

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terkait

Penelitian yang telah dilakukan oleh Indri Handayani, Erick Febryanto, Egi Wijatriana Bachri dari Perguruan Tinggi Raharja pada tahun 2018 yang berjudul “*Statcounter Sebagai Monitoring Aktivitas Website PESSTA+ Pada Perguruan Tinggi*”. Dalam penelitian yang dilakukan ini membahas tentang penerapan *statcounter* sebagai alat *monitoring* pada *website PESSTA+ Perguruan Tinggi Raharja*. Penelitian ini menggunakan metode perancangan, metode pengumpulan data, dan metode studi pustaka. Metode perancangan adalah penelitian menggunakan *Unified Modeling Language (UML)* untuk merancang secara visual perangkat lunak yang berorientasi objek. UML menyediakan elemen dan diagram yang berbasis paradigma *object-oriented* untuk merancang perangkat lunak secara efektif. Metode Pengumpulan data dapat dilakukan melalui teknik observasi, wawancara, dan studi pustaka. Observasi dilakukan dengan monitoring langsung pada website PESSTA+. Wawancara melibatkan pengelola website PESSTA+ untuk mendapatkan informasi lebih dalam tentang pengoperasian dan penggunaan situs. Metode studi pustaka digunakan untuk mengumpulkan informasi yang relevan dengan topik penelitian dan untuk mengidentifikasi kesenjangan, menghindari pengulangan, serta meneruskan penelitian sebelumnya. *Literatur review* ini membantu para peneliti dalam meningkatkan *visitor*

count sebagai alat monitoring aktivitas *website* PESSTA+. Penelitian tentang penggunaan Statcounter di *website* PESSTA+ Perguruan Tinggi Raharja menghasilkan pemahaman tentang bagaimana aktivitas pengunjung di situs *web* dapat dipantau dan diukur. Hasil penelitian mencakup data pengunjung, statcounter memberikan informasi mengenai *unique visits*, *returning visits*, *first time visits*, dan *page views*; Laporan Berkala, statcounter dapat mengirimkan laporan email secara berkala (mingguan, bulanan, atau tahunan) untuk memberikan gambaran aktivitas situs *web*; Peningkatan *Website*, informasi yang didapat dari *Statcounter* membantu pengelola *website* dalam mengevaluasi dan meningkatkan kinerja situs *web*, seperti mengidentifikasi pola penggunaan dan menentukan perbaikan yang diperlukan[3].

Penelitian yang dilakukan oleh Ayu Atika Sari yang berjudul “Perancangan dan Implementasi *System* Pendeteksi Pengunjung Pada Toko Berbasis Arduino” mengatakan pada teknologi saat ini yang mengalami kemajuan tinggi dimasa sekarang ini dan dalam dunia pertokoan. Berkembangnya teknologi didunia pertokoan, semakin berkembang pula otomatisasi dan efisiensi untuk memudahkan penjaga toko dalam memperoleh informasi. Dalam mewujudkan itu, diperlukan adanya perangkat elektronika untuk memenuhi kebutuhan dan memberi kenyamanan pada pengguna. Alat pendeteksi pengunjung dan penghitung pengunjung yang dibuat ini merupakan alat yang mempermudah ataupun meringankan penjaga toko secara langsung. Pada alat pendeteksi pengunjung dan penghitung pengunjung memiliki dua sensor yaitu sensor pir dan sensor photodiode

sebagai pendeteksi pengunjung dan penghitung pengunjung yang ingin masuk. Alat pendeteksi dan penghitung pengunjung ini berbasis Arduino dengan bahasa pemrograman C. Melalui metode *waterfall*, penelitian ini melalui tahapan analisis, desain, pengkodean, dan pengujian sistem, membantu mempermudah proses penjualan dan pengelolaan toko secara efisien. Hasil dari penelitian yang dilakukan oleh Ayu Atika Sari adalah berhasil merancang dan mengimplementasikan sistem pendeteksi dan penghitung pengunjung berbasis Arduino yang efektif dan akurat. Sistem ini membantu penjaga toko dalam memantau jumlah pengunjung dengan lebih mudah dan efisien, berkat penggunaan sensor PIR dan sensor photodiode untuk mendeteksi dan menghitung pengunjung yang masuk ke toko[4].

Penelitian yang dilakukan oleh Padma Nyoman Crisnapati dalam jurnal penelitiannya yang berjudul “*Visitor Counter System* berbasis Nodemcu dan IoT Sebagai alat penghitung jumlah pengunjung pada CV. Harmoni Permata Berbasis Mikrokontroler” mengatakan kurangnya SDM (Sumber Daya Manusia) ahli yang menguasai bagaimana merancang dan membuat mikrokontroler dan IoT dengan skala industri. Sementara permintaan pasar terhadap kombinasi teknologi ini cukup banyak. Salah satunya adalah pesanan klien mengenai alat penghitung pengunjung yang dapat dimonitor secara *realtime*. Oleh karena itu, perlu diadakan kerjasama antar perguruan tinggi dalam merancang dan membuat alat tersebut. Sebagai tahap awal dari perancangan pembuatan alat ini adalah pembuatan *flowchart* sistem dan juga skematik rangkaian elektronika. Alat ini terdiri dari dua alat yaitu pengirim

sinyal dan penerima sinyal, cara kerja daripada alat ini yaitu pengirim sinyal mengirimkan data kepada penerima sinyal lalu apabila sinyal tidak diterima itu menandakan terdapatnya halangan (manusia) antara alat penerima dan pengirim. Metode yang diterapkan pada pengembangan sistem informasi ini adalah dengan metode pengembangan perangkat lunak *waterfall*. Hasil data tersebut dikalkulasikan dan dikirim ke *internet (server)* melalui koneksi WiFi sehingga dapat ditampilkan dalam antarmuka web. Sistem ini menampilkan data jumlah pengunjung yang masuk dan keluar[5].

Penelitian yang telah dilakukan oleh Naufal Fikri, Sri Dianing Asri, S.T, M.Kom, dari Universitas Mercu Buana pada tahun 2019 yang berjudul “Aplikasi Berbasis *Web* Untuk Pendataan Masuk dan Keluar Barang Dagangan Pada UMKM Studi Kasus: CV. Tiara Cell (*Counter Gadget* dan Aksesoris)”. Dalam penelitian yang dilakukan ini membahas tentang penggunaan layanan aplikasi berbasis *web* untuk pendataan barang, manajemen stok barang serta laporan toko sebagai penunjang kinerja pada Tiara Cell agar segala sesuatunya bisa dilakukan menggunakan akses digital. Penelitian ini menggunakan metode SDLC atau *System Development Life Cycle*. SDLC merupakan suatu urutan dari beberapa proses secara bertahap didalam merancang dan mengembangkan sistem yang dikenal juga dengan nama *Information System Development* atau juga *Application Development*. Peneliti membuat *flow* penelitian agar kegiatan yang dilakukan terstruktur dengan rapih dan efektif. Berdasarkan hasil pengujian dan pembahasan, aplikasi berhasil 100% sesuai dengan apa yang diinginkan pemilik toko

maupun penulis yaitu menjadikan pendataan barang tidak lagi manual, manajemen stok barang otomatis, pendataan karyawan menjadi mudah dilihat dan laporan toko tidak lagi konvensional melihat banyak kertas tetapi hanya tabel. Aplikasi berhasil menyajikan informasi yang baik kepada pemilik toko[6].

Penelitian yang dilakukan Intan Surya Saputra Dhanar pada tahun 2015 yang berjudul “Rancang Bangun Alat Penghitung Jumlah Pengunjung di Toko Adhelina Berbasis Mikrokontroler Atmega 16” pembuatan simulator ini adalah memudahkan penghitungan orang dalam ruangan. Simulator ini dapat digunakan dalam pabrik ataupun dalam tempat hiburan. Tujuan penelitian ini adalah untuk mempermudah *user* atau pengguna penghitung jumlah pengunjung di ruangan atau gedung. Penelitian ini menggunakan metode prototype mikrokontroler. Hasil dari penelitian ini menghasilkan alat jumlah penghitung orang secara otomatis sehingga mempermudah pengguna melakukan pekerjaan dan membantu dalam melakukan rekapan jumlah pengunjung yang datang. Dari hasil pengujian mesin simulator ini disimpulkan bahwa simulator sangat akurat mendeteksi dan menghitung setiap pelanggan yang masuk dan keluar toko[7].

## **2.2 Landasan Teori**

### **2.2.1 Website**

*Website* adalah fasilitas *internet* yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website*

disebut dengan *web page* dan alamat dalam *website* memungkinkan pengguna bisa berpindah dari satu halaman ke halaman lain (*hypertext*), baik diantara halaman yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. Halaman tersebut diakses dan dibaca melalui *browser* seperti Netscape Navigator, Internet Explorer, Mozilla Firefox, Google Chrome, dan aplikasi browser lainnya[8].



Gambar 2. 1 Logo Website

### 2.2.2 PHP (*Hypertext Preprocessor*)

Menurut Tim EMS, PHP adalah singkatan dari *Hypertext Preprocessor*. Saat pertama kali dikembangkan oleh *programmer* bernama Rasmus Lerdoft, PHP awalnya adalah singkatan dari *Personal Home Page Tools*. Namun setelah dikembangkan oleh Zeev Suraski dan Andi Gutmans. Dan fiturnya bertambah, maka PHP diubah singkatannya menjadi sekarang ini. PHP merupakan jenis bahasa scripting yang lazim digunakan di halaman *web*. Artinya, kode ini langsung dimasukkan kedalam kode HTML[9].



Gambar 2. 2 Logo PHP

### 2.2.3 MySQL

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis di bawah lisensi *General Public License* (GPL). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh *user* maupun program-program aplikasinya[10].



Gambar 2. 3 Logo MySQL

### 2.2.4 XAMPP

XAMPP merupakan sebuah aplikasi perangkat lunak

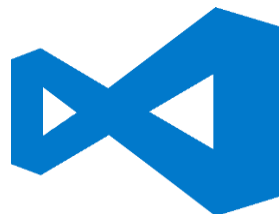
pemrograman dan database yang didalamnya terdapat berbagai macam aplikasi pemrograman seperti; Apache HTTP Server, database MySQL, bahasa pemrograman PHP serta Perl[11].



Gambar 2. 4 Logo XAMPP

### 2.2.5 *Visual Studio Code*

*Visual Studio Code* adalah sebuah teks *editor* ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks *editor* ini secara langsung 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[12].



Gambar 2. 5 *Visual Studio Code*

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

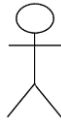


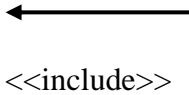
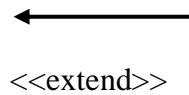
*Unified Modeling Language* (UML) merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan



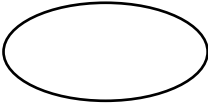

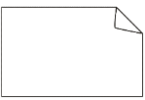


pembuatan sebuah *software* yang berorientasikan pada objek[13]. UML merupakan sebuah standar penulisan atau semacam *blue print* dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik. Terdapat beberapa diagram UML yang sering digunakan dalam pengembangan sebuah sistem, yaitu:

1. *Use Case*: Merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan merepresentasikan sebuah interaksi antara aktor dan sistem. Didalam *use case* terdapat *actor* yang merupakan sebuah gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem.

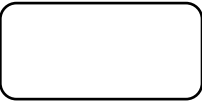



Tabel 2. 1 Simbol *Use Case*

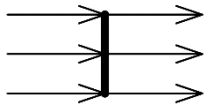
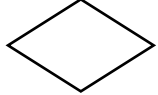
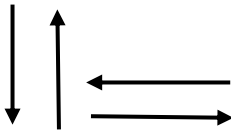
Gambar	Nama	Keterangan
	<i>Actor</i>	Untuk menentukan peran yang dimainkan oleh pengguna saat bekerja dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada unsur-unsur yang berdiri sendiri.
	<i>Generalization</i>	Hubungan yang mana objek turunan berbagi perilaku dan struktur data dengan objek induk.
	<i>Include</i>	Berikan contoh eksplisit tentang cara menggunakan sumber ini.
	<i>Extend</i>	Menetapkan bahwa kasus penggunaan target memperluas perilaku kasus penggunaan sumber tertentu.

<b>Gambar</b>	<b>Nama</b>	<b>Keterangan</b>
	<i>Association</i>	Sesuatu yang menghubungkan suatu objek dengan objek lainnya.
	<i>System</i>	Menentukan paket mana yang ditampilkan sistem dengan batasan.
	<i>Use case</i>	Deskripsi rangkaian tindakan yang dilakukan oleh suatu sistem yang menghasilkan hasil yang terukur bagi para aktor.
	<i>Collaboration</i>	Suatu interaksi di mana aturan-aturan dan unsur-unsur lain bekerja sama untuk menghasilkan suatu perilaku (sinergi) yang lebih besar daripada jumlah bagian-bagiannya.
	<i>Note</i>	Item fisik yang mewakili sumber daya komputer yang ada saat aplikasi sedang berjalan.

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




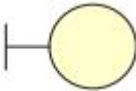

Tabel 2. 2 Simbol *Activity Diagram*





<b>Gambar</b>	<b>Nama</b>	<b>Keterangan</b>
	<i>Activity</i>	Menunjukkan bagaimana setiap kelas antarmuka berinteraksi.
	<i>Action</i>	Keadaan sistem yang mencerminkan pelaksanaan tindakan.
	<i>Initial Node</i>	Bagaimana objek dibuat atau dimulai.
	<i>Final Node</i>	Bagaimana benda diciptakan dan dimusnahkan.

Gambar	Nama	Keterangan
	<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 menampilkan aktivitas yang berjalan secara paralel atau untuk menggabungkan dua aktivitas paralel menjadi satu.

3. *Sequence Diagram*: Menggambarkan interaksi antar objek didalam dan di sekitar sistem yang berupa *message* yang digambarkan terhadap waktu.

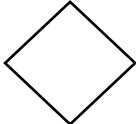


Tabel 2. 3 Simbol *Sequence Diagram*


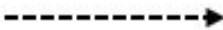

Gambar	Nama	Keterangan
	<i>Lifeline</i>	Objek kesatuan, antarmuka yang berinteraksi satu sama lain.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
	<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
	<i>Boundary Class</i>	Menggambarkan penggambaran dari form.
	<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.

Gambar	Nama	Keterangan
	<i>Control Class</i>	Menggambarkan penghubung antara <i>Boundary</i> dengan tabel.
	<i>Activation</i>	Sebagai sebuah objek yang akan melakukan sebuah aksi.
	<i>Message</i>	Mengindikasikan komunikasi antara objek dengan objek.
	<i>Self Mesagge</i>	Mengindikasikan komunikasi kembali kedalam sebuah objek itu sendiri.

4. *Class Diagram*: Merupakan gambaran struktur dan deskripsi dari *class*, *package*, dan *object* yang saling berhubungan seperti diantaranya pewarisan, asosiasi dan lainnya.

Tabel 2. 4 Simbol 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 (ancestor).
	<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

Gambar	Nama	Keterangan
		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.