

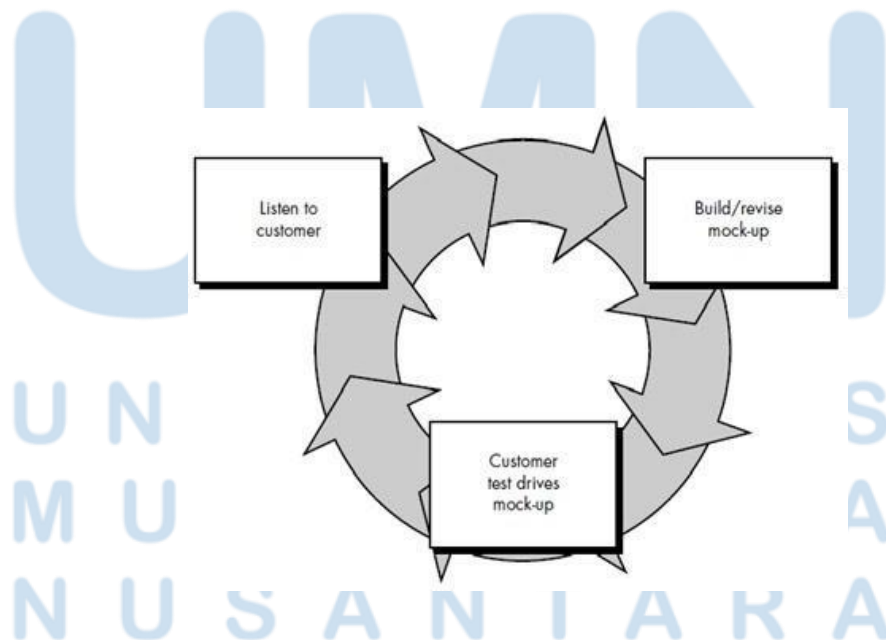
BAB II

LANDASAN TEORI

2.1 Metode Prototype

Pada penelitian kali ini metode yang digunakan dalam pembangun sistem informasi mini ERP ini yaitu dengan menggunakan metode prototype alasan peneliti memilih metode ini karena dianggap sesuai dengan kebutuhan dari pihak perusahaan, dimana metode prototype sendiri menurut Metode Prototype menurut [2], dimulai dengan mengumpulkan kebutuhan. Pengembang dan klien bertemu guna mendefinisikan obyektif keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan dari segi input dan format output serta gambaran interface, kemudian dilakukan perancangan cepat. Dari hasil perancangan cepat tersebut nantinya akan dilakukan pengujian dan evaluasi.

Berikut ini merupakan contoh dari metode prototype itu sendiri



Gambar 2.1 Metode Prototyping

Berdasarkan gambar diatas terdapat tiga siklus yaitu sebagai berikut :

a. *Listen to Customer* (Mendengarkan Pelanggan)

Pada tahap ini merupakan identifikasi kebutuhan user, proses ini dilakukan agar dapat memperoleh informasi mengenai permasalahan yang terjadi oleh klien. Data yang diperoleh dari permasalahan tersebut yang nantinya menjadi acuan untuk dilakukan proses pencarian solusi dan pengembangan pada tahap selanjutnya.

b. *Build and Revise Mock-up* (Membangun dan Memperbaiki *Prototype*)

Setelah kebutuhan sistem terkumpul, maka akan dilakukan proses perancangan *prototype* pada sistem yang diusulkan oleh *user*, yang mana tahap-tahapannya sebagai berikut:

- Perancangan proses-proses yang akan terjadi di dalam sistem, seperti, *input* (masukan), *output* (keluaran) dari sistem yang telah diusulkan.
- Perancangan UML (*Unified Modelling Language*), hal ini dilakukan untuk menspesifikasikan sistem tentang apa yang diperlukan dan bagaimana sistem tersebut direalisasikan. Perancangan UML yang digunakan pada sistem ini meliputi: *Use-Case Diagram* dan *Activity Diagram*.
- Perancangan *Interface* (antarmuka) dan fitur yang dibutuhkan oleh klien (*User*).

c. *Customer Test Drives Mock-up* (Pengujian *Prototype*)

Pada tahap ini akan dilakukan pengujian terhadap *prototype* sistem yang telah dibuat, serta mengevaluasi apakah *prototype* sistem yang sudah dibuat sudah sesuai dengan yang diharapkan. Apabila dari hasil pengujian *prototype* tersebut belum memenuhi kebutuhan klien (*user*), maka pengembang akan melakukan proses perbaikan ulang *prototype* sampai *prototype* tersebut menjadi sistem yang final dan benar-benar diterima atau sesuai dengan keinginan *user*. Proses pengujian *prototype* sistem nantinya menggunakan teknik pengujian *black box*. Hasil dari pengujian *black box* tersebut nantinya akan dibahas pada pembahasan selanjutnya di Bab V mengenai hasil pengujian.

2.2 CodeIgniter

CodeIgniter merupakan aplikasi open source berupa framework PHP dengan model MVC (Model, View, Controller) untuk membangun aplikasi web dinamis dengan cepat dan mudah. CodeIgniter memiliki desain dan struktur file yang sederhana, didukung dengan dokumentasi yang lengkap sehingga framework ini lebih mudah dipelajari. CodeIgniter ini memungkinkan para pengembang untuk menggunakan framework secara parsial atau secara keseluruhan. Artinya bahwa CodeIgniter masih memberi kebebasan kepada para pengembang untuk menulis bagian-bagian kode tertentu di dalam aplikasi menggunakan cara konvensional atau dengan syntax umum didalam PHP, tidak harus menggunakan aturan penulisan kode di CodeIgniter [3].

File `index.php`, berfungsi sebagai front controller, menginisialisasi resource utama yang dibutuhkan untuk menjalankan CodeIgniter.

1. Router, memeriksa HTTP request untuk menentukan apa yang harus dilakukan.
2. Jika file cache ada, dikirim langsung ke browser, melewati eksekusi sistem normal.
3. Keamanan, sebelum controller aplikasi dimuat, HTTP request dan setiap data pengguna yang disubmit disaring terlebih dahulu untuk keamanan
4. Controller, memuat model, library utama, helper, dan setiap resource lainnya yang diperlukan untuk memproses permintaan khusus.
5. View, proses render kemudian dikirim ke web browser agar dapat dilihat. Jika caching diaktifkan, view dicache terlebih dahulu sehingga pada permintaan berikutnya dapat dilayani

2.3 Unified Modelling Language (UML)

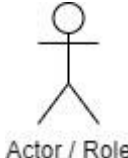

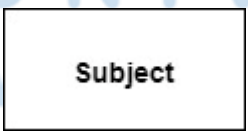
Menurut peneliti [4] UML adalah seperangkat diagram, struktur, dan teknik untuk memodelkan dan merancang program dan aplikasi berorientasi objek. Menurut [5], UML digunakan sebagai notasi untuk berbagai kegiatan, seperti memodelkan kasus bisnis, menganalisis bentuk sistem, serta arsitektur dan desain awal. Peneliti [6] juga menjelaskan bahwa UML menyajikan berbagai aspek sistem perangkat lunak yang sangat beragam dalam satu kerangka kerja menggunakan konsep object oriented.


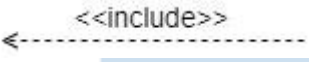
2.3.1 Use case Diagram

Use case adalah sebuah kegiatan yang menggambarkan perilaku suatu sistem dalam berbagai kondisi ketika sistem merespon permintaan dari

pelaku utama. Pelaku utama melakukan permintaan terhadap sistem terkait suatu tujuan dan sistem akan meresponnya [7] Menurut peneliti [6], use case adalah diagram yang memungkinkan kita untuk menggambarkan kemungkinan skenario penggunaan yang dikembangkan sistem. Ini mengungkapkan apa yang harus dilakukan sistem tetapi tidak membahas rincian realisasi. Menurut peneliti [8], use case adalah model persyaratan sistem pada tingkat tinggi. use case diagram terutama digunakan untuk memvisualisasikan use case, sektor terkait, dan interaksinya, berikut ini merupakan symbol dari use case.

Tabel 2.1 Symbol Usecase



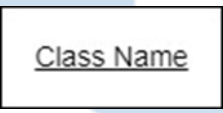
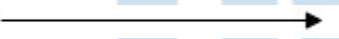
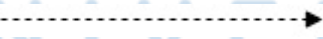

Simbol	Keterangan
 <p>Actor / Role</p>	<p><i>Actor :</i></p> <ul style="list-style-type: none"> - Merupakan orang atau sistem yang mendapatkan manfaat dari dan bersifat eksternal terhadap subjek. - Dapat dikaitkan dengan aktor lain menggunakan asosiasi spesialisasi/superclass, dilambangkan dengan panah dengan panah berongga.
 <p>Use Case</p>	<p><i>Use Case :</i></p> <ul style="list-style-type: none"> - Adalah elemen utama dari fungsionalitas sistem. - Dapat menyertakan dan memperpanjang <i>use case</i> lain. - Ditempatkan dalam batas sistem.
 <p>Subject</p>	<p><i>Subject Boundary :</i></p> <ul style="list-style-type: none"> - Tergolong dari nama subjek di dalam atau di atas. - Mewakili ruang lingkup subjek, misalnya sistem atau individu proses bisnis.




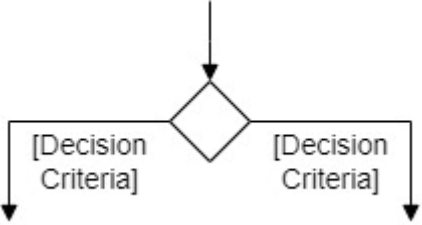
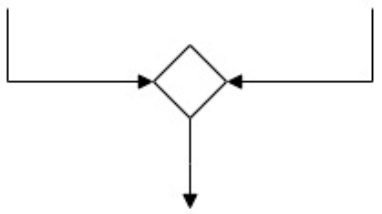
	<p><i>Association Relationship :</i></p> <ul style="list-style-type: none"> - Digunakan untuk menghubungkan <i>actor</i> dengan <i>use case</i> yang digunakan untuk berinteraksi.
	<p><i>Include Relationship :</i></p> <ul style="list-style-type: none"> - Merupakan penyertaan fungsionalitas dari satu <i>use case</i> di dalam <i>use case</i> yang lain. - Memiliki panah yang ditarik dari <i>use case</i> dasar ke <i>use case</i> yang akan digunakan.

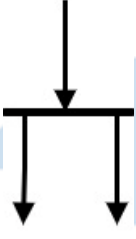
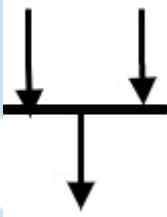
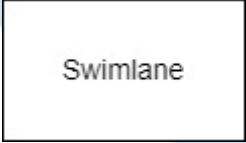
2.3.2 Activity Diagram

Activity diagram menunjukkan logika kondisional untuk urutan aktivitas sistem yang diperlukan untuk menyelesaikan suatu proses bisnis. Aktivitas individu bisa saja manual atau otomatis. Selanjutnya, setiap kegiatan adalah tanggung jawab unit organisasi tertentu [9]. Menurut Peneliti [10], activity diagram adalah diagram yang menyerupai obrolan bagan alur horisontal yang menunjukkan tindakan dan peristiwa saat terjadi. Activity diagram menunjukkan urutan tindakan yang dilakukan dan mengidentifikasi hasilnya. Menurut peneliti [11], activity diagram adalah diagram yang menawarkan mekanisme aliran kontrol serta mekanisme aliran data yang mengkoordinasikan aksi yang membentuk suatu aktivitas [12] mengungkapkan bahwa activity diagram adalah diagram yang menggambarkan aktivitas utama dan hubungan antara aktivitas dalam suatu proses. Berikut ini merupakan symbol activity Diagram.

Tabel 2.2 Activity Diagram

Simbol	Keterangan
	<p><i>Action :</i></p> <ul style="list-style-type: none"> - Merupakan perilaku yang sederhana dan tidak dapat dijabarkan. - Diberi label sesuai dengan namanya.
	<p><i>Activity :</i></p> <ul style="list-style-type: none"> - Digunakan untuk mewakili serangkaian tindakan. - Diberi label sesuai dengan namanya.
	<p><i>Object Node :</i></p> <ul style="list-style-type: none"> - Digunakan untuk merepresentasikan suatu objek yang terhubung ke sekumpulan arah gabungan objek. - Diberi label sesuai dengan namanya.
	<p><i>Control Flow :</i></p> <ul style="list-style-type: none"> - Menunjukkan urutan eksekusi.
	<p><i>Object Flow :</i></p> <ul style="list-style-type: none"> - Menunjukkan aliran suatu objek dari satu aktivitas(tindakan) ke aktivitas(tindakan) lainnya.
	<p><i>Initial Node :</i></p> <ul style="list-style-type: none"> - Menggambarkan awal dari serangkaian tindakan atau kegiatan.

	
	<p>Final – Activity Node :</p> <ul style="list-style-type: none"> - Digunakan untuk menghentikan semua aliran kontrol dan objek mengalir dalam suatu kegiatan atau tindakan.
	<p>Final – Flow Node :</p> <ul style="list-style-type: none"> - Digunakan untuk menghentikan aliran kontrol atau aliran objek tertentu.
	<p>Decision Node :</p> <ul style="list-style-type: none"> - Digunakan untuk mewakili kondisi pengujian untuk memastikan bahwa aliran kontrol atau aliran objek hanya turun satu jalur. - Diberi label dengan kriteria keputusan untuk melanjutkan ke jalur tertentu.
	<p>Merge node :</p> <ul style="list-style-type: none"> - Digunakan untuk menyatukan kembali jalur keputusan yang berbeda yang dibuat menggunakan decision node.

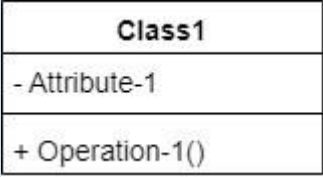
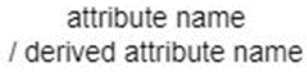
	<p>Fork Node :</p> <ul style="list-style-type: none"> - Digunakan untuk membagi perilaku menjadi sekumpulan aktivitas paralel atau bersamaan.
	<p>Join Node :</p> <ul style="list-style-type: none"> - Digunakan untuk menyatukan kembali serangkaian activity paralel atau bersamaan.
	<p>Swimlane :</p> <ul style="list-style-type: none"> - Digunakan untuk memecah activity diagram menjadi baris dan kolom untuk menetapkan aktivitas individu (atau tindakan) untuk individu atau objek yang bertanggung jawab untuk menjalankan aktivitas (atau tindakan). - Diberi label sesuai dengan nama individu atau objek yang bertanggung jawab.

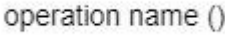
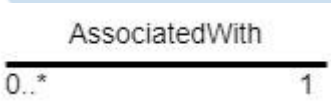
2.3.3 Class Diagram

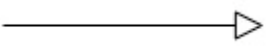
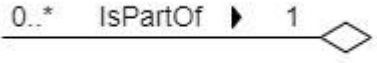
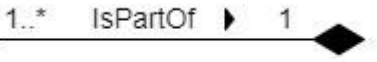
Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

Berikut ini merupakan symbol dari class diagram sebagai berikut:

Tabel 2.3 Symbol Class Diagram

Simbol	Keterangan
	<p><i>A class :</i></p> <ul style="list-style-type: none"> - Mewakili jenis orang, tempat, atau hal yang diperlukan sistem untuk menyimpan informasi. - Memiliki nama yang ditulis dengan menggunakan huruf tebal dan berada di kompartemen atas. - Memiliki daftar atribut yang berada di kompartemen tengah. - Memiliki daftar operasi yang berada di kompartemen bawah.
	<p><i>An attribute :</i></p> <ul style="list-style-type: none"> - Menggambarkan keadaan suatu objek. - Dapat diturunkan dari atribut lain.

	<p><i>An operation :</i></p> <ul style="list-style-type: none"> - Menggambarkan tindakan atau fungsi yang dapat dilakukan oleh <i>class</i>. - Dapat diklasifikasikan sebagai <i>constructor</i>, <i>query</i>, atau <i>update operation</i>.
	<p><i>An association :</i></p> <ul style="list-style-type: none"> - Merupakan hubungan antara beberapa <i>class</i> atau dengan <i>class</i> itu sendiri. - Diberi label dengan menggunakan kata kerja atau peranan yang mewakili hubungan tersebut. - Berisi simbol multiplisitas, yang mewakili waktu minimum dan maksimum suatu <i>instance class</i> yang dapat diasosiasikan dengan <i>instance class</i> terkait.

	<p><i>A generalization :</i> Menggambarkan hubunganantara beberapa <i>class</i>.</p>
	<p><i>An aggregation :</i></p> <ul style="list-style-type: none"> - Mewakili hubungan logis antara beberapa kelas atau dengan kelas itu sendiri. - Merupakan bentuk khusus dari asosiasi.
	<p><i>A composition :</i></p> <ul style="list-style-type: none"> - Mewakili hubungan fisik antara beberapa kelas atau dengan kelas itu sendiri. - Merupakan bentuk khusus dari asosiasi

2.3.4 Hyper Text Markup Language (HTML)

HTML merupakan salahsatu bahasa yang biasa digunakan oleh pengguna dalam membuat tampilan yang digunakan oleh web application [13].

2.3.5 PHP

Hypertext Preprocessor atau dengan nama lain PHP merupakan bahasa pemrograman untuk pembuatan website dinamis, yang mampu berinteraksi dengan pengunjung atau penggunanya [14].

2.3.6 XAMPP

Xampp merupakan salah satu bundle web server yang populer digunakan untuk windows karena kemudahan dalam proses instalasinya Bundel program open source tersebut berisi antara lain server web apache, interpreter PHP, dan basis data MySQL [15].

2.3.7 MYSQL

Mysql adalah basis data yang bersifat open source sehingga banyak digunakan di dunia walaupun gratis, MySQL tetap berkualitas dan sudah cukup memberikan performa yang memadai menurut [16].

2.4 Penelitian Terdahulu

Berikut ini merupakan penelitian terdahulu pada penelitian kali ini :

Tabel 2.4 Penelitian Terdahulu

No	Jurnal	Hasil
1	Judul Jurnal : Analysis and Design of an Web-Based Ticketing Service Helpdesk at Food and Packaging Machinery Company Penulis : Jansen Wiratama, Fenina Adline Twince Tobing Tahun Agustus 2022 Jurnal :	Hasil dari penelitian tersebut merupakan sebuah sistem informasi yang mampu untuk membantu <i>customer service</i> dalam melakukan update data secara real-time.

	Ultima Infosys : Jurnal Ilmu Sistem Informasi, Vol.13, No.1	
2	Judul Jurnal : Perancangan Sistem Informasi Penjualan Berbasis Web pada Toko Campus Mart Unimuda Sorong dengan PHP Dan MySql Penulis : Winarti, Muhammad Ihsan, Novita Wulandari Tahun : Februari 2020 Jurnal : Jurnal PETISI, Vol.1, No.1	Hasil dari penelitian pada UNIMUDA Sorong yang dibangun menggunakan dengan basis data MySQL mampu memberikan informasi yang dibutuhkan toko sehingga pengguna dapat diberikan kemudahan untuk melakukan transaksi pembayaran.
3	Judul Jurnal : Pengembangan Sistem Informasi Untuk Memfasilitasi Proses Adopsi Anjing Berbasis WEB Penulis : Amelinda Chendra, Kristina G. Simanjuntak, Andree E. Widjaja, Suryasari Tahun : July 2019 Jurnal : Jurnal Ilmiah MATRIK, Vol.21 No.1	Hasil dari penelitian ini merupakan sistem adopsi anjing berbasis web. Sistem informasi ini telah dilakukan pengembangan dan penyesuaian dengan kebutuhan objek penelitian. Pengujian sistem telah di uji coba oleh pengguna dan telah berjalan dengan baik.
4	Judul Jurnal : Penerapan Metode Prototype Dalam Perancangan Sistem Informasi Perhitungan Volume dan Cost Penjualan Minuman Berbasis Website Penulis :	Hasil dari penelitian ini merupakan sistem yang dapat menghitung volume penjualan dan cost minuman di hotel mercure Pontianak yang masih

Wahyu Nugraha, Muhamad Syarif Tahun : Desember 2018 Jurnal : JUSIM, Vol.3, No.2	menggunakan catatan pembukuan.
---	--------------------------------

1. Pada penelitian terdahulu yang pertama ini memberikan kontribusi referensi untuk membuat dan merancang modul approval pada data PO.
2. Penelitian kedua memberikan gambaran dan contoh pembuatan pada modul transaksi seperti pemesanan PO.
3. Penelitian ketiga memberikan gambaran dan contoh pada modul konfirmasi pemesanan.
4. Penelitian keempat memberikan gambaran dan contoh pada modul pencatatan laporan transaksi.

