

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Terdapat beberapa penelitian yang memiliki kolerasi yang searah dengan penelitian yang akan dibahas dalam skripsi ini antara lain:.

Penelitian dari Irwansyah, (2015) dengan judul Pembuatan aplikasi penjualan kue Kering berbasis *web* membahas tentang Aplikasi penjualan kue kering berbasis *web* pada toko kue Winda. Toko kue Winda pada sistem penjualan, promosi, pencatatan laporan penjualan, pencatatan data pelanggan, dan pemesanan secara manual. Sistem promosinya sendiri masih menggunakan *word of mouth* (WOM) dan pemesanannya melalui telepon. Hal yang dibutuhkan oleh Winda adalah suatu sistem aplikasi berbasis *web*, dimana pada sistem *web* tersebut terdapat sistem yang berfungsi untuk menangani proses penjualan, promosi, pencatatan laporan penjualan, pencatatan data pelanggan dan pemesanan.

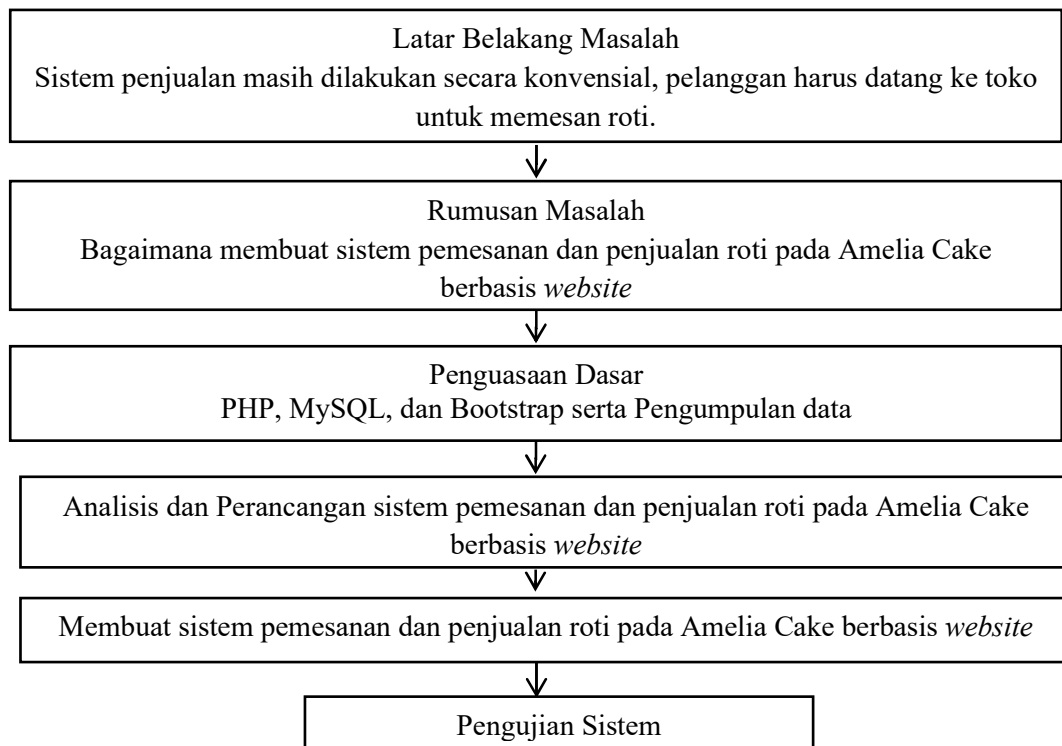
Penelitian Indrawasari, (2019) dengan judul pembuatan toko online Punpun Shop . Punpun Shop merupakan toko yang bergerak di bidang penjualan *bucket*. Toko ini masih menggunakan sistem penjualan memanfaatkan sosial media dalam proses melakukan pemesanan produk dan digunakan menawarkan produknya ke pelanggan. *Software* yang digunakan dalam pembuatan sistem ini yaitu XAMPP, MySQL, PHP, Codeigniter.

Penelitian Ishak, dkk, (2018) dengan judul Rancang Bangun Sistem Informasi Penjualan Kue Dan Roti Berbasis *Web* pada Yuki Bakery Jakarta. Yuki Bakery merupakan toko yang bergerak dalam bidang penjualan Kue dan Roti yang berlokasi di Jalan Joglo Raya, Jakarta Barat. Terdapat beberapa permasalahan pada Yuki Bakery seperti belum maksimalnya dalam penyampaian informasi produk yang ditawarkan dikarenakan kurangnya media penyampaian informasi untuk promosi, Proses transaksi jual-beli berjalan lambat dan terlalu memerlukan waktu yang banyak, dimana pelanggan yang tertarik untuk membeli produk yang diinginkan harus datang

langsung ke toko. Tujuan penelitian ini untuk membangun sistem penjualan roti berbasis *web* sehingga dapat memudahkan promosi barang dan mempercepat proses transaksi jual-beli produk.

2.2 Kerangka Pemikiran

Berikut ini adalah tahapan kerangka pemikiran dalam membuat aplikasi sistem pemesanan dan penjualan pada Amelia Cake dapat disajikan pada Gambar 2.1.



Gambar 2.1. Kerangka Pemikiran

Penjelasan dari kerangka pemikiran tersebut adalah :

1. Latar Belakang Masalah

Sistem penjualan pada Amelia Cake masih menggunakan sistem konvensional. Sistem penjualan yang seperti ini memiliki beberapa kendala yaitu memesan produk harus langsung datang ke toko dan juga ketika konsumen ingin memesan produk di luar jam kerja. Karena itu dibutuhkannya suatu sistem yang

dapat mengatasi permasalahan ruang dan waktu, dimana konsumen dapat memesan produk dari mana saja dan kapan saja.

2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan maka rumusan masalahnya adalah bagaimana membuat sistem pemesanan dan penjualan roti pada Amelia Cake?.

3. Penguasaan Dasar PHP dan MySQL

Penguasaan dasar PHP dan MySQL merupakan suatu kunci terciptanya sistem tersebut, karena pembuatan sistem ini menggunakan bahasa pemrograman PHP dan menggunakan database MySQL sebagai media penyimpan datanya.

4. Analisis dan Perancangan Sistem

Analisis dan perancangan sistem pemesanan dan penjualan roti pada Amelia Cake berbasis *website* bertujuan untuk mengetahui kesiapan data dan rancangan dari aplikasi sistem pemesanan dan penjualan roti pada Amelia Cake yang diharapkan dapat membantu. Perancangan sistem pada penelitian ini menggunakan UML (*Unified Modeling Language*).

5. Rancang Bangun Sistem

Membuat sistem pemesanan dan penjualan roti pada Amelia Cake menggunakan bahasa pemrograman PHP dan database MySQL sebagai media penyimpanan datanya.

6. Pengujian Sistem

Pengujian sistem merupakan tahap setelah sistem berhasil dibuat. Pengujian sistem bertujuan untuk mengetahui kelebihan, kelemahan serta mengetahui kelayakan suatu sistem untuk digunakan. Pengujian sistem pada penelitian ini menggunakan metode *Black Box* dan metode kuisioner.

2.3 Teori – teori Pendukung

2.3.1 E-commerce

E-commerce atau Perdagangan elektronik (bahasa Inggris: *electronic commerce* atau *e-commerce*) adalah penyebaran, pembelian, penjualan, pemasaran barang dan jasa melalui sistem elektronik seperti internet atau televisi, www, atau jaringan komputer lainnya. *E-commerce* dapat melibatkan transfer dana elektronik, pertukaran data elektronik, sistem manajemen inventori otomatis, dan sistem pengumpulan data otomatis (Fauyhi Eko Nugroho, 2016).

2.3.2 Penjualan

Menurut Ginting, (2013), Penjualan adalah penerimaan yang diperoleh dari pengiriman barang dagangan atau dari penyerahan pelayanan dalam bursa sebagai barang pertimbangan. Pertimbangan ini dapat dalam bentuk tunai peralatan kas atau harta lainnya. Pendapatan dapat diperoleh pada saat penjualan, karena terjadi pertukaran, harga jual dapat ditetapkan dan bebannya diketahui.

2.3.3 Sistem

Menurut Khadir, (2014), Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Tujuan utama sistem yang umum ada tiga macam yaitu:

- a. Untuk mendukung fungsi kepengurusan manajemen.
- b. Untuk mendukung pengambilan keputusan manajemen.
- c. Untuk mendukung kegiatan operasi perusahaan.

2.3.4 Website

Website merupakan fasilitas *internet* yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *web page* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu halaman ke

halaman lain (*hyper text*), baik diantara halaman yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. Halaman diakses dan dibaca melalui *browser* seperti *Netscape Navigator*, *Internet Explorer*, *Mozilla Firefox*, *Google Chrome* dan aplikasi *browser* lainnya (Fritz, 2014).

2.3.5 PHP

Menurut Hakim, (2014), PHP merupakan bahasa pemrograman yang bisa digunakan untuk membuat halaman HTML. File .php yang dibuat akan diproses di dalam *server*, sedangkan halaman yang akan dikirim ke *browser* pengunjung hanyalah tampilan HTML-nya. Dengan PHP, halaman *website* yang dibuat akan menjadi dinamis, yakni dapat selalu berubah tanpa harus mengubah isi website secara manual. Informasi akan diproses ulang oleh *web server* sehingga akan didapatkan isi paling mutakhir dari halaman *web*.

2.3.6 Unified Modeling Language (UML)


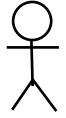

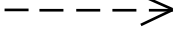
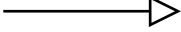
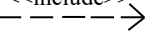
Menurut Sukamto dan Shalahuddin, (2016), dijelaskan pada perkembangan teknologi perangkat lunak diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Pada perkembangan teknik pemrograman berorientasi objek, muncul sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek yaitu *Unified Modelling Language*(UML). Ada 6 (enam) macam diagram dalam *Unified Modeling Language* (UML), yaitu :

2.3.6.1 Use Case Diagram

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol *use case diagram* dapat dilihat pada Tabel 2.1.


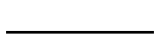
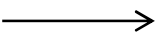
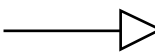
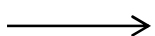
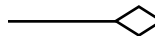
Tabel 2.1 Simbol-simbol *Use Cass Diagram*

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

2.3.6.2 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode operasi. Kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas (Sukamto dan Shalahuddin, 2016). Simbol *class diagram* dapat dilihat pada Tabel 2.2.



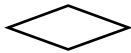


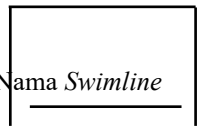
Tabel 2.2 Simbol-simbol *Class Diagram*

NO	SIMBOL	NAMA	KETERANGAN			
1	<table border="1" style="border-collapse: collapse; width: 100px;"> <tr><td>Nama kelas</td></tr> <tr><td>+ Atribut</td></tr> <tr><td>+ Operasi()</td></tr> </table>	Nama kelas	+ Atribut	+ Operasi()	<i>Class</i>	Kelas pada struktur system
Nama kelas						
+ Atribut						
+ Operasi()						
2		<i>Interface</i>	Sama dengan konsep interface dalam pemrograman berorientasi objek.			
3		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .			
4		<i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain biasanya juga disertai dengan <i>multiplicity</i> .			
5		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi - spesialisasi (umum – khusus).			
6		<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.			
7		<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).			

2.3.6.3 Activity Diagram

Activity diagram menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Sukamto dan Shalahuddin, 2016). Simbol *Activity Diagram* ditunjukkan pada Tabel 2.3.

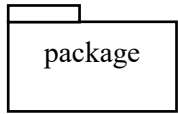
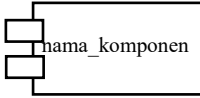
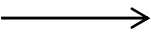


Tabel 2.3 Simbol-simbol *Activity Diagram*

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

2.3.6.4 Component Diagram

Diagram komponen *atau component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem (Sukamto dan Shalahuddin, 2016). Simbol *component diagram* dilihat pada Tabel 2.4.



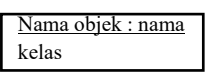

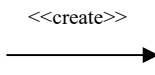
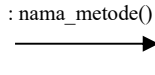
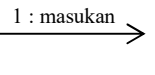
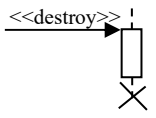
Tabel 2.4 Simbol *Component Diagram*.

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Package</i>	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen
2		<i>Component</i>	Komponen system
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

2.3.6.5 *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (Sukamto dan Shalahuddin, 2016). Simbol *sequence* diagram dapat dilihat pada Tabel 2.5.

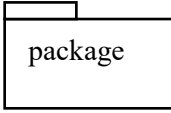
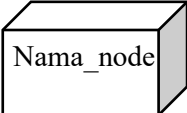
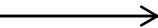
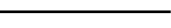
Tabel 2.5. Tabel *Sequence Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat 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		<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 create	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6		Pesan tipe call	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.
7		Pesan tipe send	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8		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> .

2.3.6.6 Deployment Diagram

Deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram deployment juga dapat digunakan untuk memodelkan sistem tambahan yang menggambarkan rancangan *device*, *node*, dan *hardware*. Sistem *client*, sistem terdistribusi murni, rekayasa ulang aplikasi (Sukamto dan Shalahuddin, 2016). Simbol *deployment* diagram ditunjukkan pada Tabel 2.6.

Tabel 2.6. Simbol - Simbol *Deployment*

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> .

2.3.7 Framework

Menurut Raharjo, (2015), *Framework* adalah suatu kumpulan kode berupa pustaka (*library*), dan alat (*tool*) yang dipadukan sedemikian rupa menjadi satu kerangka (*framework*) guna memudahkan dan mempercepat proses pengembangan aplikasi berbasis *website*.

2.3.8 MySQL

Menurut Edy dan Zaki, (2014), MySQL adalah sebuah *software database*. MySQL adalah sebuah database yang dapat berjalan sebagai *server* maupun *client*. Produk *database* MySQL ini memiliki *General Public Licence*, dan bersifat *open source*, sehingga bebas untuk digunakan, diedarkan, maupun dikembangkan kembali tanpa harus khawatir dengan hak cipta. MySQL merupakan tipe data rasional yang artinya MySQL menyimpan datanya dalam bentuk Tabel-Tabel yang saling berhubungan.

2.3.9 Xampp

Menurut Sutanto, (2014), XAMPP merupakan singkatan dari X (empat operasi apapun), Apache, MySQL, PHP, dan Perl. XAMPP merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dalam paketnya sudah terdapat Apache (web server), MySQL (database), PHP (server side scripting), Perl, FTP server, PhpMyAdmin dan berbagai pustaka bantu lainnya.

2.3.10 Bootstrap

Menurut Husein, (2013), Bootstrap merupakan *framework* ataupun *tools* untuk membuat aplikasi *web* ataupun *web responsive* secara tepat, mudah, dan gratis. Bootstrap terdiri dari CSS (*Cascading Style Sheet*) dan HTML (*Hypertext Markup Language*) untuk menghasilkan *Grid*, *Layout*, *Typhography*, *Table*, *Form*, *Navigation*, dan lain-lain. Didalam Bootstrap juga sudah terdapat *JQuery plugins* untuk menghasilkan komponen UI yang menarik seperti *Transitions*, *Modal*, *Dropdown*, *Scrollspy*, *Tooltip*, *Tab*, *Popover*, *Alert*, *Button*, *Carousel*, dan lain-lain.

2.3.11 Black Box

Menurut Sukamto dan Shalahuddin, (2015), *Black Box Testing* (Pengujian Kotak Hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Maka dapat disimpulkan bahwa *Black Box Testing*

(Pengujian Kotak Hitam) merupakan pengujian perangkat lunak untuk menguji *unit* tanpa menguji desain dan pengkodean bahwa suatu program telah sesuai dengan proses yang diinginkan.

2.3.12 CSS

Menurut Hartono, (2013), CSS (*Cascading Style Sheet*) adalah kumpulan aturan-aturan pemformatan yang mengontrol tampilan konten dalam sebuah halaman *web*. Terdapat tiga jenis CSS, yaitu:

- a. *Inline style sheet* : cukup menambahkan atribut *style* di *tag* yang ingin kita berikan pemformatan.
- b. *Internal style sheet* : meletakkan aturan pemformatan dengan CSS dibagi `<head>` dari *html* dengan tambahan *tag* `<style>`.
- c. *External style sheet* : memisahkan antara file CSS dengan file HTML-nya.

2.3.13 JQuery

Menurut Beighly, (2010), JQuery adalah *add-on* library javascript. Keunggulan menggunakan JQuery dibandingkan dengan javascript, yaitu menyederhanakan kode javascript dengan cara memanggil fungsi-fungsi yang disediakan oleh JQuery . Javascript sendiri merupakan bahasa scripting yang bekerja di sisi *client/browser* sehingga *website* menjadi lebih interaktif. JQuery pertama kali dirilis tahun 2006 oleh John Resig. JQuery menjadi sangat populer sehingga telah digunakan pada banyak Website –Website kelas dunia seperti Google, Amazon, Twitter, ESPN, dan lain-lain.