

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terkait**

Penelitian terkait adalah salah satu acuan bagi peneliti dalam melakukan penelitian. Penelitian terdahulu dapat menjadi referensi atau pembandingan serta pembeda bagi penelitian yang dilakukan saat ini untuk menghasilkan pengembangan ilmu pengetahuan. Berikut contoh bentuk penelitian terkait :

Pertama, penelitian yang dilakukan oleh Doni Mantovani pada tahun 2020 dengan jurnal “Perancangan sistem informasi keuangan sekolah berbasis web pada SMA Yapermas Jakarta”. Tujuan dari penelitian ini adalah transaksi pembayaran dan laporan keuangan sekolah pada SMA Yapermas bisa dilakukan secara online dan tersistem oleh database dan tidak perlu lagi menggunakan kertas dalam hal pembayaran SPP, uang gedung dan pembayaran lainnya serta mengelola laporan keuangan sekolah secara digital. Hasil dari penelitian ini adalah dapat menghasilkan suatu aplikasi sistem informasi dalam melakukan proses pencatatan keuangan secara digital.

Kedua, penelitian yang dilakukan oleh Raafiud Dwinata pada tahun 2019 dengan jurnal “Sistem Informasi Keuangan Berbasis Web Untuk Pembayaran Uang Pendidikan Studi Kasus BMT Bening Suci Prambanan Yogyakarta”. Tujuan dari penelitian ini adalah Untuk mempermudah pihak tata usaha dalam melakukan pembayaran dan laporan keuangan sekolah. Penelitian ini telah menghasilkan aplikasi yang cukup baik, terdapat fitur pembayaran dan pengolahan laporan keuangan sekolah.[2]

Ketiga, penelitian yang dilakukan oleh Heri Purnom pada tahun 2018 dengan jurnal “Sistem Informasi Pengolahan Data Keuangan Berbasis Web”. Tujuan dari penelitian ini adalah terwujudnya Implementasi Sistem Informasi Keuangan berbasis web untuk mendukung penyajian informasi dalam proses pengambilan keputusan oleh pimpinan, maupun bagian keuangan.

Metode yang digunakan adalah metode analisis dan metode perancangan. Metode perancangannya yaitu membuat implentasi sistem informasi, desain basis data dan analisis data keuangan. Penelitian terdahulu dapat dilihat pada Tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

No	Penulis (tahun)	Judul	Metode	Persamaan	Perbedaan
1.	Doni mantovani 2020, Jurnal perangkat lunak,(JATI) Vol. 2 No.2	Perancangan sistem informasi keuangan sekolah berbasis web pada sma yapermas jakarta”.	OOP (Object Oriented Programming)	Sistem yang di bahas tentang keuangan sekolah	Sistem yang dibangun berbasis <i>web native</i> , sedangkan penulis menggunakan framework ci 4
2.	Raafiud Dwinata 2019 jurnal sarjana informatika Vol. 7 No.1	Sistem Informasi Keuangan Berbasis Web Untuk Pembayaran Uang Pendidikan	Metode <i>waterfall</i>	Sistem yang di bahas tentang keuangan sekolah	Sistem yang dibangun berbasis <i>web native</i> , sedangkan penulis menggunakan framework ci 4
3.	Heri purnomo, 2018 Jurnal informatika merdeka Vol.3 No.1	Sistem Informasi Pengolahan Data Keuangan Berbasis Web	Metode <i>diagram</i> perancangan	Sistem yang di bahas tentang keuangan sekolah	Sistem yang dibangun berbasis <i>web native</i> , sedangkan penulis menggunakan framework ci 4
4	Aldi Prasetio,2019 Juenal Sistem Informasi dan Komputer Akuntansi	Rancang Bangun Sistem Informasi Administrasi Keuangan Sekolah Berbasis Web	<i>Metode Prototyping</i>	Sistem yang dibahas tentang pencatata keuangan sekolah	Sistem yang dibangun berbasis <i>web native</i> , sedangkan penulis menggunakan framework ci 4

## 2.2 Teori yang berhubungan dengan sistem secara umum

### 2.2.1 Pengertian Sistem

Sistem dapat didefinisikan dengan dua (2) pendekatan sistem, yaitu didefinisikan dengan pendekatan prosedur dan didefinisikan dengan pendekatan komponen. Pendekatan sistem yang didefinisikan dengan pendekatan prosedur ialah sebagai berikut :

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk tujuan tertentu [2].

Sedangkan pendekatan sistem yang didefinisikan dengan pendekatan komponennya ialah sebagai berikut:

Sistem merupakan kumpulan elemen-elemen yang saling terkait dan bekerja sama untuk memproses masukan (*input*) yang ditunjukkan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (*output*) yang diinginkan [3].

### 2.2.2 Pengertian Keuangan

Keuangan adalah mempelajari bagaimana individu, bisnis, dan organisasi meningkatkan, mengalokasi, dan menggunakan sumber daya moneter sejalan dengan waktu, dan juga menghitung risiko dalam menjalankan proyek mereka. Istilah dapat berarti, Ilmu keuangan dan asset lainnya, Manajemen asset tersebut, Menghitung dan mengatur risiko proyek. Menurut (Ridwan dan Inge, 2003) Keuangan Merupakan Ilmu dan seni dalam mengelola uang yang mempengaruhi kehidupan setiap organisasi. Keuangan berhubungan dengan proses, lembaga, pasar, dan instrument yang terlibat dalam transfer uang diantara individu maupun antara bisnis dan pemerintah.

Keuangan juga biasa didefinisikan sebagai manajemen keuangan, yaitu segala kegiatan perusahaan yang berhubungan dengan cara memperoleh, menggunakan, serta mengelola keuangan perusahaan untuk mencapai tujuan utama perusahaan. Manajemen keuangan merupakan hal yang penting untuk diterapkan pada

perusahaan. Hal ini dikarenakan, keuangan merupakan salah satu pondasi yang kuat untuk pertumbuhan dan perkembangan perusahaan [4].

## **2.3 Teori yang berhubungan dengan Topik yang diangkat**

Teori yang berhubungan dengan topik yang diangkat meliputi :

### **2.3.1 Pengertian *Framework* ( Kerangka kerja )**

Kecerdasan Kerangka kerja atau framework adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah kompleks. Istilah ini sering digunakan antara lain dalam bidang perangkat lunak untuk menggambarkan suatu desain sistem perangkat lunak yang dapat digunakan kembali, serta dalam bidang manajemen untuk menggambarkan suatu konsep yang memungkinkan penanganan berbagai jenis atau entitas bisnis secara homogen.

### **2.3.2 Pengertian *Codeigniter***

CodeIgniter merupakan aplikasi sumber terbuka yang berupa framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP. CodeIgniter memudahkan developer untuk membuat aplikasi web dengan cepat mudah dibandingkan dengan membuatnya dari awal. CodeIgniter dirilis pertama kali pada 28 Februari 2006. Versi stabil terakhir adalah versi 3.0.3.1.2 CodeIgniter Tujuan dari pembuatan framework Codeigniter ini menurut user manualnya adalah untuk menghasilkan framework yang akan dapat digunakan untuk pengembangan proyek pembuatan website secara lebih cepat dibandingkan dengan pembuatan website dengan cara koding secara manual, dengan menyediakan banyak sekali pustaka yang dibutuhkan dalam pembuatan website, dengan antarmuka yang sederhana dan struktur logika untuk mengakses pustaka yang dibutuhkan. CodeIgniter membiarkan kita untuk memfokuskan diri pada pembuatan website dengan meminimalkan pembuatan kode untuk berbagai tujuan pembuatan website. Seperti sudah disebutkan penulis di

atas, bahwa CodeIgniter menerapkan lingkungan pengembangan dengan metode MVC (Model View Controller).

MVC memisahkan antara logika pembuatan kode dengan pembuatan template atau tampilan website. Penggunaan MVC membuat pembuatan sebuah proyek website menjadi lebih terstruktur dan lebih sederhana. Secara sederhana konsep MVC terdiri dari tiga bagian yaitu bagian Model, bagian View dan bagian Controller. Didalam website dinamis setidaknya terdiri dari 3 hal yang paling pokok, yaitu basis data, logika aplikasi dan cara menampilkan halaman website. 3 hal tersebut direpresentasikan dengan MVC yaitu model untuk basis data, view untuk cara menampilkan halaman website dan controller untuk logika aplikasi.

## **2.4 Teori pendukung lainnya**

Teori pendukung lainnya yakni meliputi Basis Data (*database*), HTML (*Hyper Text markup Language*), PHP, MySQL dan Visual Studio.

### **2.4.1 Basis Data (*database*)**

#### **1. Pengertian Basis Data**

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. [7].

Basis dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data merupakan representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep, keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Basis data merupakan kumpulan data yang saling berhubungan. Relasinya biasanya biasanya ditunjukkan dengan kunci dari tiap *file* yang ada. Dalam satu *file* terdapat *record-record* yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu *record* terdiri dari *field-field* yang saling

berhubungan dan menunjukkan dalam satu pengertian yang lengkap dalam satu *record* [8].

Basis data (*database*) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Basis data dimaksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbaiss berkas (Abdul Kadir, 2014).

#### 2.4.2 HTML (*Hyper Text markup Language*)

*HTML (Hyper Text markup Language)* merupakan bahasa yang digunakan untuk mendeskripsikan struktur sebuah halaman *web*. *HTML* berfungsi untuk mempublikasi dokumen online. *Statement* dasar dari mempubliasi dokumen *online*. *Statement* dasar dari *HTML* disebut *tags*. Sebuah *tag* dinyatakan dalam sebuah kurung siku (<>). *Tags* yang ditujukan untuk sebuah dokumen atay bagian dari suatu doumen haruslah dibuat berupa pasangan. Terdiri dari *tag* pembuka dan *tag* penutup. Dimana *tag* penutup menggunakan tambahan tanda garis miring (/) di awal nama *tag* (Pahlevi et al., 2018). Dokumen *HTML* disusun oleh elemen-elemen. Elemen merupakan istilah bagi komponen-komponen dasar pembentuk dokumen *HTML*. Bberapa *HTML* contoh adalah: *head*, *body*, *table*, *paragraf*, dan *list*.

Untuk menandai berbagai elemen dalam suatu dokumen *HTML*, dapat menggunakan tag. Tag *HTML* terdiri atas sebuah kurung sudut kiri (<, Tanda lebih kecil), sebuah nama *tag*, dan sebuah kurung sudut kanan (>, tanda lebih besar). *Tag* umumnya berpasangan (misalnya <H1> dengan </H1>, tag yang berpasangan selalu diawali dengan karakter garis miring (/). Tag-tag yang pertama menunjukkan *tag* awal yang berarti awal elemen, dan yang kedua menunjukkan tag akhir, berarti akhir elemen.

Elemen yang dibutuhkan untuk membuat suatu dokumen HTML dinyatakan dengan tag <html>, <head>, dan <Body> berikut *tag-tag* pasangannya. Setiap dokumen terdiri atas tag *head* dan *body*. Elemen

*head* berisi informasi tentang dokumen tersebut, dan elemen *body* berisi tentang teks yang sebenarnya yang tersusun dari *link*, *grafik*, paragraf dan elemen lainnya [9].

### 2.4.3 Pengertian *PHP*

*PHP* adalah bahasa pemrograman untuk dijalankan melalui halaman *web*, umumnya digunakan untuk mengolah informasi di internet. Sedangkan dalam pengertian lain *PHP* adalah singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *webserver side* yang bersifat *open source* atau gratis. *PHP* merupakan script yang menyatu dengan *HTML* dan berada pada *server* [10].Keunggulan *PHP*

Seluruh aplikasi berbasis *web* dapat dibuat dengan *PHP*. Namun kekuatan yang paling utama *PHP* adalah pada konektivitasnya dengan sistem *database* didalam *web*. Kelebihan-kelebihan dari *PHP* diantaranya adalah:

- a. *PHP* mudah dibuat dan dijalankan, maksudnya *PHP* dapat berjalan dalam *web Server* dan dalam sistem operasi yang berbeda pula.
- b. *PHP* adalah *software open-source* yang gratis dan bebas didistribusikan kembali di bawah lisensi *GPL (GNU Public License)*. User dapat mendownload kode-kode *PHP* tanpa harus mengeluarkan uang atau khawatir dituntut oleh pihak pencipta *PHP*.
- c. *PHP* bisa dioperasikan pada platform *Linux* ataupun *Windows*.
- d. *PHP* sangat efisien, karena *PHP* hanya memerlukan *resource system* yang sangat sedikit dibanding dengan bahasa pemrograman lain.
- e. Ada banyak *Web Server* yang mendukung *PHP*, seperti *Apache*, *PWS*, *IIS*, dan lain-lain.
- f. *PHP* juga didukung oleh *database*, seperti *MySQL*, *PostgreSQL*, *interbase*, *SQL*, dan lain-lain.
- g. Bahasa pemrograman *PHP* sintaknya sederhana, singkat dan mudah dipahami.

- h. *HTML-embedded*, artinya PHP adalah bahasa yang dapat ditulis dengan menempelkan pada sintak-sintak HTML.

#### 2.4.4 Pengertian *MySQL*

*MySQL* adalah salah satu jenis database yang banyak digunakan untuk membuat aplikasi berbasis web yang dinamis. *MySQL* termasuk jenis *RDBMS (Relational Database Management Sistem)*. *MySQL* ini mendukung Bahasa pemrograman *PHP*. *MySQL* juga mempunyai query atau bahasa *SQL (Structured Query Language)* yang simple dan menggunakan escape character yang sama dengan *PHP* (Pahlevi et al., 2018)

##### 1. Keunggulan *MySQL*

Beberapa keunggulan dari *MySQL* yaitu :

- a. Cepat, handal dan mudah dalam penggunaannya *MySQL* lebih cepat tiga sampai empat kali dari pada *database server* komersial yang beredar saat ini, mudah diatur dan tidak memerlukan seseorang yang ahli untuk mengatur administrasi pemasangan *MySQL*.
- b. Didukung oleh berbagai bahasa *Database server MySQL* dapat memberikan pesan error dalam berbagai bahasa seperti Belanda, Portugis, Spanyol, Inggris, Perancis, Jerman dan Italia.
- c. Mampu membuat tabel berukuran sangat besar 23.
- d. Ukuran maksimal dari setiap tabel yang dapat dibuat dengan *MySQL* adalah 4 GB sampai dengan ukuran file yang dapat ditangani oleh sistem operasi yang dipakai.
- e. Lebih murah *MySQL* bersifat *open source* dan didistribusikan dengan gratis tanpa biaya untuk *UNIX platform*, *OS/2* dan *Windows platform*.

#### 2.4.5 *Sublime Text*

##### 1. Pengertian *Sublime Text*

*Sublime Text* adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai *platform operating system* dengan

menggunakan teknologi *python* API. Terciptanya aplikasi ini terinspirasi dari aplikasi *Vim*, Aplikasi ini sangatlah fleksibel dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*. *Sublime Text* bukanlah aplikasi *open source* dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (*packages*) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki lisensi aplikasi gratis.

*Sublime Text* mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur *syntax highlight* hampir di semua bahasa pemrograman yang didukung ataupun dikembangkan oleh komunitas seperti; *C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, Javascript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile* dan *XML*. Biasanya bagi bahasa pemrograman yang didukung ataupun belum terdukug secara *default* dapat lebih dimaksimalkan atau di dukung dengan menggunakan *add-ons* yang bisa di download sesuai kebutuhan *user*

## **2.5 Teori yang berhubungan dengan teknik analisa yang digunakan**

Teori yang berhubungan dengan teknik analisa yang digunakan yakni meliputi UML.

### **2.5.1 UML (*Unified Manipulation Language*)**

*Unified Modeling Language (UML)* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. *UML* merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem” (Windu dan Grace, 2013). *Unified Modeling Language (UML)* adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan

pendokumentasian dari sebuah sistem pengembangan software berbasis OO (Object-Oriented). UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software (<http://www.omg.org>). Diagram Unified Modelling Language [11].

*Unified Manipulation Language* (UML) salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rossa dan Shalahuddin, 2013).

Berikut dibawah ni beberapa diagram *Unified Manipulation Language* (UML).

#### 1. Diagram *Use Case* (*Use Case Diagram*)

*Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case diagram* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat (Rossa dan Shalahuddin, 2013). Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada suatu hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

##### a. Aktor

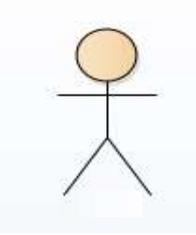
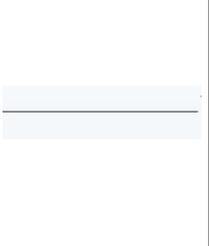
Merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang.

##### b. *Use case*

Merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut dibawah ini simbol dan penejelasan yang ada pada diagram *use case*.

Tabel 2.2 Simbol - simbol *Use Case Diagram*

No	Simbol	Nama	Deskripsi
1.		<i>UseCase</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar proses antar unit atau aktor. Biasanya
2.		Aktor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang , tapi aktornya belum tentu merupakan orang, biasanya dinyatakan mengguankan kata benda diawal frase aktor.
3.		Asosiasi / <i>Association</i>	Komunikasi antara aktor dan <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.		Ekstensi / <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.		Generalisasi / <i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih dari umum dari lainnya.

No	Simbol	Nama	.Deskripsi
6.		<i>Include</i>	Relasi <i>use case</i> tambahan 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. Asa dua sudut pandang yang cukup besar mengenai include di <i>use case</i> .

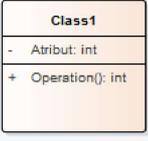
## 2. Diagram Kelas (*Class Diagram*)

*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem [12]. Kelas memiliki apa yang disebut atribut dan metode atau operasi :

- 1) Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
- 2) Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh kelas.

Berikut dibawah ini adalah simbol-simbol pada *Class Diagram* :

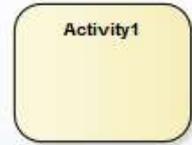
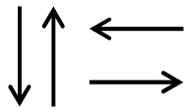
Tabel 2.3 Simbol - simbol *Class Diagram*

No	Simbol	Nama	Deskripsi
1.		<i>Class</i>	Kelas pada struktur sistem.
2.		Antarmuka / <i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.		Asosiasi / <i>association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.		Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus).
5.		Kebergantungan / <i>dependency</i>	Relasi antarkelas dengan makna kebergantungan antarkelas.

### 3. *Activity diagram*

*Activity diagram* atau disebut diagram aktivitas menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. [13].

Tabel 2.4 Simbol - simbol *Activity diagram*

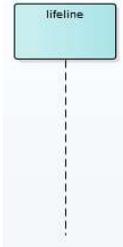
No	Simbol	Nama	Deskripsi
1.		<i>Activity/ Aktivitas</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2.		State	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial node/Status awal</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity final node/ Status Akhir</i>	Objek dibentuk untuk diakhiri.
5		<i>Decesion</i>	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu.
6		<i>Line connector</i>	Digunakan untuk menghubungkan suatu simbol dengan simbol lainnya.

#### 4. Sequence Diagram

*Sequence Diagram* atau diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek [14].

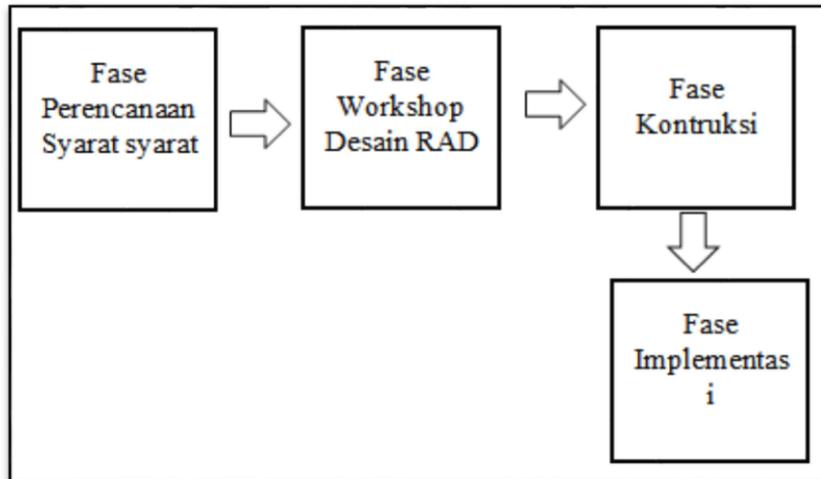
Tabel 2.5 Simbol - simbol *Sequence Diagram*

No	Simbol	Nama	Deskripsi
1.		Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat jalan itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2.		Garis hidup <i>lifeline</i>	Menyatakan kehidupan suatu objek.
3.		Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi
4		Pesan tipe create	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.

No	Symbol	Nama	Deskripsi
6.	<p>1 : keluaran</p> 	Pesan tipe return	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
7		Objek	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
8	<p>1 : masukan</p> 	Pesan tipe send	Menyatakan bahwa suatu objek mengirimkan data atau masukan atau informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.

## 2.6 Metode pengembangan sistem

Pemengembang sistem informasi reservasi tiket travel yang akan dilakukan pada peneliti ini yaitu menggunakan metode *Rapid And Development*



Gambar 2.1 Metode RAD (*Rapid Application Development*) (kendal ,2003)

RAD adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat perangkat lunak.[15].

Perancangan sistem informasi reservasi tiket travel berbasis web ini menggunakan metode *Rapid Application Development (RAD)*, terdapat empat fase yang ada pada metode RAD dan pada tahap penilaian melibatkan penganalisis dan pengguna Empat Fase tersebut yaitu :

a. *Requirements Planing* (Perencanaan syarat syarat).

Fase ini merupakan fase pertemuan antara penganalisis dan pengguna untuk mengidentifikasi tujuan dari sistem yang akan di bangun serta mengidentifikasi syarat syarat informasi yang akan timbul untuk mencapai tujuan tersebut serta menganalisa semua sistem yang dibutuhkan oleh pengguna. Fase ini berorientasi pada penyelesaian masalah yang ada di Smk Ma'arif NU Sirampog.

b. *Design Workshop (Whorkshop Desain RAD)*

Fase ini merupakan fase dalam bentuk workshop *desain RAD* antara penganalisis dan pemrograman untuk merancang sebuah sistem yang akan di bangun. Penganalisis dan pemrogram saling bekerja sama dalam membangun sistem dan menunjukkan representasinya dalam bentuk visual desan dan pola kerjanya kepada pengguna sistem. Pada fase ini juga pengguna merespon *prototype* yang telah dirancang. Penganalisis dan pemrogram dapat memperbaiki serta menganalisis modul modul yang dirancang berdasarkan dari respon pengguna sistem.

c. *Fase Inrtruction (Kontruksi)*

Fase kontruksi merupakan fase eksekusi dalam pembuatan script program dan merupakan kelanjutan dari fase kedua. Pada fase ini juga menunjukkan *platform , hardware , dan software* yang digunakan. Setiap desain yang dibuat pada fase sebelumnya, akan di tingkatkan dengan menggunakan perangkat *RAD*. Setelah fungsi baru tersedia, fungsi baru tersebut ditunjukkan kepada pengguna untuk mendapatkan interaksi dan revisi selanjutnya penganalisis akan melakukan perubahan dalam setiap desain aplikasi berdasarkan intrsuksi pengguna.

d. *Fase implementation (implementasi)*

Fase ini penganalisis bekerja sama dengan para pengguna secara intensif selama workshop berlangsung, dan merancang beberapa aspek dan nonteknis yang dibutuhkan. Segera setelah aspek aspek ini disetujui dan sistem sistem dibangun dan di saring, sistem sistem baru atau bagian dari sistem ujicoba dan kemudian diperkenalkan kepada organisasi. [16].

## **2.7 Pengujian Perangkat Lunak**

Pengujian perangkat lunak adalah proses menjalankan dan mengevaluasi sebuah perangkat lunak secara manual maupun otomatis untuk menguji apakah perangkat lunak sudah memenuhi persyaratan atau belum. [17]. Singkat kata, pengujian adalah aktivitas untuk menemukan dan menentukan perbedaan antara hasil yang diharapkan dengan hasil sebenarnya.

Berikut merupakan penjelasan lebih rinci dari pengujian perangkat lunak baik itu teknik maupun metode yang digunakan didalam pengujian perangkat lunak.

### 2.7.1 Pengujian *blackbox*

Pendekatan *black-box* merupakan pendekatan pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai kebutuhan fungsional yang telah di definisikan . Kasus uji ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya. Teknik pengujian ini berfokus pada domain informasi dari perangkat lunak, yaitu melakukan kasus uji dengan mempartisi domain input dan output program. Metode *black-box* memungkinkan rekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian ini berusaha menemukan kesalahan dalam kategori fungsi-fungsi yang tidak benar atau hilang, kesalahan *interface*, kesalahan dalam struktur data akses basis data eksternal, kesalahan kerja, inisialisasi dan kesalahan terminal (Sulistyanto & SN, 2014).

Pengujian *black-box* berfokus pada persyaratan fungsional peraanngkat lunak. Dengan demikian, pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi-fungsi yang tidak benar atau ilang
2. Fungsi-fungsi yang tidak benar atau hilang
3. Kesalahan *interface*
4. Kesalahan dalam struktur data atau akses database eksternal
5. Kesalahan kerja
6. Inisialisasi dalam kesalahan terminasi

Dengan mengaplikasikan teknik *black-box*, maka user menarik serangkaian *test case* yang memenuhi criteria berikut ini:

1. *Test case* yang mengurangi, dengan harga lebih dari satu, jumlah test case tambahan yang harus di desain untuk mencapai pengujian yang dapat di pertanggung jawabkan.
2. *Test case* memberitahu kesalahan yang berhubungan hanya dengan pengujian spesifik yang ada.

### **2.7.1 Pengujian UAT**

Pengujian UAT adalah suatu proses pengujian oleh pengguna yang dimaksudkan untuk menghasilkan dokumen yang dijadikan bukti bahwasistem yang dikembangkan dapat diterima atau tidaknya oleh pengguna,apabila hasil pengujian sudah bisa dianggap memenuhi kebutuhan dari pengguna maka aplikasi dapat diterapkan. Pengujian dengan UATdilakukan dengan mengajukan beberapa pertanyaan terhadap pegawai dan anggota perpustakaan yang bertindak sebagai pengguna, pengujian ini melibatkan 10 pegawai perpustakaan dan 10 anggota perpustakaan. Hasil user acceptance test dinilai dengan 5 kategori, yaitu SS (Sangat Sesuai), S (Sesuai), C (CUKUP), TS (Tidak Sesuai) dan STS (Sangat tidak sesuai). Berikut ini rincian hasilnya. [18].

Menurut Perry, William E, User Acceptance Testing (UAT) merupakanpengujian yang dilakukan oleh end-user dimana user tersebut adalah staff/karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengankebutuhan/fungsinya. Setelah dilakukan sistem testing, acceptance testingmenyatakan bahwa sistem perangkat lunak memenuhi persyaratan.<sup>11</sup> Setelah dilakukan system testing, acceptance testing menyatakan bahwa sistem software memenuhi persyaratan. Acceptance testing merupakan pengujian yang dilakukan oleh pengguna yang menggunakan teknik pengujian black box untuk menguji sistem terhadap spesifikasinya. Pengguna akhir bertanggung jawab untuk memastikan semua fungsionalitas yang relevan telah diuji.

Pengujian penerimaan pengguna (UAT) adalah fase terakhir dari proses pengujian perangkat lunak. Selama UAT, perangkat lunak diuji untuk memastikan tugas-tugas apakah sudah sesuai dengan spesifikasinya. UAT adalah salah satu prosedur proyek perangkat lunak final dan paling penting yang harus terjadi sebelum perangkat lunak tersebut dikembangkan dan diluncurkan ke pasar. UAT juga dikenal sebagai pengujian beta, pengujian aplikasi atau pengujian pengguna akhir.

Menurut Black, acceptance testing biasanya berusaha menunjukkan bahwa sistem telah memenuhi persyaratan-persyaratan tertentu. Pada pengembangan software dan hardware komersial, acceptance test biasanya disebut juga "alpha tests" (yang dilakukan oleh pengguna in-house) dan "betatests" (yang dilakukan oleh pengguna yang sedang menggunakan atau akan menggunakan sistem tersebut). Alpha dan beta test biasanya juga menunjukkan bahwa produk sudah siap untuk dijual atau dipasarkan. Acceptance testing mencakup data, environment dan skenario yang sama atau hampir sama pada saat live yang biasanya berfokus pada skenario penggunaan produk tertentu.

Pengujian UAT dilakukan untuk mengetahui bahwa software atau sistem yang telah dibuat dapat diterima oleh pengguna dan telah memenuhi kebutuhan sesuai dengan yang diinginkan. Untuk mengetahui tanggapan responden terhadap sistem keuangan yang akan diimplementasikan, maka dilakukan pengujian dengan memberikan 6 pertanyaan kepada 6 responden dimana jawaban dari pertanyaan tersebut terdiri dari tingkatan yang dipilih sebagai berikut :

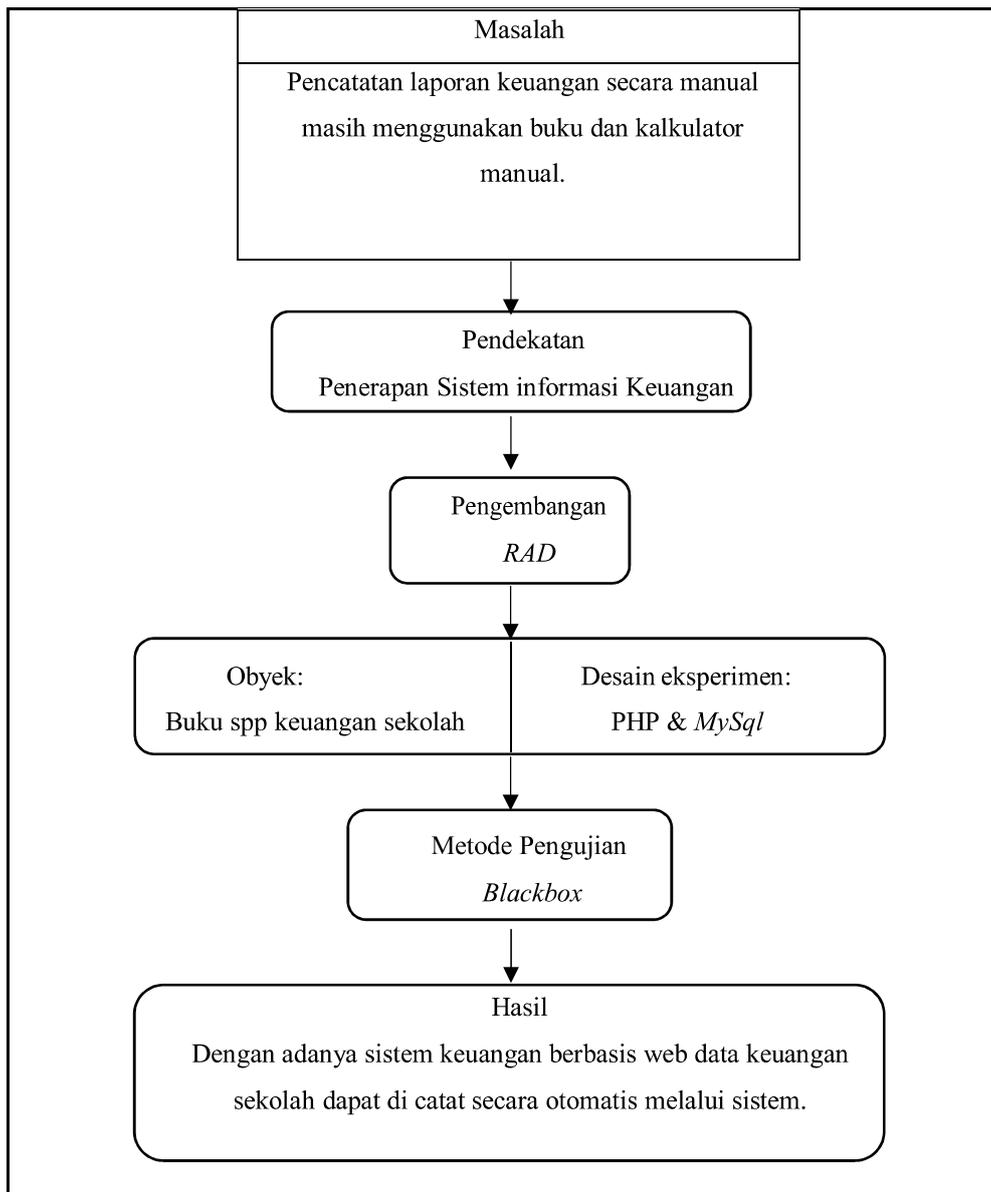
Tabel 4.21 Pilihan jawaban UAT

A	Sangat Bagus
B	Bagus
C	Cukup
D	Tidak Bagus
E	Sangat Tidak Bagus

Tabel 4.22 Bobot Nilai Jawaban UAT

Jawaban	Bobot
A = Sangat Bagus	5
B = Bagus	4
C = Cukup	3
D = Tidak Bagus	2
E = Sangat Tidak Bagus	1

## 2.8 Kerangka berfikir



Gambar 2.2 Kerangka Berfikir