

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Teori Terkait**

Penelitian yang dilakukan oleh Rahmatdhan dkk (2021) dalam jurnal penelitiannya yang berjudul “Pengembangan Sistem Informasi Penjualan Ikan Cupang Berbasis Web di Labetta Solo” berhasil dibuat dengan menggunakan metode *Waterfall* meliputi *analysis, design, coding, testing, implementation* sistem ini nantinya akan dibuat menggunakan *database MySQL* dan PHP Codeigniter sebagai bahasa pemrogramannya. Hasil yang diharapkan nantinya penjual dapat menjual ikan melalui sistem ini dan terdapat *rating* atau *grade* dari kualitas ikan sehingga memudahkan customer dalam melakukan pemilihan ikan yang akan dibeli[8].

Penelitian yang dilakukan oleh Fajar Ramadhan dkk (2020) dalam jurnal penelitiannya yang berjudul “Perancangan Sistem Informasi Penjualan Ikan Air Tawar Berbasis *Web*” Perancangan sistem informasi penjualan ikan air tawar berbasis *web* dibangun dengan metode SDLC (*System Development Life Cycle*) yang memiliki beberapa tahap yaitu, *Planning, Analisis, Desain, Implementasi, dan Pengujian*. Dimana untuk melakukan analisis sistem menggunakan metode PIECES memiliki beberapa tahap yaitu berdasarkan kategori yang disebutkan dalam tiap hurufnya *Performance, Information, Economic, Control, Efficiency, Service*. untuk melakukan pengumpulan data dengan Studi pustaka, Wawancara, dan Observasi. Hasil dari penelitian ini

adalah aplikasi penjualan ikan berbasis *website*[5].

Penelitian yang dilakukan oleh Marianus Gega Kelen dkk (2023) dalam jurnal penelitiannya yang berjudul “Sistem Informasi Penjualan Ikan Berbasis *Web* (Studi Kasus Di Toko Sang Surya Kecamatan Titehena Kabupaten Flores Timur)”, Penelitian ini menggunakan Metode *waterfall* sebagai Metode Pengembangan dan pemodelan sistem menggunakan bahasa pemrograman *Java* pada bagian *web* dan bahasa pemrograman PHP Pada bagian Administrator dengan *MySQL* untuk pengolahan databasenya. Sistem Informasi Penjualan Pada Toko Sang Surya Berbasis *Web* Dapat mengumpulkan, mengidentifikasi, menyimpan, mengolah dan memberikan informasi Penjualan Pada Toko Sang Surya yang menjadi target dalam penelitian ini[10].

Penelitian yang dilakukan oleh Firda Mayang Sari dkk (2021) dalam jurnal penelitiannya yang berjudul “Sistem Informasi Penjualan Ikan Cupang Berbasis *Web* Di Cals.Bettaqueens Jakarta Pusat”, dijelaskan dalam penelitiannya penelitian ini adalah untuk merancang aplikasi penjualan dan desain sistem penjualan berbasis *web* pada Cals.Bettaqueens. Hal ini dibuat untuk meningkatkan keuntungan dan memberikan kemudahan bagi pelanggan untuk memesan dan membeli ikan. Metode ini menggunakan pengumpulan data, analisis (dengan survei, wawancara dan studi pustaka ) dan desain (bentuk masukan, proses, laporan, dan database). Hasil dari penelitian ini adalah sistem penjualan berbasis *web* yang dapat digunakan oleh pelanggan 24 jam. Kesimpulannya adalah sistem penjualan berbasis *web*

memiliki fasilitas seperti halaman informasi, halaman order, dan membatalkan halaman[11].

Penelitian yang dilakukan oleh Athallah Rafi Perdana dkk (2025) dalam jurnal penelitiannya yang berjudul “Perancangan Sistem Informasi Penjualan Makanan Olahan Ikan Beku Berbasis Web Pada CV Rizky Food”, dijelaskan dalam penelitiannya Dengan tujuan membuat data-data untuk pelaporan gampang dicari, dirasa perlu dibuat sistem informasi dengan menggunakan pendekatan metode *Waterfall* dalam perancangan sistem informasi yaitu dari menganalisis *web* yang akan dibuat, desain, pelaksanaan, dan uji coba. Sistem informasi yang dirancang pada CV Rizky Food dan setelah dibuat perancangan dengan diagram *use case* manager atau karyawan bisa menggunakan serta dapat mencapai ketepatan dan kecepatan sehingga pekerjaan tersebut dalam aktivitas setiap harinya menjadi cepat dan akurat.[12].

Penelitian ini akan dilakukan dan sedang berjalan yaitu merancang dan membangun sistem penjualan bandeng presto berbasis *website* yang bertujuan untuk meningkatkan efisiensi operasional bisnis bandeng presto. Dengan diterapkannya fitur pengelolaan stok, pencatatan transaksi, serta perhitungan laba rugi, pembayaran *online*, diharapkan dapat mempermudah proses bisnis bandeng presto serta meningkatkan kepuasan pelanggan dalam berbelanja *online*.

## 2.2 Landasan Teori

Landasan teori merupakan konsep, teori, prinsip dan pendapat yang mendukung proses pengembangan sistem dan sumber daya yang digunakan dalam pengembangan sistem yang dibuat. Berikut dijelaskan teori-teori yang menjadi dasar atau pedoman dalam Perancangan Aplikasi Penjualan Dan Pemesanan Bandeng Presto ZKF 11 Berbasis *Website*, diantaranya adalah pengolah data yaitu *PHPMyAdmin*, *MySQL*, *Visual Studio Code* dan lain-lain.

### 2.2.1 Perancangan

Perancangan merupakan langkah pertama di dalam fase pengembangan rekayasa suatu produk atau sistem yang menyangkut berbagai komponen sehingga akan menghasilkan sistem yang sesuai dengan hasil dari tahap analisa sistem. Jadi dapat disimpulkan bahwa perancangan merupakan tahap awal untuk merencanakan atau menggambarkan suatu sistem yang akan dibuat yang meliputi berbagai komponen atau elemen terpisah yang nantinya akan menjadi satu sistem utuh dan berfungsi seperti yang diinginkan. (Himawan et al., 2014).

### 2.2.2 Website

*Website* adalah sebuah tempat yang memungkinkan seseorang menyatakan dirinya, hobinya, pengetahuannya, produk yang dijualnya dan apapun juga yang dapat di akomodasikan oleh teks, tulisan, gambar, video, animasi dan file multimedia lainnya[13].

Sebuah halaman *web* merupakan berkas yang ditulis sebagai

berkas teks biasa (*plain text*) yang diatur dan dikombinasikan sedemikian rupa dengan instruksi-instruksi berbasis HTML atau XHTML. Berkas tersebut kemudian diterjemahkan oleh peramban *web* dan ditampilkan seperti layaknya sebuah halaman pada monitor komputer. Halaman-halaman *web* tersebut diakses oleh pengguna melalui protokol komunikasi jaringan yang disebut sebagai HTTP, sebagai tambahan untuk meningkatkan aspek keamanan dan aspek privasi yang lebih baik. Situs *web* dapat pula mengimplementasikan mekanisme pengaksesan melalui protokol HTTPS.

### 2.2.3 HTML

HTML adalah Bahasa pemrograman yang fleksibel di mana kita bisa meletakkan *script* dari bahasa pemrograman lain seperti *JAVA*, *Visual Basic*, *C* dan lain-lain. Jika HTML tersebut tidak dapat mendukung suatu perintah pemrograman tertentu. *Browser* tidak akan menampilkan kotak dialog "*Syntax Error*" jika terdapat penulisan kode yang keliru pada skrip HTML sepanjang kode-kode yang kita tuliskan merupakan kode-kode HTML tanpa penambahan kode-kode dari luar seperti *Java*. Oleh karena itu, jika terjadi *syntax error* pada skrip HTML, efek yang paling jelas adalah HTML tersebut tak akan ditampilkan pada halaman jendela[14].



Gambar 2.1. Logo HTML

#### 2.2.4 Visual Studio Code

*Visual studio code* adalah perangkat lunak penyunting kode sumber buatan microsoft untuk *linux*, *macOS*, dan *windows*. *Visual studio code* menyediakan fitur seperti penyusunan sintaksis, penyelesaian kode, kutipan kode, merefaktor kode, pengawakutuan, dan Git. *Microsoft* merilis sumber kode visual studio code di repositori *Github* dengan *Visual studio code* merupakan penyunting kode-sumber yang dapat digunakan untuk beragam bahasa pemrograman, termasuk java, javascript, Go, Node.js, Python dan C++[6]. *Visual studio code* memperkerjakan komponen penyunting yang sama dengan nama *code name* “monaco” yang digunakan di Azure devops (awalnya dipanggil *Visual Studio Online* dan *Visual Studio Team Services*).

#### 2.2.5 Framework CodeIgniter

*Codeigniter* merupakan aplikasi *opensource* yang berupa kerangka kerja PHP dengan model MVC (*Model, View, Controller*) untuk membangun situs *web* dinamis dengan menggunakan PHP. *Codeigniter* memudahkan pengembang *web* dengan cepat dan mudah dibandingkan dengan membuatnya dari awal. *Codeigniter* rilis

pertama kali pada 28 Februari 2006[15].

Kerangka kerja secara sederhana dapat diartikan kumpulan dari fungsi-fungsi/prosedur-prosedur dan kelas-kelas untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang pemrogram, tanpa harus membuat fungsi atau kelas dari awal. Ada beberapa alasan mengapa menggunakan kerangka kerja:

1. Mempercepat dan mempermudah pembangunan sebuah aplikasi *web*.
2. Relatif memudahkan dalam proses pemeliharaan karena sudah ada pola tertentu dalam sebuah kerangka kerja (dengan syarat pemrogram mengikuti pola standar yang ada).
3. Umumnya kerangka kerja menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya validasi, ORM, paginasi pangkalan data ganda, scaffolding, pengaturan sesi, penanganan pengecualian, dan lain-lain).
4. Lebih bebas dalam pengembangan jika dibandingkan CMS.
5. Mendukung pembuatan CRUD otomatis (*Create, Read, Update, Delete*) dari pihak ketiga.



Gambar 2.2. Logo CodeIgniter

### 2.2.6 Database

Basis data dapat didefinisikan atau diartikan sebagai kumpulan data yang disimpan secara sistematis didalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (*software*) program atau aplikasi untuk menghasilkan informasi. Pendefinisian basis data meliputi spesifikasi berupa tipe data, struktur data dan juga batasan-batasan pada data yang kemudian disimpan.

Basis data merupakan aspek yang sangat penting dalam sistem informasi karena berfungsi sebagai gudang penyimpanan data untuk diolah lebih lanjut. Basis data menjadi penting karena dapat mengorganisasikan data, menghindari duplikasi data, menghindari hubungan antar data yang tidak jelas dan juga update yang rumit.

Proses memasukan dan mengambil data ke dan dari media penyimpanan data merupakan perangkat lunak yang disebut dengan sistem manajemen basis data (*database management system*). DBMS merupakan sistem perangkat lunak yang memungkinkan pengguna basis data (*database user*) untuk memelihara, mengontrol dan juga mengakses data secara praktis dan efisien[16].

### 2.2.7 MySQL

*MySQL* adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. *MySQL* menggunakan bahasa SQL untuk mengakses *database* nya. *MySQL* tersedia untuk beberapa *platform*, di antara nya adalah untuk versi *windows* dan versi *linux*. Untuk melakukan administrasi secara lebih mudah terhadap *MySQL*, dapat menggunakan *software* tertentu, di antara nya adalah *phpmyadmin* dan *mysql yog*. Pada kesempatan kali ini, kita akan menggunakan *phpmyadmin*, yang terdapat dalam *bundle* XAMPP, yang dapat di peroleh di [www.apachefriends.org](http://www.apachefriends.org) [17].



Gambar 2.3. Logo MySQL

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

Menurut (Pressman, 2010:841) *Unified modeling language* (UML) adalah bahasa standar untuk menulis denah perangkat lunak. UML dapat digunakan untuk memvisualisasikan, menentukan,

membangun, dan mendokumentasikan artefak dari sistem perangkat lunak. Dengan kata lain, seperti arsitek bangunan membuat denah yang akan digunakan oleh sebuah perusahaan konstruksi, arsitek *software* membuat diagram UML untuk membantu pengembang perangkat lunak membangun perangkat lunak. Jika memahami kosakata UML, dapat lebih mudah memahami dan menentukan sistem dan menjelaskan desain sistem kepada orang lain.

*Unified Modeling Language* merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasikan pada objek. UML merupakan sebuah standar penulisan atau semacam *blue print* dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik[18]. Terdapat beberapa diagram UML yang sering digunakan dalam pengembangan sebuah sistem yaitu:

1. *Use case*

Merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem dan mempresentasikan sebuah interaksi antara aktor dan sistem. Didalam *use case* terdapat aktor 6 yang merupakan sebuah gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem.

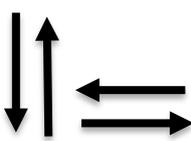
Tabel 0.1. *Use Case Diagram*

Gambar	Nama	Keterangan
	<i>Actor</i>	Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancertor</i> ).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Yang menghubungkan antara objek satu dengan objek lainnya.
	<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.

## 2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram: Merupakan gambaran alur dari aktivitas-aktivitas didalam sistem yang berjalan.

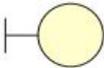
Tabel 0.2. *Activity Diagram*

Gambar	Nama	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi suatu aksi.
	<i>Initial Node</i>	Sebuah diagram aktivitas memiliki status awal.
	<i>Final Node</i>	Status akhir yang dilakukan sistem.
	<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.
	<i>Decision</i>	Pilihan untuk mengambil keputusan.
	<i>Fork/Join</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

### 3. Diagram Urutan (*Sequence Diagram*)

*Sequence Diagram*: Menggambarkan interaksi antar objek didalam dan disekitar yang berupa *message* yang digambarkan terhadap waktu.

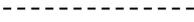
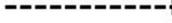
Tabel 0.3. *Sequence Diagram*

Gambar	Nama	Keterangan
	<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
	<i>Boundary Class</i>	Menggambarkan penggambaran dari form.
	<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.
	<i>Control Class</i>	Menggambarkan penghubung antara Boundary dengan tabel.
	<i>Activation</i>	Sebagai sebuah objek yang akan melakukan sebuah aksi.
	<i>Message</i>	Mengindikasikan komunikasi antara objek dengan objek.
	<i>Self Message</i>	Mengindikasikan komunikasi kembali ke dalam sebuah objek itu sendiri.

#### 4. *Class Diagram*

Merupakan gambaran struktur dan deskripsi dari *class*, *package* dan objek yang saling berhubungan seperti diantaranya pewarisan, asosiasi dan lainnya.

Tabel 0.4. Class Diagram

Gambar	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Dependency</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.