkan sesuai keinginan. Sehingga, membuat web untuk tampilan *dekstop* saja dan apabila di-*reader* oleh *mobile browser* maka tampilan dari *web* yang dibuat sesuai layar. Dengan *bootstrap* kita juga bisa membangun *web* dinamis ataupun statis (Widyantoro, 2014).

### **2.3.11 Perancangan Sistem**

Perancangan sistem secara umum didefinisikan sebagai pengidentifikasian komponen-komponen sistem informasi dengan tujuan untuk dikomunikasikan dengan pemakai (Yuniati, 2017).

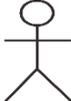
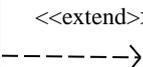
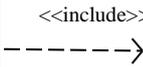
*Unified Modeling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan (Shalahuddin dan Sukamto, 2016).

#### **2.3.11.1 Use Case Diagram**

*Use case diagram* adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use case diagram* mendefinisikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. *Use case diagram* berhubungan erat dengan kejadian-kejadian. *Use case diagram* dibuat untuk menggambarkan hubungan

antara *actor* dan *use case*. Simbol-simbol dari *use case diagram* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol-simbol *Use Case Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.
2.		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> memiliki interaksi dengan aktor.
3.		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antar unit atau aktor.
4.		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5.		<i>Generalization</i>	Hubungan generalisasi dan spesifikasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6.		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

Sumber : Shalahuddin dan Sukamto (2016).

### 2.3.11.2 Class Diagram

*Class diagram* adalah himpunan dari objek-objek yang sejenis. Sebuah objek memiliki keadaan sesaat (*state*) dan perilaku (*behavior*). *State* sebuah objek adalah kondisi objek tersebut yang dinyatakan dalam *attribute* atau *propoerties*. Sedangkan perilaku suatu objek mendefinisikan bagaimana sebuah objek bertindak atau beraksi dan memberikan reaksi.

*Class diagram* mempunyai 3 macam *relationship* (hubungan) sebagai berikut:

a. *Association*

Suatu hubungan antara bagian dari dua kelas. Terjadi *association* antara dua kelas jika salah satu bagian dari kelas mengetahui yang lainnya dalam melakukan suatu kegiatan. Di dalam diagram, sebuah *association* adalah penghubung yang menghubungkan dua kelas.

b. *Aggregation*

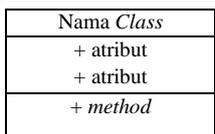
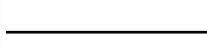
Suatu *association* dimana salah satu kelasnya merupakan bagian dari kumpulan. *Aggregation* memiliki titik pusat mencakup keseluruhan bagian.

c. *Generalization*

Suatu hubungan turunan dengan mengasumsikan satu kelas merupakan suatu *super class* (kelas *super*) dari kelas yang lain. *Generalization* memiliki tingkatan yang berpusat pada *super class*.

Simbol-simbol dari *class diagram* dapat dilihat pada Tabel 2.2.

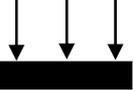
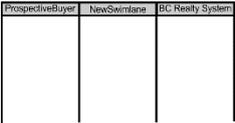
Tabel 2.2 Simbol-simbol *Class Diagram*

NO.	SIMBOL	NAMA	KETERANGAN
1.		<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian antar kelas.
2.		<i>Class</i>	Kelas pada struktur sistem.
3.		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.		<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
5.		<i>Directed Association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain.
6.		<i>Generalization</i>	Relasi antar kelas makna generalisasi-spesialisasi (umum khusus).
7.		<i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.

### 2.3.11.3 Activity Diagram

*Activity diagram* adalah diagram yang menggambarkan logika prosedural. Dalam beberapa hal, *activity diagram* memainkan peranan yang mirip dengan diagram alur tetapi perbedaan prinsip antara notasi diagram alur adalah *activity diagram* mendukung *behavior paralel*. *Node* pada sebuah *activity diagram* disebut dengan *action* sehingga diagram tersebut menampilkan sebuah *activity* yang tersusun dari *action*. Simbol-simbol dari *activity diagram* terdapat pada Tabel 2.3.

Tabel 2.3 Simbol-simbol *Activity Diagram*

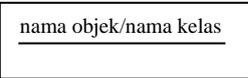
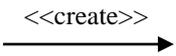
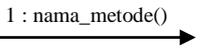
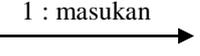
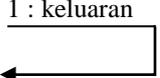
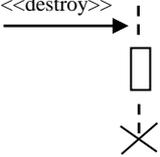
NO	GAMBAR	NAMA	KETERANGAN
1.		Status Awal	Status awal aktivitas sistem.
2.		Status Akhir	Status akhir dilakukan sebuah sistem.
3.		Aktivitas	Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja.
4.		Decision	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
5.		<i>Join</i>	Asosiasi penggabungan lebih dari satu aktivitas digabungkan jadi satu.
6.		<i>Swimlanes</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : Shalahuddin dan Sukamto (2016).

### 2.3.11.4 Sequence Diagram

*Sequence diagram* adalah grafik dua dimensi dimana objek ditunjukkan dalam dua dimensi horizontal, sedangkan *lifeline* ditunjukkan dalam dimensi vertikal. Simbol-simbol dari *sequence diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Simbol-simbol *Sequence Diagram*

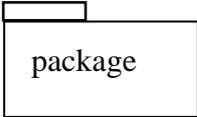
NO.	SIMBOL	NAMA	KETERANGAN
1.		Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2.		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
3.		Objek	Menyatakan objek yang berinteraksi pesan.
4.		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi.
5.		Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.		Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.
7.		Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.		Pesan tipe <i>return</i>	Menyatakan bahwa suatu objek telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah ke objek kembalian.
9.		Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber : Shalahuddin dan Sukamto (2016).

### 2.3.11.5 Component Diagram

Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem. Simbol-simbol yang ada pada *component diagram* dapat dilihat pada Tabel 2.5.

Tabel 2.5 Simbol *Component Diagram*

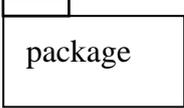
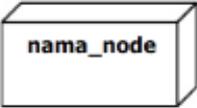
NO	GAMBAR	NAMA	KETERANGAN
1.	 package	<i>Package</i>	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen
2.	 component	<i>Component</i>	Komponen sistem
3.		<i>Dependency</i>	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
4.		<i>Interface</i>	Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen
5.		<i>Link</i>	Relasi antar komponen

Sumber : Shalahuddin dan Sukamto (2016).

### 2.3.11.6 Deployment Diagram

*Deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan sistem tambahan, sistem *client*, sistem terdistribusi murni, rekayasa ulang aplikasi. Simbol-simbol yang ada pada *deployment diagram* dapat dilihat pada Tabel 2.6.

Tabel 2.6 Simbol-Simbol *Deployment Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1.		<i>Package</i>	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih <i>node</i> .
2.		<i>Node</i>	Biasanya mengacu pada perangkat keras ( <i>hardware</i> ), perangkat lunak yang tidak dibuat sendiri ( <i>software</i> ), jika didalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelum pada diagram komponen.
3.		<i>Dependency</i>	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai.
4.		<i>Link</i>	Relasi antar <i>node</i> .

Sumber : Shalahuddin dan Sukamto (2016).

### 2.3.12 *Black Box Testing*

Pengertian *blackbox testing* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian ini dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji coba yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses *login* maka kasus uji yang dibuat adalah :

- a. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.

- b. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misal nama pemakai benar kata sandi salah atau sebaliknya, atau keduanya salah (Winarno, dkk, 2014).

Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian pengujian *black-box* memungkinkan perancang perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black-box* bukan merupakan alternatif dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan daripada metode *white-box*.

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut: (1) fungsi-fungsi yang tidak benar atau hilang, (2) kesalahan *interface*, (3) kesalahan dalam struktur data atau akses *database external*, (4) kesalahan kinerja, (5) inisialisasi dan kesalahan terminasi.

Pengujian *black-box* memperhatikan struktur kontrol, maka perhatian berfokus pada *domain* informasi. Pengujian di desain untuk menjawab pertanyaan-pertanyaan berikut :

1. Bagaimana validitas fungsional diuji ?
2. Kelas input apa yang akan membuat *test case* menjadi baik ?
3. Apakah sistem sangat sensitif terhadap harga input tertentu ?
4. Bagaimana batasan dari suatu data diisolasi ?
5. Kecepatan data apa dan volume data apa yang dapat ditolelir oleh sistem ?

Dengan mengaplikasikan teknik *black-box*, maka serangkaian *test case* akan memenuhi kriteria berikut ini : (1) *test case* yang mengurangi, dengan harga lebih dari satu, jumlah *test case* tambahan yang harus didesain untuk mencapai pengujian yang dapat dipertanggungjawabkan, dan (2) *test case* yang memberi tahu mengenai kehadiran atau ketidakhadiran kelas kesalahan, daripada memberitahu kesalahan yang berhubungan hanya dengan pengujian spesifik yang ada (Pressman, 2002).

### 2.3.13 Pengujian dengan Kuesioner

Kuesioner merupakan daftar pertanyaan yang sudah distandarisasikan dan menstrukturkan serta memperluas proses pengumpulan fakta. Kuesioner dapat mengarahkan sistem analisis melalui wawancara dan arena analisis menggunakan pertanyaan-pertanyaan yang sama, maka kuesioner dapat meningkatkan konsistensi pengumpulan fakta.

Teknik kuesioner untuk pengumpulan data merupakan salah satu cara yang baik untuk mendapatkan data yang akurat, dimana daftar pertanyaan yang dibuat adalah daftar yang berisi pertanyaan-pertanyaan untuk tujuan khusus yang memungkinkan sistem analisis untuk mengumpulkan data dan pendapat dari responden-responden yang dipilih. Daftar pertanyaan atau kuesioner ini akan dikirim kepada responden yang akan mengisinya sesuai dengan pendapat mereka. Penggunaan daftar pertanyaan ini mendapat banyak kritik karena diragukan hasilnya. Akan tetapi untuk mengumpulkan data dengan jumlah sumber yang banyak, tidak ada teknik pengumpulan data lainnya yang lebih dibandingkan kuesioner. Adapun petunjuk untuk membuat daftar pertanyaan adalah :

- a. Rencanakanlah terlebih dahulu fakta/opini apa saja yang ingin dikumpulkan.
- b. Berdasarkan opini tersebut di atas, tentukan tipe dari pertanyaan yang paling tepat untuk masing-masing fakta dan opini.
- c. Tulislah pertanyaan-pertanyaan yang diajukan. Pertanyaan itu tidak boleh mengandung kesalahan, serta harus jelas dan sederhana.
- d. Lakukan uji coba atas pertanyaan itu ke beberapa responden terlebih dahulu, misalnya dua atau tiga orang. Apabila responden mengalami kesulitan dalam mengisi daftar pertanyaan itu maka pertanyaan-pertanyaan itu harus diperbaiki lagi.
- e. Perbanyaklah dan distribusikanlah daftar pertanyaan yang memang sudah dianggap baik dan solid (Sutabri, 2012).