BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Penelitian ini terdiri atas 2 tahap, yaitu tahap pertama, pembuatan tempe sedangkan tahap kedua uji organoleptik dan uji daya terima produk tempe dari hasil penelitian. Penelitian pembuatan tempe ini menggunakan rancangan eksperimental yang terdiri dari 3 perlakuan. Data yang digunakan pada penelitian ini dianalisis secara kuantitatif. Hasil penelitian didapatkan kesimpulan bahwa ditinjau dari hasil uji organoleptik dan uji daya terima produk tempe[6].

Penelitian ini bertujuan merancang prototipe pengendali suhu dan kelembaban berbasis *IoT* untuk meningkatkan produktivitas fermentasi tempe dan juga bisa memonitoring nya melalui *website* yang akan di buat. Prototipe ini menggunakan mikrokontroler yang akan memproses data dari sensor suhu (DHT22)[7].

Sistem informasi sangat dibutuhkan dalam menghitung jumlah barang dagangan dan persediaan yang dimiliki oleh sebuah perusahaan atau yang biasa di sebut dengan stock opname, tujuan dilakukan stock opname ini adalah untuk mengetahui keakuratan catatan pembukuan yang merupakan salah satu fungsi sistem pengendalian internal kemudian hasilnya dibandingkan dengan jumlah menurut catatan persediaan produk yang tersedia, Anjar Prayogi (2018). Sistem informasi tersebut bisa langsung terhubung ke jaringan seperti *website [8]*. Di zaman yang

perkembangan teknologi dan internet yang berkembang pesat pemilihan menggunangan aplikasi *website* merupakan salah satu pilihan terbaik, dikarenakan aplikasi berbasis *website* dapat digunakan menggunakan *platform* dengan sistem operasi manapun tanpa perlu melakukan instalasi lagi serta spesifikasi yang diperlukan tidak terlalu tinggi, cukup dengan ketersediaan *browser* dan akses internet.

Berdasarkan permasalahan tersebut, maka dirancanglah sebuah Sistem Penjualan Online Berbasis *Website* Pada UMKM Keripik Tempe Idola Pati agar memudahkan UMKM tersebut dalam melakukan pemasaran dan proses transaksi yang lebih efektif. Dengan adanya sistem tersebut maka pelayanan penjual kepada konsumen dapat meningkat, terutama dalam hal penjualan online. Dengan adanya *website* online pribadi maka penjual dapat mengelola produk secara lebih efektif dan efisien tanpa perantara [9].

2.2 Landasan Teori

2.2.1 Sistem Monitoring

Sistem monitoring adalah sebuah mekanisme atau proses yang digunakan untuk mengamati, mengawasi, dan menganalisis kondisi atau performa suatu sistem, objek, atau proses secara *real-time* maupun periodik. Sistem ini bertujuan untuk memastikan segala sesuatunya berjalan sesuai dengan harapan, mendeteksi masalah sejak dini, dan memberikan informasi yang relevan untuk pengambilan keputusan atau tindakan perbaikan.

2.2.2 PHP

Salah satu bahasa pemrograman yang memungkinkan untuk dapat mendukung melihat jadwal secara online adalah PHP (PHP *Hypertext Preprocessor),* dimana PHP merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses data dinamis[10].

2.2.3 *MySQL*

MySQL merupakan suatu jenis database server yang sangat terkenal. *MySQL* termasuk jenis RDBMS (*Relational Database Manajement System*). *MySQL* mendukung bahasa pemrograman PH, bahasa permintaan yang terstruktur, karena pada penggunaannya SQL memiliki berberapa aturan yang telah distandarkan oleh asosiasi yang bernama ANSI [11].

2.2.4 *Visual Studio Code*

Visual Studio Code adalah kode *editor* sumber yang dikembangkan oleh *Microsoft* untuk *Windows, Linux* dan *mac OS. Visual Code* memudahkan dalam penulisan *code* yang mendukung beberapa jenis bahasa pemrograman yang digunakan dan memberi variasi warna sesuai dengan fungsi dalam rangkaian *code* tersebut. Selain itu, fitur lainnya adalah kemampuan untuk menambah *ekstensi* dimana para pengembang dapat menambah *ekstensi* untuk menambah fitur yang tidak ada di *Visual Studio Code* [12].

2.2.5 *CodeIgniter*

codeIgniter adalah sebuah *framework* yang dibuat menggunakan bahasa pemrograman PHP yang bertujuan untuk memudahkan para programmer web untuk membuat atau mengembangkan aplikasi berbasis web. *CodeIgniter* memiliki eksekusi tercepat dibandingkan dengan *framework* lainnya [13].

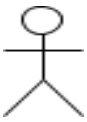
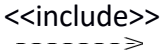
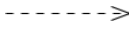
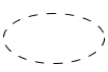
2.2.6 *UML (Unified Modelling Language)*

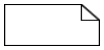
Unified Modeling Language merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasikan pada objek. UML adalah sebuah bahasa pemodelan perangkat lunak yang telah distandarisasi sebagai media penulisan cetak biru *(blueprints)* perangkat lunak. UML merupakan sebuah standar penulisan atau semacam *blue print* dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas - kelas dalam sebuah bahasa yang spesifik [14].

Dalam perancangan sistem terdapat UML yang sering digunakan sebagai berikut:

1) Dalam pembuatan sistem informasi, usecase diagram merupakan pemodelan untuk behavior atau kelakuan sistem informasi. Diagram ini juga bersifat statis. Untuk simbol UseCase Diagram disajikan pada Tabel 1

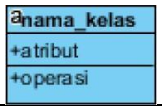
Tabel 2. 1 *Use Case Diagram*

No	Gambar	Nama	Keterangan
1.		*Actor*	Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan *usecase.*
2.		*Dependency*	Menunjukan spesialisasi aktor untuk dapat berpatisipasi dengan *usecase.*
3.		*Generalization*	Abstraksi dari penghubung antara aktor dengan *usecase.*
4.		*Include*	Menunjukan bahwa suatu *use case* seluruhnya merupakan *fungsionalitas* dari *use case* lainnya.
5.		*Extend*	Menunjukan bahwa suatu *use case* merupakan tambahan fungsional dari *usecase* lainnya, jika suatu kondisi terpenuhi.
6.		*Association*	Komunikasi antar aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.
7.		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem
9.		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari

			jumlah dan elemen-elemennya (sinergi).
10.		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

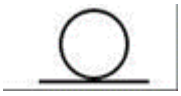
2) *Activity Diagram* atau Diagram Aktivitas merupakan diagram yang bersifat statis, yang menggambarkan aktivitas dari suatu sistem bisnis. Untuk simbol dari diagram aktivitas disajikan pada Tabel 2.

Tabel 2. 2 *Activity Diagram*

No.	Gambar	Nama	Keterangan
1		Kelas	Kelas pada struktur sistem
2	 Nama_interface	Antarmuka/ *Interface*	Sama dengan konsepn interface dalam pemrograman berorientasi objek
3		Asosiasi/ Association	Relasi antar kelas dengan makna umum
4		Asosiasi berarah/ Directed Association	Relasi antar kelas dengan makna kelas
5		Generalisasi	Relasi antar kelas dengan makna generalisasi
6		Kebergantungan / Dependency	Relasi antar kelas dengan maksa kebergantungan antar kelas
7		Agregasi / Aggregation	Relasi antar kelas dengan makna whole- part

3) *Sequence Diagram* atau Diagram Urutan mendeskripsikan diagram interaksi yang mengirimkan pesan dan diterima antar objek. Untuk simbol diagram urutan disajikan pada Tabel 3

Tabel 2. 3 *Sequence Diagram*

Simbol	Nama	Keterangan
	*Actor*	Menggambarkan orang yang sedang berinteraksi dengan sistem.
	*Entity Class*	Menggambarkan hubungan yang akan dilakukan.
	*Boundary Class*	Menggambarkan sebuah gambaran dari *foem*.
	*Control Class*	Menggambarkan penghubung antara *boundary* dengan tabel.
	*A focus of Control & A Life Line*	Menggambarkan tempat mulai dan berakhirnya *massage*.
	A massage	Menggambarkan pengiriman pesan.

4) *Class Diagram* merupakan diagram yang bersifat statis, dalam diagram ini memperlihatkan himpunan kelas, antarmuka, serta relasi. Simbol dari *Class Diagram* disajikan pada Tabel 4.

Tabel 2. 4 *Class diagram*

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