

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Teori Terkait**

Pada penelitian yang dilakukan oleh Febriyanti Darnis, Et al, (2019) pesatnya persaingan pasar menjadikan masyarakat Desa Pedado memutar otak dalam mempromosikan Usaha Kecil dan Menengah (UKM) dengan memanfaatkan *website* yang dirintis oleh Darnis dan Azdy. Selain itu kegiatan pemasaran dengan pengaplikasian *website* mampu meningkatkan ilmu pengetahuan dan teknologi (IPTEK) bagi masyarakat, khususnya bagi mitra UKM yang bersangkutan[4].

Pada penelitian yang dilakukan oleh Ni Komang Sri Julyantari, Et al, (2021) dalam zaman teknologi informasi seperti sekarang sangat diperlukan sarana pemasaran produk secara *online* dengan tujuan untuk mempromosikan barang ke jangkauan yang lebih luas, seperti UD Yadnya Wijaya Asri nantinya dapat memasarkan produknya ke seluruh daerah, tidak hanya daerah Denpasar dan sekitarnya saja. Dengan pemanfaatan *website* dalam pemasaran produknya, cara ini bertujuan untuk meningkatkan produksi serta pendapatan bagi perusahaan[5].

## 2.2 Landasan Teori

### 2.2.1 *Website*

*Website* adalah kumpulan halaman dan dokumen pendukungnya, seperti rekaman gambar, rekaman, dan rekaman terkomputerisasi lainnya yang disimpan di server *web* yang umumnya dapat diakses melalui *web*. Dengan kata lain, situs *web* adalah 10 kumpulan folder dan file yang berisi banyak perintah dan fungsi berbeda, seperti fungsi tampilan dan fungsi penyimpanan data, antara lain. Halaman situs web biasanya berupa dokumen yang ditulis dalam *Hyper Text Markup Language* (HTML), yang dapat diakses melalui HTTP. HTTPS adalah protokol yang mengirimkan berbagai informasi dari server situs *web* ke *browser web* untuk ditampilkan kepada pengguna. *Codeigniter Framework* kerja PHP yang disebut *Framework Codeigniter* dapat membantu pengembang mengembangkan aplikasi *web* berbasis PHP dengan lebih cepat[6].

### 2.2.2 *Bootstrap*

Sebuah paket aplikasi yang disebut *bootstrap* digunakan untuk membuat situs *web front-end*. Secara sederhana, *bootstrap* adalah templat desain *web* dengan banyak fitur. *Bootstrap* dibuat untuk mempermudah proses desain *online* bagi pengguna dengan berbagai Tingkat keterampilan, dari pemula hingga ahli. Secara alami, *bootstrap* bersifat sumber terbuka dan dapat ditemukan di

<http://getbootstrap.com>. Logo *bootstrap* dapat dilihat pada Gambar 2.1.



Gambar 2.1 Logo *Bootstrap*

### 2.2.3 *JavaScript*

*Javascript* adalah bahasa yang terdiri dari kumpulan skrip yang bekerja pada dokumen *HTML*. Dalam konteks internet, bahasa ini adalah bahasa skrip utama untuk *web*. Bahasa ini adalah bahasa pemrograman yang dirancang untuk memberikan lebih banyak fungsionalitas kepada *HTML* dengan memungkinkan eksekusi perintah sisi pengguna, yang terlihat di *browser* tetapi tidak di *server web*. Logo *javascript* dapat dilihat pada Gambar 2.2 [7].



Gambar 2.2 Logo *JavaScript*

#### 2.2.4 XAMPP

*XAMPP* adalah perangkat lunak server web apache yang mendukung pemrograman php dan sebelumnya telah tersedia sebagai server database *MySQL*. *XAMPP* adalah program gratis yang mudah digunakan. dan mendukung instalasi *Linux* dan *Windows*, manfaat lainnya termasuk kemampuan untuk menggunakan *Apache Web Server*, *MySQL Database Server*, *PHP Support* (PHP 4 dan PHP 5) dan beberapa modul lainnya (Dinata et al., 2015). Logo *xampp* dapat dilihat pada Gambar 2.3 [8].



Gambar 2.3 Logo *XAMPP*

#### 2.2.5 UML (*Unified Modelling Language*)

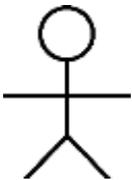
*UML* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek[9].

#### 4.1 Use Case Diagram

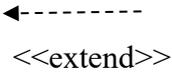
*Use Case* diagram adalah jenis interaksi yang saling berkaitan dengan sistem dan aktor. *Use case* dilakukan dengan menggambarkan jenis interaksi antarpengguna dan program (sistem) serta sistem itu sendiri, *Use case* melalui sebuah cerita yang menggambarkan bagaimana sistem digunakan. *Use case* juga digunakan untuk membentuk perilaku (*behaviour*) sistem yang akan dibuat. Sebuah *Use Case* menggambarkan interaksi antara pengguna (aktor) dan sistem yang ada. Simbol-simbol yang digunakan dalam *Use Case* Diagram yaitu:

Tabel 2.1 *Usecase Diagram*

No	Simbol	Pengertian	Keterangan
1.		<i>Use Case</i>	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang berkerja sama antar unit dan aktor, yang dinyatakan menggunakan kata kerja.

2.		<i>Actor/Aktor</i>	<p>Abstraction dari orang atau sistem lain yang mengaktifkan fungsi sistem target. Untuk mengidentifikasi aktor, pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran dalam konteks sistem target harus dipertimbangkan untuk mengidentifikasi aktivitas. Seseorang atau sistem muncul dalam beberapa cara yang berbeda aktor berintraksi.</p>
----	---	--------------------	--

3.		<i>Association/ Assosiasi</i>	Asosiasi antara aktor dan use case, digambarkan dengan garis sederhana yang menunjukkan siapa atau apa yang meminta intraksi, menunjukkan data.
4.		<i>Generalization/ Generalisasi</i>	Asosiasi antara aktor dan use case yang menggunakan panah terbuka menunjukkan apakah aktor berinteraksi secara pasif dengan sistem.
5.	 <<include>>	<i>Include/Uses</i>	Merupakan di dalam use case lain (required) atau pemanggilan <i>use case</i> oleh use case lain, contohnya penugasan fungsi perangkat lunak..

6.		<i>Extend/Ekstensi</i>	Merupakan perluasan dari use case lain jika kondisi atau syarat terpenuhi.
----	---	------------------------	--

#### 4.2 Diagram Aktivitas (*Activity Diagram*)

*Activity* diagram adalah menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem terlepas dari apa yang dilakukan aktor. Simbol yang digunakan activity diagram yaitu :

Tabel 2.2 Diagram Activity

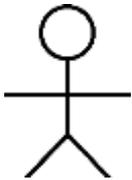
No	Simbol	Pengertian	Keterangan
1.		<i>Activity</i>	Memperlihatkan bagaimana setiap antarmuka kelas saling berinteraksi secara konsisten.
2.		<i>Activity</i> <i>Final Mode</i>	bagaimana objek dibentuk dan dihancurkan.

3.		<i>Initial Node</i>	bagaimana objek dibentuk atau diawali.
4.		<i>Fork Node</i>	Satu aliran berubah menjadi beberapa aliran.

### 4.3 Diagram Urutan (*Sequence Diagram*)

*Sequence Diagram* adalah jenis diagram dalam *Unified Modeling Language (UML)* yang menggambarkan interaksi antara objek dan sistem secara kronologis. Diagram ini menggambarkan bagaimana objek saling berkomunikasi dan bekerjasama untuk menjalankan skenario atau proses. Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

Tabel 2.3 *Sequence Diagram*

No	Simbol	Pengertian	Keterangan
1.		<i>Aktor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2.		<i>Entity Class</i>	Menggambarkan penghubung antara

			<i>boundary</i> dengan tabel.
3.		<i>Asynchronous Message</i>	Menggambarkan proses pengiriman pesan.
4.		<i>Control Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.
5.		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari <i>foem</i> .
6.		<i>A Focus Of Control &amp; A Life Line</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.

#### 4.4 Diagram kelas (*Class Diagram*)

Berfungsi sebagai penghubung antar kelas dan perjas setiap detail dalam desain sebuah sistem. Memerlihatkan aturan-

aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

*Class Diagram* secara khas meliputi : Kelas (*Class*), *Relasi Assosiations*, *Generalitaton* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*.

Tabel 2.4 *Class Diagram*

No	Simbol	Pengertian	Keterangan
1.	1	Hanya Satu	Satu dan hanya satu.
2.	0..*	Nol Atau Lebih	Boleh tidak ada satu atau satu atau lebih
3.	1..*	Satu Atau Lebih	satu atau lebih
4.	0...1	Nola Atau Satu	Boleh tidak ada, maksimal satu.
5.	n...n	n Sampai n	Batasan antara contoh 2..4 mempunyai arti minimal dua maksimal empat.

### 2.2.6 *Visual Studio Code*

*Visual Studio Code* adalah text editor yang ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi multiplatform, termasuk versi *Linux*, *Mac*, dan *Windows*. Teks editor ini dengan mudah mendukung bahasa pemrograman *Javascript*, *Typescript*, dan *Node. Js*, serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang via *marketplace Visual Studio Code* seperti : *C++*, *C#*, *Python*, *Go*, *Java*, *PHP*, dst. (Ummy Gusti Salamah, 2021). Logo visual studio code dapat dilihat pada Gambar 2.4[10].



Gambar 2.4 Logo Visual Studio Code