

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Studi Literatur**

Pada penulisan skripsi ini, peneliti menggali informasi dari penelitian-penelitian sebelumnya sebagai bahan perbandingan, baik mengenai metode yang digunakan dan kesimpulan yang ada. Selain itu, peneliti juga menggali dari buku-buku maupun skripsi dalam rangka mendapatkan suatu informasi yang ada sebelumnya tentang teori tentang judul yang digunakan untuk memperoleh landasan teori ilmiah. Beberapa penelitian mengenai pengembangan dashboard analisa keuangan antaranya adalah sebagai berikut:

Penelitian yang dilakukan oleh (Jayanti & Ani, 2017) Pembangunan Dashboard Untuk Visualisasi Analisa Keuangan. Pada penelitian yang dilakukan oleh Elisa Dwi Jayanti dan Nur Ani pada tahun 2017 ini menceritakan sebuah perusahaan PT HFI yang bergerak dibidang pembiayaan alar berat dengan berkembangnya perusahaan semakin banyak dan beragam data yang dimiliki, hal ini menyulitkan pihak perusahaan dalam melakukan analisis terhadap data tersebut, berkaitan dengan hal tersebut maka dirancang dan dibangun sebuah dashboard di PT HFI yang dapat memudahkan analisis terhadap data yang dimiliki oleh perusahaan dan dashboard yang dapat menyajikan informasi kedalam bentuk visual.

Penelitian yang dilakukan oleh (Mahsud & Herni, 2018) Aplikasi Analisis Laporan Keuangan Dengan Metode Analisis Rasio Keuangan Berbasis Web Pada PT. Sinar Galesong Mandiri. Pada penelitian yang dilakukan oleh Mahsud dan Herni melakukan implementasikan sebuah metode analisis rasio keuangan berbasis web, studi kasus penelitian ini pada PT. Sinar Galesong Mandiri dengan adanya aplikasi Analisis Laporan Keuangan ini diharapkan dapat membantu perusahaan untuk mengetahui posisi keuangan perusahaan dalam astu periode, mengetahui keberhasilan dan kelemahan-kelemahan perusahaan serta dapat mengetahui langkah-langkah perbaikan kinerja manajemen perusahaan.

Penelitian yang dilakukan oleh (Leon, 2019) Analisis Laporan Keuangna Usaha Mikro Kecil Dan Menengah (UMKM) Industri Tas Dan Sepatu New Hunteria Dengan Pendekatan Standar Akuntansi Keuangan Entitas Mikro Kecil

Dan Menengah (SAKEMKM). Pada penelitian ini diceritakan bahwa laporan keuangan yang dibuat belum sesuai dengan standar akuntansi (SAK EMKM). Pada Industri Tas Dan Sepatu Hunteria Baru hanya membuat pencatatan keuangan berupa pencatatan pembelian bahan baku, penjualan, gaji karyawan, dan biaya lainnya.

Tabel 2.1 Penelitian Terkait

No	Penulis	Judul	Keterangan
1	Elisa Dwi Jayanti, Nur Ani (2017), Universitas Mercu Buana	Perancangan Dashboard Untuk Visualisasi Analisa Keuangan	<ol style="list-style-type: none"> <li>1. Metode pengumpulan data yang dilakukan melalui proses wawancara, studi pustaka, dan observasi.</li> <li>2. Tujuan penelitian yang dilakukan membangun data warehouse untuk mempermudah proses analisis data.</li> <li>3. Hasil dari penelitian ini adalah data warehouse yang dapat memudahkan analisis terhadap data yang dimiliki oleh perusahaan dan dashboard yang dapat menyajikan informasi kedalam bentuk visual.</li> </ol>
2	Mashud, Heni (2018), STMIK AKBA	Aplikasi Analisis Laporan Keuangan Dengan Metode Analisis Rasio Keuangan Berbasis Web Pada Pt.Sinar Galesong Mandiri	<ol style="list-style-type: none"> <li>1. Mengimplementasikan aplikasi analisis laporan keuangan dengan metode analisis rasio keuangan berbasis website.</li> <li>2. Analisis rasio keuangan yang digunakan: rasio likuiditas, rasio</li> </ol>

No	Penulis	Judul	Keterangan
			<p>solvabilitas, rasio aktivitas atau rasio profitabilitas.</p> <p>3. Studi kasus pada penelitian PT. Sinar Galesong Mandiri menggunakan metode observasi dan studi literatur dan menggunakan desain Model sistem pada penelitian ini menggunakan UML.</p>
3	Misheley Leon (2019), Universitas Krisnadwipayama	Analisis Laporan Keuangan Usaha Mikro Kecil Dan Menengah Industri Tas Dan Sepatu New Hunteria Dengan Pendekatan Standar Akuntansni Keuangan Entitas Mikro Kecil Dan Menengah (SAKEMKM)	<p>1. Pada penelitian ini menggunakan menggunakan penelitian kualitatif dengan tipe deskriptif.</p> <p>2. Pengumpulan data melalui wawancara dan pengamatan.</p>

Tabel 2.2 Penelitian Yang Dilakukan Penulis

No	Penulis	Judul	Keterangan
1	Yoko Bertu (2022), Universitas Tanjungpura Pontianak	Pengembangan Dashboar Analisa Keuangan Pada Usaha Mikro Kecil Dan Menengah (UMKM) Di Kota Pontianak	<p>1. Aplikasi berbasis <i>website</i>.</p> <p>2. Mengimplementasikan Laporan Keuangan Laba Rugi, Laporan Perubahan</p>

No	Penulis	Judul	Keterangan
			Modal, Laporan Neraca, Laporan Arus Kas. 3. Mengimplementasikan Analisis Keuangan Horizontal. 4. Menampilkan grafik pemasukan dan pengeluaran perbulan.

## 2.2 Laporan Keuangan

Laporan keuangan merupakan hasil akhir dari proses pencatatan transaksi keuangan suatu perusahaan yang menunjukkan kondisi keuangan perusahaan yang menunjukkan kondisi keuangan suatu perusahaan pada satu periode akuntansi dan merupakan gambaran umum mengenai kinerja suatu perusahaan. Laporan keuangan merupakan produk akhir proses akuntansi suatu perusahaan dalam satu periode tertentu dimana informasi dari laporan keuangan merupakan hasil pengumpulan dan pengolahan data keuangan, dengan tujuan untuk membantu perusahaan membuat keputusan yang tepat. Proses dalam penyusunan laporan keuangan menggunakan berbagai data, seperti faktur, bon, nota, kredit, laporan, bank, dan lainnya (Munawir, 2010).

Laporan keuangan terdiri dari empat macam, yaitu:

### 1. Laporan Laba Rugi

Laporan Laba Rugi adalah bagian dari laporan keuangan suatu perusahaan yang dihasilkan pada suatu periode akuntansi yang menjabarkan unsur-unsur pendapatan dan beban perusahaan sehingga menghasilkan suatu laba bersih.

### 2. Laporan Perubahan Modal

Laporan perubahan modal adalah laporan keuangan yang menyajikan perubahan modal dalam suatu periode. Laporan perubahan modal dibuat setelah membuat laporan laba rugi dan sebelum membuat neraca, dikarenakan laba bersih suatu periode harus dimasukkan dalam laporan perubahan modal,

laporan perubahan modal menjadi penghubung antara laporan laba rugi dan neraca.

### 3. Laporan Neraca

Laporan neraca atau *balance sheet* adalah laporan berbentuk sistematis yang berisi tentang aktiva, hutang, dan modal dari suatu perusahaan dalam suatu periode.

### 4. Laporan Arus Kas

Laporan arus kas atau *cash flow* sebuah perincian yang menunjukkan jumlah pemasukan dan pengeluaran dalam suatu periode tertentu.

## 2.3 Analisa Keuangan

Analisa keuangan kemampuan yang digunakan untuk mengevaluasi kelangsungan bisnis, stabilitas, profitabilitas bisnis, sub-bisnis atau proyek. Analisa keuangan dilakukan oleh seorang yang menggunakan informasi yang ditampilkan dalam laporan keuangan untuk menampilkan laporan dalam bentuk rasio.

Analisa laporan keuangan adalah menguraikan pos laporan keuangan menjadi informasi yang lebih kecil sehingga dapat dipahami dengan tujuan mengetahui kondisi keuangan dalam proses pengambilan keputusan.

Analisa laporan keuangan merupakan alat analisis bagi manajemen keuangan perusahaan yang bersifat menyeluruh, dapat digunakan untuk mendeteksi tingkat kesehatan perusahaan, melalui analisis kondisi arus kas atau kinerja organisasi perusahaan baik yang bersifat parsial maupun kinerja organisasi secara keseluruhan.

## 2.4 Analisis *Horizontal*

Analisis *Horizontal* adalah suatu analisis persentase kenaikan ataupun penurunan yang terdapat di dalam berbagai pos akun keuangan yang komparatif. Berdasarkan analisis horizontal, kita bisa mengetahui jumlah pada setiap pos laporan tahun terakhir bila dibandingkan dengan pos yang ada di dalam laporan keuangan sebelumnya, tujuan dari analisis laporan keuangan yaitu :

1. Menilai kinerja bisnis pada tahun berjalan.
2. Mengetahui kemampuan perusahaan dalam memanfaatkan aset ke arah yang lebih menguntungkan.

3. Mengetahui perubahan posisi keuangan pada periode tertentu.

Analisis Horizontal adalah metode analisis dengan mengadakan perbandingan laporan keuangan yang sama pada dua periode yang berbeda sehingga akan diketahui perkembangannya (Sujarweni, 2017).

Laporan Posisi Keuangan (Neraca) Unilever - (Dalam jutaan rupiah)				
Akun	2016	2015	Peningkatan	(Penurunan)
			Jumlah	Persentase
<b>Aset</b>				
Aset Lancar	6.588.109	6.623.114	(35.005)	-0,53%
Aset Tetap	10.157.586	9.106.831	1.050.755	11,54%
Aset Total	16.745.695	15.729.946	1.015.749	6,46%
<b>Kewajiban</b>				
Kewajiban Lancar	10.878.074	10.127.542	750.532	7,41%
Kewajiban Jangka Panjang	1.163.363	775.043	388.320	50,10%
Total Kewajiban	12.041.437	10.902.585	1.138.852	10,45%
<b>Ekuitas</b>				
Modal Saham	76.300	76.300	-	0,00%
Tambahan Modal Disetor	96.000	96.000	-	0,00%
Saldo Laba Dicadangkan	15.260	15.260	-	0,00%
Saldo Laba Belum Dicadangkan	4.516.698	4.639.800	(123.102)	-2,65%
Total Ekuitas	4.704.258	4.827.360	(123.102)	-2,55%
Total Kewajiban dan Ekuitas	16.745.695	15.729.945	1.015.750	6,46%

Gambar 2.1 Analisis Horizontal Laporan Neraca

Pada laporan posisi keuangan atau Neraca menunjukkan bahwa nilai aset lancar perusahaan turun sebesar -0,53%. Namun aset total Unilever mengalami kenaikan sebesar 6,46%. Kenaikan aset total ini disebabkan kenaikan aset tetap yang mencapai 11,54%.

Laporan Laba Rugi Unilever - (Dalam jutaan rupiah)				
Akun	2016	2015	Peningkatan	(Penurunan)
			Jumlah	Persentase
Penjualan Bersih	40.053.732	36.484.030	3.569.702	9,8%
Harga Pokok Penjualan	19.594.636	17.835.061	1.759.575	9,9%
Laba Kotor	20.459.096	18.648.969	1.810.127	9,7%
Beban Penjualan	7.791.556	7.239.165	552.391	7,6%
Beban Umum dan Administrasi	3.960.830	3.465.924	494.906	14,3%
Penghasilan / Beban lain-lain	951	(4.479)	5.430	121,2%
Total Beban Usaha	11.753.337	10.700.610	1.052.727	9,8%
Laba Usaha	8.707.661	7.939.401	768.260	9,7%
Pendapatan Keuangan	7.468	10.616	(3.148)	-29,7%
Beban Keuangan	142.344	120.527	21.817	18,1%
Laba Sebelum Pajak	8.571.885	7.829.490	742.395	9,5%
Pajak Penghasilan	2.181.213	1.977.685	203.528	10,3%
Laba Bersih	6.390.672	5.851.805	538.867	9,2%

Gambar 2.2 Anlisis Horizontal Laporan Laba Rugi

Pada gambar Analisis Horizontal Laba Rugi Unilever Diatas dapat melihat apakah pada tahun 2016 mengalami peningkatan atau penurunan dibandingkan dengan tahun 2015 serta dapat melihat persentase.

## 2.5 Dashboard

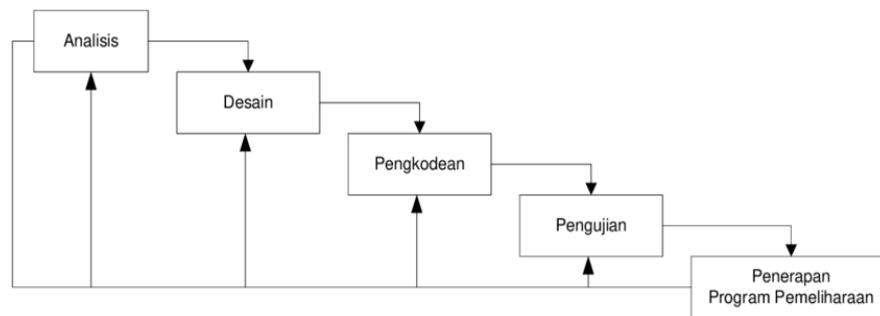
*Dashboard* adalah sebuah visualisasi yang menampilkan bagan, grafik, laporan, indikator visual, dan mekanisme *alert*, yang akan dikonsolidasikan kedalam *platform* informasi yang dinamis dan relevan. Dashboard berperan sebagai *live console* untuk mengelola inisiatif bisnis (Maulida et al., 2020). Manfaat dari *dashboard* visualisasi keuangan adalah untuk melakukan *monitoring* dalam pengukuran kinerja UMKM untuk mencapai target yang diinginkan, sehingga data dan informasi dapat dikelola dan disajikan dengan efektif dan efisien. Hal ini memudahkan pengelola UMKM dalam melakukan analisa terhadap pendapatan dan kinerjanya.

## 2.6 Waterfall

Model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Model ini disebut juga dengan “*Classic Life Cycle*” atau metode waterfall. Model ini termasuk ke dalam model generic pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winson Royce sekitar tahun

1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan (Pressman, 2015).

Fase-fase dalam *Waterfall Model* menurut referensi Pressman:



Gambar 2.3 Model *Waterfall*

Penjelasan tahapan *waterfall* tersebut yaitu:

#### 1. Analisis

Fase ini merupakan proses analisis terhadap aplikasi yang sedang berjalan dengan tujuan untuk mendapatkan kebutuhan sesuai dengan pengguna mengenai pengguna sistem, cara kerja sistem dan waktu penggunaan sistem, sehingga kebutuhan yang diperlukan untuk sistem baru akan didapatkan.

#### 2. Desain

Desain merupakan proses menentukan cara kerja aplikasi seperti perancangan antarmuka, *database*, dan perancangan alur program. Perancangan diperlukan untuk menggambarkan sistem baru untuk memenuhi kebutuhan *user*.

#### 3. Pengkodean

Tahap pengkodean yaitu tahap rancangan sistem yang dibentuk menjadi suatu kode program untuk pembuatan aplikasi.

#### 4. Pengujian

Pada tahap pengujian untuk mengetahui kesesuaian aplikasi apakah berjalan sesuai dengan kebutuhan dan memastikan sistem terhindar dari *error* terjadi. Pengujian juga dilakukan untuk memastikan kevalidan dalam proses *input* sehingga dapat menghasilkan *output* yang sesuai.

## 5. Penerapan Program Pemeliharaan

Pada fase penerapan program dan pemeliharaan sistem yang berguna untuk melihat kemampuan aplikasi, dan mengecek jika masih ada ditemukan *error* atau ada penambahan fitur-fitur yang belum ada pada sistem tersebut. Pemeliharaan diperlukan ketika adanya perubahan dari pengguna seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

### 2.7 *Unified Modeling Language* (UML)

*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 merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Sukamto & Salahuddin, 2014).

*Unified Modelling Language* (UML) adalah bahasa pemodelan untuk sistem atau perangkat lunak. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami (Nugroho, 2010).

Sedangkan menurut Rosa dan Shalahuddin (2014) UML (*Unified Modeling Language*) adalah salah satu standar bahasa visual yang banyak digunakan di dunia industri untuk mengidentifikasi *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML hanya berfungsi untuk melakukan pemodelan, jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada

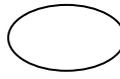
kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

### 2.7.1 Use Case Diagram

*Use Case Diagram* menurut Widodo (2011) Diagram *use case* bersifat statis, yang memperlihatkan himpunan *Use Case* dan aktor-aktor (suatu jenis khusus dari kelas) dan menggambarkan apa saja aktivitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna yang menjadi persoalan itu apa yang dilakukan bukan bagaimana melakukannya. *Use Case Diagram* menggambarkan fungsionalitas yang diharapkan dari sistem.

Deskripsi simbol yang digunakan pada *Use Case* diagram dapat dilihat pada Tabel 2.3 (Ningsih, 2020).

Tabel 2.3 Deskripsi Notasi pada *Use Case* Diagram

No	Gambar	Nama	Keterangan
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
3		Generalization	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4		Association	Yang menghubungkan antara objek satu dengan objek lainnya.

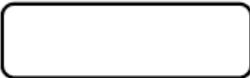
No	Gambar	Nama	Keterangan
5	<pre>&lt;&lt;include&gt;&gt; &lt;-----</pre>	Include	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
6	<pre>&lt;&lt;extend&gt;&gt; -----&gt;</pre>	Extend	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada titik yang diberikan.

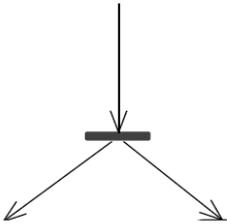
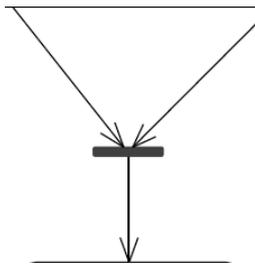
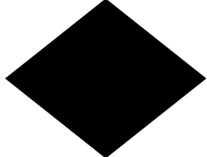
### 2.7.2 Activity Diagram

*Activity* diagram menggambarkan berbagai alir kerja dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Berikut merupakan simbol notasi *activity* diagram pada tabel 2.4 (Ningsih, 2020).

Tabel 2.4 Deskripsi Notasi pada *Activity* Diagram

Simbol	Nama	Keterangan
	Start Point	Start Point diletakan pada pojok kiri atas dan merupakan awal aktivitas.
	End Point	End Point adalah tanda akhir aktivitas.
	Activities	Activities menggambarkan suatu proses/kegiatan.

Simbol	Nama	Keterangan
	Fork/Percabangan	Fork atau percabangan digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu.
	Join/Penggabungan	Join atau penggabungan atau rake digunakan untuk menunjukan adanya dekomposisi.
	Decision Point	Decision Point menggambarkan pilihan untuk pengambilan keputusan, true atau false.

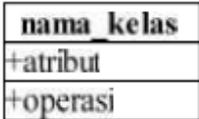
### 2.7.3 Class Diagram

Diagram kelas dibuat agar para pembuat program atau programmer membuat kelas-kelas sesuai dengan rancangan yang ada di dalam diagram kelas sehingga antara dokumentasi perancangan dan perangkat lunak sinkron. Kelas terdiri dari atribut yang mendeskripsikan variabel-variabel yang dimiliki oleh suatu kelas dan operasi atau metode yang menjelaskan tentang fungsi-fungsi yang dimiliki oleh kelas (Hendini, 2016).

Tabel 2.5 Jenis-jenis *Multiplicity*

No	Indikator	Keterangan
1	0 .. 1	Nol atau satu
2	1	Hanya satu
3	0 .. *	Nol atau lebih
4	1 .. *	Satu atau lebih

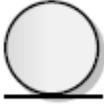
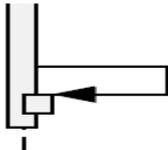
Tabel 2.6 Simbol *Class Diagram*

No	Simbol	Deskripsi
1	Kelas 	Kelas pada struktur sistem.
2	Antarmuka/Interface 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3	Assosiasi/association 	Yang menghubungkan antara objek satu dengan objek lainnya.
4	Asosiasi berarah/directed association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
5	Generalization 	Relasi antar kelas dengan makna generalisasi-spesialisasi
6	Agregasi/aggregation 	Relasi antar kelas dengan makna semua-bagian (whole part).

#### 2.7.4 Sequence Diagram

Menurut (Sukamto & Shalahuddin, 2018), “Diagram *Sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Jadi, untuk menggambar diagram sekuen maka harus diketahui objek – objek yang terlibat dalam sebuah *use case* beserta metode – metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat scenario yang ada pada *use case*”.

Tabel 2.7 Simbol *Sequence Diagram*

Simbol	Nama	Keterangan
	Entity Class	Entity Class merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	Boundary Class	Boundary Class, berisi kumpulan kelas yang menjadi interfaces atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak.
	Control Class	Control Class, merupakan suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab, kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	Message	Message adalah simbol untuk mengirim pesan antar class
	Recursive	Recursive, merupakan menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	Activation	Activation mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.

Simbol	Nama	Keterangan
	Lifeline	Lifeline merupakan garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.

## 2.8 Teknologi Pendukung

### 2.8.1 PHP: *Hypertext Preprocessor*

Menurut (MADCOM, 2016) “PHP (*Hypertext Preprocessor*) adalah bahasa *script* yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk membuat program situs *web* dinamis”. PHP dapat digunakan dengan gratis (*free*) dan bersifat *Open Source*. PHP dirilis dalam lisensi *PHP license*. Untuk membuat program PHP kita diharuskan untuk menginstal *web server* terlebih dahulu. PHP mendukung komentar seperti pada bahasa ‘C’, ‘C++’, dan Unix shell-style. (*Perl style*).

### 2.8.2 XAMPP

Menurut (MADCOM, 2016) “Xampp adalah sebuah paket kumpulan software yang terdiri dari Apache, MySQL, PhpMyAdmin, PHP, Perl, Filezilla, dan lain.” Xampp berfungsi untuk memudahkan instalasi lingkungan PHP, di mana biasanya lingkungan pengembangan *web* memerlukan PHP, Apache, MySQL dan PhpMyAdmin.

### 2.8.3 *JavaScript*

*JavaScript* adalah bahasa *script* yang disisipkan pada kode HTML dan diproses di sisi klien yaitu sering disebut *client side*. Bahasa ini menjadikan dokumen HTML menjadi semakin luas (Agusriandi, 2018).

### 2.8.4 HTML

Menurut (Wardana, 2016). *Hypertext markup language* (HTML) merupakan bahasa pemrograman dasar untuk mengelola website. Akan tetapi HTML hanya terbatas pada pembuatan website statis (website yang tidak dapat

berinteraksi aktif dengan *user*). Maka dari itu HTML biasa dikombinasikan dengan Bahasa pemrograman web lainnya

### **2.8.5 CSS**

CSS adalah suatu Bahasa pemrograman web yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan web akan lebih rapi, terstruktur, dan seragam. CSS merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan atau *layout* halaman web agar lebih menarik. CSS adalah sebuah teknologi internet yang direkomendasikan oleh *World Wide Web Consortium* atau W3C pada tahun 1996 (Wahyudi, 2017).

### **2.8.6 Laravel**

Laravel merupakan *framework* bahasa pemrograman PHP yang memiliki banyak fitur dan sangat membantu *developer* dalam membangun sebuah aplikasi berbasis web (Tahir et al., 2019). Laravel adalah sebuah web *development framework* yang didesain untuk meningkatkan kualitas aplikasi dengan mengurangi beban biaya pengembangan dan memudahkan proses maintenance serta meningkatkan produktifitas pekerjaan dengan kode program yang rapi dan terstruktur (Luthfi, 2017). Laravel memiliki beberapa kelebihan diantaranya: menggunakan *Command Line Interface (CLI) Artisan*, dapat menggunakan *package manager PHP Composer*, Penulisan kode program yang rapi, singkat dan terstruktur, dan mudah dimengerti *developer*

## **2.9 Pengujian Aplikasi**

Pengujian dapat menjadi alat ukur kualitas perangkat lunak yang dibutuhkan sebelum sebuah perangkat lunak tersebut dijalankan. Hal tersebut perlu dilakukan untuk membuktikan bahwa perangkat lunak tersebut sudah layak untuk digunakan dan memenuhi kebutuhan kinerja sesuai kebutuhan dari pengguna.

### **2.10 Pengujian User Acceptance Test (UAT)**

*User Acceptance Test (UAT)* atau uji penerimaan pengguna adalah suatu proses pengujian oleh pengguna yang untuk menghasilkan dokumen yang dijadikan bukti bahwa perangkat lunak yang dikembangkan dapat diterima oleh pengguna.

Pengujian UAT dilakukan oleh pengguna target dari aplikasi yang dibangun untuk mengetahui aplikasi berjalan sesuai dengan proses yang telah ditentukan bertujuan untuk memeriksa *bug* yang ditemukan sebelum aplikasi dirilis. Pengujian UAT dapat dinilai oleh pengguna aplikasi, dengan menggunakan kuisioner sesuai *test case* yang diinginkan. Pengguna akhir bertanggung jawab untuk memastikan semua fungsionalitas yang relevan telah diuji. Hasil dari UAT adalah dokumen yang menunjukkan bukti pengujian berdasarkan bukti pengujian dapat diambil kesimpulan, apakah *software* yang diuji telah dapat diterima atau tidak.

Tabel 2.8 Pilihan Jawaban UAT

A	Sangat : Mudah/Bagus/Sesuai/Jelas
B	Mudah/Bagus/Sesuai/Jelas
C	Netral
D	Cukup: Sulit/Bagus/Sesuai/Jelas
E	Sangat: Sulit/Bagus/Sesuai/Jelas

Tabel 2.9 Bobot Nilai Jawaban

Jawaban	Bobot
A. Sangat : Mudah/Bagus/Sesuai/Jelas	5
B. Mudah/Bagus/Sesuai/Jelas	4
C. Netral	3
D. Cukup : Sulit/ Bagus/Sesuai/Jelas	2
E. Sangat : Sulit/ Bagus/Sesuai/Jelas	1

Tabel 2.10 Pertanyaan Kuesioner

No	Pertanyaan	A	B	C	D	E
1						
2						
3						
4						
5						

<b>No</b>	<b>Pertanyaan</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>
6						
7						