

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tinjauan Teori Mengenai Topik Penelitian**

##### **2.1.1 Rancang Bangun**

Rancang bangun dapat diartikan sebagai segala kegiatan yang melibatkan perancangan, pengembangan, dan implementasi hasil analisis kebutuhan untuk menjadi sebuah gambaran atau bentuk sketsa yang belum pernah dibuat sebelumnya [12]. Dalam konteks perancangan sistem informasi, rancang bangun merujuk pada perancangan dan pembangunan suatu desain sistem informasi baru atau merupakan perbaikan dan hasil modifikasi dari sistem yang sudah ada. Tujuan dari rancang bangun sistem informasi adalah untuk menciptakan infrastruktur penyebaran informasi yang efektif dan efisien dalam perusahaan, serta mempermudah pengguna dalam melaksanakan operasi bisnis [13].

##### **2.1.2 Website**

*Website* adalah sekumpulan halaman *web* yang terdapat dalam sebuah domain, di mana setiap halamannya saling terkait dan mengandung informasi yang berharga [14]. Untuk dapat mengakses sebuah *website* diperlukan koneksi internet, perangkat keras (*hardware*), dan perangkat lunak (*software*) yang memadai. *Website* biasanya berisi informasi atau konten dalam bentuk teks, gambar, video, atau elemen multimedia lainnya. Penggunaan *website* diperlukan dalam berbagai keperluan, mulai dari media penyampaian informasi, tempat penjualan produk dan jasa, atau sebagai sarana interaksi antara pengguna dan perusahaan. Saat ini, penggunaan dari *website* telah menjadi bagian yang kritical dalam

aktivitas bisnis, perancangan sistem informasi, sarana komunikasi, dan kegiatan sosial di era digital [15].

### 2.1.3 *Inventory Management*

*Inventory Management* merupakan salah satu bagian dari proses *Supply Chain Management* (SCM), yang berperan dalam merencanakan, mengimplementasikan, dan mengendalikan aliran penyimpanan barang dalam inventaris perusahaan [16]. Tujuan utama dalam penggunaan *Inventory Management* adalah untuk mengoptimalkan level persediaan dengan mempertimbangkan faktor-faktor seperti permintaan pelanggan, biaya penyimpanan, dan risiko kekurangan atau kelebihan persediaan. Salah satu proses yang merupakan cakupan dari *Inventory Management* adalah proses pemantauan inventaris atau *Inventory Monitoring* [17]. *Inventory Monitoring* merupakan salah satu bagian penting dalam penerapan *Inventory Management*, di mana proses yang dilakukan berfokus pada pemantauan inventaris secara akurat dan *real time*. Berikut ini adalah kontribusi dari penerapan *Inventory Monitoring* pada perusahaan industri manufaktur obat [18].

#### 1. **Visibilitas Stok Obat secara *Real Time***

*Inventory Monitoring* memungkinkan perusahaan industri manufaktur obat untuk memantau persediaan secara *real time*. Dengan memiliki visibilitas stok obat secara *real time*, membuat perusahaan dapat mengimplementasikan sistem kontrol yang lebih kuat untuk mencegah terjadinya *fraud*. Hal ini meliputi penggunaan otentikasi dan otorisasi dalam mengakses dan memproses transaksi inventaris. Dengan demikian, perusahaan dapat meningkatkan keandalan dan keamanan inventaris, serta mengurangi risiko kecurangan dalam pengelolaan persediaan obat.

## **2. Meningkatkan Kontrol Stok Obat Kedaluwarsa**

Dalam perusahaan industri manufaktur obat, memiliki persediaan produk yang kedaluwarsa dapat mengakibatkan kerugian finansial yang signifikan. Dengan *Inventory Monitoring*, perusahaan dapat dengan mudah memantau stok obat yang akan segera kedaluwarsa dan dapat mengambil langkah-langkah yang diperlukan untuk menghindari stok obat kedaluwarsa. Penggunaan *Inventory Monitoring* juga dapat memberikan kemudahan bagi manajemen inventaris untuk mengetahui lokasi dari stok obat yang akan kedaluwarsa, sehingga dapat ditempatkan di tempat penyimpanan strategis.

## **3. Pemantauan Transaksi *Inventory* dengan Otorisasi**

Dengan menggunakan *Inventory Monitoring*, memungkinkan perusahaan untuk melibatkan otorisasi dalam pemantauan transaksi *inventory*. Penggunaan otorisasi dapat membuat perusahaan memiliki kontrol yang lebih baik terhadap pergerakan dan penggunaan stok obat, mencegah kerugian atau kehilangan yang tidak terduga, dan memastikan kepatuhan terhadap kebijakan dan prosedur yang ditetapkan dalam manajemen inventaris. Otorisasi dapat dilakukan oleh berbagai pihak, seperti manajer departemen terkait, atau atasan langsung.

## **4. Penyediaan Informasi untuk Pengambilan Keputusan**

*Inventory Monitoring* menyediakan data yang akurat dan terkini mengenai kondisi persediaan. Hal ini memungkinkan perusahaan untuk melakukan analisis persediaan dengan lebih cepat dan tepat, sehingga dapat mengoptimalkan proses pengambilan keputusan dan meningkatkan efisiensi proses bisnis.


## 2.1.4 Unified Modeling Language (UML)



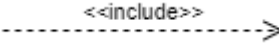
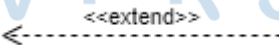
*Unified Modeling Language* atau yang biasa disebut UML adalah sebuah bahasa yang digunakan untuk membantu menggambarkan dan mendokumentasikan desain, struktur, dan perilaku sistem perangkat lunak [19]. Dalam penerapannya, UML terdiri dari sejumlah simbol penulisan yang dapat membantu untuk menjelaskan atau mendeskripsikan elemen - elemen sistem perangkat lunak beserta hubungannya. Penyusunan simbol dan notasi UML dapat diklasifikasikan menjadi berbagai jenis diagram. Menurut Peker, UML dapat diklasifikasikan menjadi 11 jenis diagram yang berbeda, namun pada penelitian ini hanya akan digunakan 3 jenis diagram UML yaitu *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram* [20].

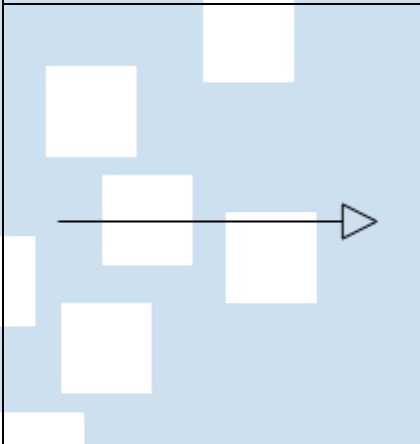
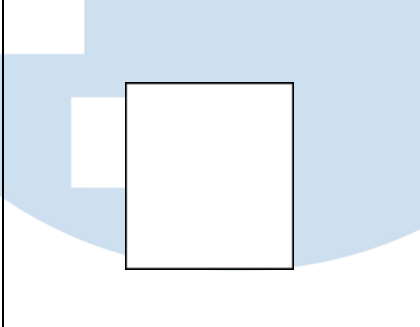
### 2.1.4.1 Use Case Diagram

*Use Case Diagram* merupakan salah satu jenis diagram UML yang bertujuan untuk menampilkan interaksi antara aktor (*user*) dan fitur-fitur (*use case*) dalam sistem perangkat lunak [20]. Pada penerapannya, terdapat notasi dan simbol yang digunakan untuk memvisualisasikan sebuah *Use Case Diagram*. Berikut ini adalah 7 notasi yang digunakan dalam *Use Case Diagram*.

Tabel 2. 1 Notasi *Use Case Diagram*

Notasi	Keterangan
	<i>Actor</i> : merupakan orang, sistem, atau perangkat lain yang berinteraksi dengan sistem.





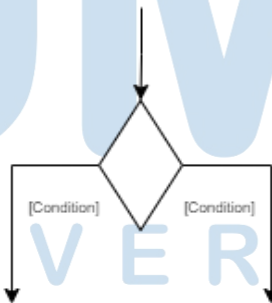
	<p><i>Use Case</i>: merupakan fungsi atau tugas yang dilakukan oleh sistem.</p>
	<p><i>Association Relationship</i>: merupakan salah satu jenis hubungan dalam <i>Use Case Diagram</i>, yang digunakan untuk menghubungkan <i>Actor</i> dan <i>Use Case</i>.</p>
	<p><i>Include Relationship</i>: merupakan salah satu jenis hubungan dalam <i>Use Case Diagram</i>, yang digunakan untuk menunjukkan bahwa suatu <i>Use Case</i> melibatkan penggunaan atau eksekusi dari <i>Use Case</i> lain.</p>
	<p><i>Extend Relationship</i>: merupakan salah satu jenis hubungan dalam <i>Use Case Diagram</i>, yang digunakan untuk menunjukkan bahwa suatu <i>Use Case</i> bisa diperpanjang (<i>extended</i>)</p>

	dengan <i>Use Case</i> lain. tergantung pada kondisi atau situasi tertentu.
	<i>Generalization Relationship</i> : merupakan salah satu jenis hubungan dalam <i>Use Case Diagram</i> , yang digunakan untuk menunjukkan batasan-batasan sebuah sistem.
	<i>System</i> : merupakan simbol berbentuk kotak yang digunakan untuk merepresentasikan sistem atau aplikasi yang digunakan.

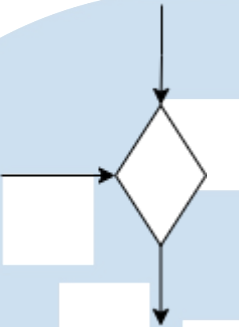
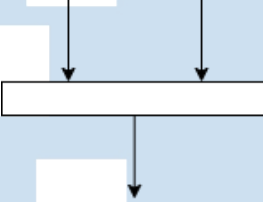

#### 2.1.4.2 Activity Diagram

*Activity Diagram* merupakan salah satu jenis diagram UML yang digunakan untuk memberikan visualisasi mengenai alur kerja atau proses bisnis dalam sistem [20]. Diagram ini digunakan untuk menggambarkan alur kerja atau serangkaian aktivitas yang terjadi dalam suatu proses atau sistem. *Activity Diagram* menunjukkan urutan aktivitas, keputusan yang dibuat selama proses, dan aliran kontrol antara aktivitas-aktivitas tersebut. Pada penerapannya, terdapat notasi dan simbol yang digunakan untuk memvisualisasikan sebuah *Activity Diagram*. Berikut ini adalah 8 notasi yang digunakan dalam *Activity Diagram*.

Tabel 2. 2 Notasi Activity Diagram

Notasi	Keterangan
	<p><i>Initial Node</i>: merupakan titik awal dari aktivitas diagram dan digambarkan dengan simbol lingkaran berisi panah yang menunjuk ke <i>node</i> pertama.</p>
	<p><i>Action</i>: merupakan suatu kegiatan atau tindakan yang dilakukan.</p>
	<p><i>Control Flow</i>: merupakan simpul yang digunakan untuk menyatakan transisi dari satu aktivitas ke aktivitas yang lain.</p>
	<p><i>Fork Node</i>: merupakan simpul yang digunakan untuk memecah alur aktivitas menjadi beberapa jalur.</p>
	<p><i>Decision Node</i>: merupakan sebuah simbol berbentuk <i>diamond</i> yang digunakan untuk proses pembuatan keputusan dalam <i>Activity Diagram</i>.</p>



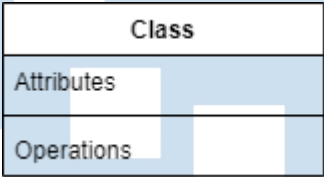



	<p><i>Merge Node</i>: merepresentasikan titik di mana beberapa jalur di dalam aktivitas diagram bertemu. Simbol <i>diamond</i> dengan dua garis masuk dan satu garis keluar digunakan untuk merepresentasikan simpul penggabungan.</p>
	<p><i>Join Node</i>: digunakan untuk menggabungkan beberapa jalur ke dalam satu jalur. Simbol tiga garis lurus yang bertemu di satu simpul digunakan untuk merepresentasikan simpul penggabungan.</p>
	<p><i>Final Node</i>: merepresentasikan titik akhir dari aktivitas diagram dan digambarkan dengan simbol lingkaran yang tidak memiliki panah keluar.</p>


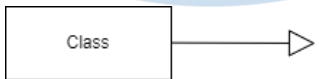

### 2.1.4.3 Class Diagram

*Class Diagram* merupakan salah satu jenis diagram UML yang digunakan untuk mendeskripsikan struktur dari kelas-kelas atau entitas yang ada dalam sistem [20]. Pada penerapannya, terdapat notasi dan simbol yang digunakan untuk memvisualisasikan sebuah *Class Diagram*. Berikut ini adalah 7 notasi yang umumnya digunakan dalam *Class Diagram*.



**Tabel 2. 3** Notasi *Class Diagram*

Notasi	Keterangan
	<p><i>Class</i>: merupakan sebuah kumpulan objek yang memiliki karakteristik dan perilaku yang sama. Kelas digambarkan sebagai sebuah persegi panjang terbagi menjadi tiga bagian yaitu nama kelas, atribut, dan operasi.</p>
	<p><i>Association Relationship</i>: merupakan relasi dasar yang menggambarkan hubungan antara dua kelas. Relasi ini menunjukkan bahwa suatu objek dari kelas yang satu dapat menggunakan objek dari kelas yang lain.</p>
	<p><i>Aggregation Relationship</i>: Merupakan relasi di mana suatu objek dari kelas yang satu terdiri dari objek-objek dari kelas yang lain.</p>
	<p><i>Composition Relationship</i>: merupakan bentuk khusus dari relasi agregasi, di mana objek dari kelas yang satu</p>

	adalah bagian dari objek dari kelas yang lain.
	<p><i>Dependency Relationship:</i> merupakan relasi di mana suatu kelas menggunakan kelas lain sebagai parameter dalam <i>method</i> atau sebagai tipe variabel. Relasi ini menunjukkan bahwa perubahan pada kelas yang satu dapat memengaruhi kelas yang lain.</p>
	<p><i>Inheritance Relationship:</i> merupakan relasi di mana suatu kelas mengambil atribut dan <i>method</i> dari kelas lain. Relasi ini menunjukkan bahwa suatu kelas (<i>subClass</i> atau <i>child Class</i>) adalah turunan dari kelas lain (<i>superClass</i> atau <i>parent Class</i>).</p>
	<p><i>Realization Relationship:</i> merupakan relasi yang menunjukkan bahwa suatu kelas mengimplementasikan satu atau beberapa <i>method</i> yang didefinisikan dalam <i>interface</i>. Relasi ini sering digunakan dalam bahasa</p>

	<p>pemrograman yang mendukung konsep pemrograman berorientasi objek dan <i>interface</i> seperti Java.</p>
--	--

### 2.1.5 Metode FEFO

Metode FEFO (*First Expired First Out*) merupakan salah satu metode manajemen persediaan atau yang berfokus pada prinsip bahwa barang dengan tanggal kedaluwarsa paling dekat harus digunakan atau dijual lebih dahulu [11]. Dalam metode ini, barang-barang yang memiliki tanggal kedaluwarsa mendekati atau melewati batas waktu akan diutamakan penggunaannya atau penjualannya sebelum barang-barang dengan tanggal kedaluwarsa yang lebih jauh. Penerapan metode FEFO dalam pengelolaan persediaan memastikan bahwa produk dengan tanggal kedaluwarsa mendekati dikeluarkan dari inventaris terlebih dahulu, sehingga dapat mengurangi risiko pemborosan dan kerugian akibat barang yang kadaluwarsa [21]. Selain itu, dengan penggunaan metode FEFO juga memungkinkan perusahaan untuk mengelola persediaan dengan lebih efisien, meminimalkan pemborosan, dan menjaga kualitas produk yang dihasilkan atau dijual [22].

### 2.1.6 Metode Pengembangan Perangkat Lunak

#### 2.1.6.1 Metode Waterfall

Waterfall merupakan salah satu metode pengembangan perangkat lunak dalam *Software Development Life Cycle* (SDLC) yang pertama kali diperkenalkan [23]. Metode Waterfall menggunakan pendekatan linear dan berurutan dalam perancangan perangkat lunak, di mana setiap fase bergantung pada selesainya fase sebelumnya. Langkah

pengembangan perangkat lunak menurut metode Waterfall antara lain, *Requirement Analysis, Design, Implementation, Testing*, dan *Maintenance* [23].

#### **2.1.6.2 Metode RAD**

Metode RAD (*Rapid Application Development*) merupakan sebuah model pengembangan perangkat lunak yang menekankan pada kecepatan dan fleksibilitas [23]. Metode RAD memprioritaskan pengembangan iteratif dan kolaborasi antara pengembang perangkat lunak dan pengguna. Tujuannya adalah untuk menghasilkan aplikasi yang dapat dikembangkan dengan cepat dan dapat disesuaikan dengan kebutuhan yang berubah. Dengan pendekatan RAD, pengembang dapat menghasilkan aplikasi dengan cepat dan responsif terhadap perubahan kebutuhan. Metode ini sangat cocok untuk digunakan dalam proyek-proyek yang membutuhkan waktu pengembangan yang singkat, memiliki kebutuhan yang dapat berubah-ubah, serta memerlukan keterlibatan aktif dari pengguna. Langkah pengembangan perangkat lunak menurut metode RAD antara lain, *Requirement Planning, Design System*, dan *Implementation* [23].

#### **2.1.6.3 Metode Prototype**

Metode Prototype merupakan sebuah model pengembangan perangkat lunak yang melibatkan pembuatan prototipe awal dari aplikasi sebelum pembuatan versi akhirnya [23]. Prototipe dibuat sebagai bentuk representasi dari sistem informasi yang direncanakan, dengan menunjukkan fungsionalitas utama dan antarmuka pengguna. Metode Prototype memungkinkan pengembang dan pengguna berinteraksi lebih awal dalam proses

pengembangan, sehingga memungkinkan perubahan dan perbaikan yang cepat berdasarkan umpan balik pengguna. Pendekatan ini membantu memastikan aplikasi yang dikembangkan memenuhi kebutuhan pengguna dan mencapai tingkat yang diharapkan sebelum diluncurkan secara resmi. Tahapan dalam metode prototipe antara lain, *Requirement Identification, Prototype Development, Evaluation, Iterative Development, Implementation, dan Final Evaluation.*

### **2.1.7 Black Box Testing**

*Black Box Testing* adalah salah satu metode pengujian perangkat lunak yang berfokus pada verifikasi fungsionalitas perangkat lunak, di mana pengujian dilakukan tanpa melihat struktur *internal* maupun kode pemrograman dari perangkat lunak yang diuji [24]. Pengujian *Black Box Testing* dilakukan dengan cara mencoba memberikan *input* ke dalam perangkat lunak dan melakukan pengecekan apakah *output* yang dihasilkan sesuai dengan yang diharapkan. Langkah ini akan terus dilakukan secara berulang hingga perangkat lunak dinilai telah memenuhi persyaratan fungsional yang telah ditentukan sebelum pengujian. Setelah perangkat lunak sudah berhasil melalui pengujian ini dan memenuhi persyaratan fungsionalitas, maka perangkat lunak kemudian dapat dinyatakan layak untuk digunakan dan siap untuk diuji lebih lanjut lagi [25].

### **2.1.8 User Acceptance Testing**

*User Acceptance Testing* (UAT) adalah proses pengujian yang dilakukan untuk mengevaluasi apakah sistem atau perangkat lunak yang dikembangkan telah memenuhi persyaratan dan kebutuhan pengguna secara keseluruhan [26]. UAT dilakukan oleh pengguna akhir atau pihak yang mewakili pengguna akhir, dengan tujuan untuk memastikan bahwa sistem telah siap digunakan sebelum

dirilis secara resmi. Hasil pengujian UAT akan memberikan umpan balik kepada tim pengembang tentang keberhasilan sistem atau perangkat lunak dalam memenuhi harapan pengguna. Jika terdapat masalah atau kekurangan, tim pengembang akan mengambil tindakan perbaikan yang diperlukan sebelum merilis produk tersebut secara resmi.

## **2.2 Framework yang Digunakan**

### **2.2.1 Laravel**

Laravel adalah sebuah *framework open-source* berbasis *web* yang ditulis dengan bahasa pemrograman PHP, tidak berbayar, dan diperuntukkan untuk pengembangan perangkat lunak yang menggunakan pendekatan *Model-View-Controller* (MVC) [27]. Dalam penerapannya, struktur pola MVC pada *framework* Laravel sedikit berbeda dibandingkan dengan struktur pola MVC pada umumnya. Struktur pola MVC dalam Laravel mempunyai *routing* yang menjembatani antara *request* dari *user* dan *controller*, sehingga membuat *controller* tidak langsung menerima *request* tersebut. Selain itu, Laravel juga dilengkapi dengan fitur-fitur canggih seperti ORM (*Object Relational Mapping*), *templating engine*, fitur keamanan, dan *routing* [28]. Dengan menggunakan *framework* Laravel, pengembang dapat memproses secara lebih cepat pembuatan aplikasi *web* dengan kode yang lebih bersih, modular, dan mudah dipelihara.

## **2.3 Tools yang Digunakan**

### **2.3.1 Visual Studio Code**

Visual Studio Code atau yang biasa disebut VS Code merupakan sebuah perangkat lunak pengolah kode yang dikembangkan oleh Microsoft. VS Code dirancang secara khusus untuk mendukung pengembangan perangkat lunak secara efisien dengan menyediakan

fitur-fitur yang kuat dan fleksibel. Salah satu yang menjadi keunggulan utama dari VS Code adalah ekstensibilitas. Pengguna dapat menginstal ekstensi dari marketplace untuk menambahkan fitur-fitur tambahan sesuai dengan kebutuhan [29]. VS Code juga menyediakan lingkungan pengembangan terintegrasi (*Integrated Development Environment/IDE*) dengan sejumlah fitur yang dapat membantu pengguna dalam menulis, mengubah, dan mengelola kode dengan mudah.

### 2.3.2 MySQL

MySQL adalah sebuah sistem manajemen *database* relasional (RDMS) berbasis *open-source*, yang sangat populer untuk digunakan dalam pengembangan aplikasi *web* [30]. Untuk dapat mengoperasikan MySQL, dibutuhkan bahasa pemrograman SQL (*Structured Query Language*). SQL sudah dikenal luas oleh kebanyakan pengembang karena digunakan untuk dapat melakukan operasi dalam MySQL. Dengan SQL memungkinkan pengguna untuk dapat mengambil, memperbaharui, dan menghapus data yang ada di dalam *database*. MySQL sudah mendukung protokol-protokol seperti TCP/IP dan HTTP, sehingga sangat mudah untuk mengintegrasikannya ke aplikasi *web*. Selain itu MySQL juga bersifat sangat fleksibel dalam pengembangan aplikasi *web*, karena mampu berinteraksi dengan berbagai *platform* sistem operasi, seperti Linux, Windows, dan Mac. Untuk melakukan administrator MySQL secara lebih mudah dan cepat, pengguna dapat menggunakan tambahan perangkat lunak seperti phpMyAdmin dan MySQL Yog [31].

### 2.3.3 XAMPP

XAMPP merupakan sebuah paket perangkat lunak berbasis *open-source* yang lengkap dan dapat digunakan untuk mempelajari pemrograman *web*, khususnya PHP dan MySQL [32]. Tujuan



utama dari XAMPP adalah memudahkan pengembang *web* dalam menginstal, mengkonfigurasi, dan menjalankan aplikasi *web* pada komputer lokal (*localhost*). Dengan XAMPP, pengembang dapat menguji aplikasi *web* secara lokal terlebih dahulu sebelum *web* diunggah ke server publik. XAMPP juga mudah digunakan dan dipelajari, sehingga sangat cocok untuk digunakan pengembang *web* pemula yang ingin belajar dan memulai pengembangan aplikasi *web*. Karena bersifat *open-source*, XAMPP juga dapat diunduh dan digunakan secara gratis. Dalam penerapannya, XAMPP juga terdiri atas beberapa program yang membantu pembangunan aplikasi *web* seperti Apache HTTP Server, MySQL, PHP, dan Perl [33].

#### 2.3.4 HTML

HTML (*HyperText Markup Language*) merupakan salah satu bahasa pemrograman yang digunakan untuk membangun struktur dan menampilkan konten pada halaman *web* [34]. HTML digunakan oleh *web browser* untuk menafsirkan dan menampilkan elemen-elemen halaman *web*. HTML menggunakan sejumlah *tag* atau elemen untuk mengatur dan memformat konten pada halaman *web*. Setiap *tag* memiliki sintaksis yang spesifik dan mampu memberikan instruksi kepada *web browser* tentang bagaimana konten tersebut harus ditampilkan. *Tag* HTML juga dapat digunakan untuk menyusun elemen seperti teks, gambar, tautan, tabel, formulir, video, dan lainnya.

#### 2.3.5 CSS

CSS (*Cascading Style Sheets*) merupakan salah satu bahasa pemrograman yang digunakan untuk mengatur tampilan, tata letak, dan *styling* dari elemen-elemen pada halaman *web* [35]. CSS pertama kali diperkenalkan pada tahun 1966 sebagai bagian dari spesifikasi HTML 3.0. Dalam pengembangan *web*, CSS bekerja bersama dengan HTML untuk mengendalikan gaya dan tampilan

dari elemen-elemen pada halaman *web*, sehingga halaman terlihat lebih menarik. CSS memisahkan tampilan dari struktur dan konten halaman *web*, sehingga memungkinkan pengembang untuk dengan mudah mengubah tampilan global dari beberapa halaman dengan *mengedit* file CSS tunggal.

### 2.3.6 JavaScript

JavaScript merupakan sebuah bahasa pemrograman berbentuk kumpulan *script* yang berjalan pada suatu dokumen HTML [36]. JavaScript memiliki syntax yang dinilai cukup serupa dengan bahasa pemrograman C dan C++, perbedaannya terletak pada fitur-fitur unik yang dimiliki seperti *closure*, *prototype*, dan *event loop*. Dalam penerapannya, JavaScript biasanya digunakan untuk menyempurnakan tampilan dan sistem pada halaman aplikasi berbasis *web* yang dikembangkan. JavaScript juga digunakan untuk membuat interaksi antara pengguna dan halaman *web*, seperti animasi, validasi form *input*, efek visual, dan lain-lain. Selain itu, JavaScript juga digunakan untuk membuat aplikasi *web* yang lebih kompleks seperti aplikasi berbasis *single-page* (SPA) dan aplikasi berbasis RESTful API [37].

### 2.3.7 PHP

PHP (Hypertext Preprocessor) adalah sebuah bahasa pemrograman berbasis *open-source* yang terpasang pada HTML dan digunakan untuk pengembangan *website* [38]. PHP dirancang khusus untuk memproses kode pada sisi server (*server-side*), dan juga dapat digunakan untuk membuat tampilan *web* menjadi lebih dinamis. Sebagian besar dari sintak PHP sebenarnya mirip dengan bahasa pemrograman C, Java, ASP, dan Perl, hanya saja berbeda penggunaannya dalam pengembangan aplikasi *web*. Selain itu, PHP juga sangat fleksibel karena sudah dapat terhubung dan digunakan dalam *framework* dan *database*. Untuk *framework*, PHP dapat

digunakan dalam *framework* CodeIgniter, Laravel, dan Symfony. Untuk *database* PHP dapat berinteraksi dengan *database* walaupun dengan kelengkapan yang berbeda seperti DBM, MySQL, dan Oracle.

## 2.4 Penelitian Terdahulu

Pada bagian ini dijabarkan kumpulan artikel penelitian terdahulu yang telah dilakukan dan digunakan sebagai referensi dalam menyelesaikan penyusunan penelitian ini.

Tabel 2. 4 Penelitian Terdahulu

No	Nama Jurnal	Judul Artikel & Penulis	Metode	Hasil
1.	JSiI (JSiI (Jurnal Sistem Informasi), 2022, 9(1), pp. 27-35.	<p><b>Nama Artikel:</b> <i>Design and Build Inventory Management Information System Using the Scrum Method</i></p> <p><b>Nama Penulis:</b> Ilham Firman Ashari, Annisa Jufe Aryani, Alief Moehamad Ardhi.</p>	Scrum (Agile)	Sistem informasi <i>inventory</i> yang dihasilkan mampu mengumpulkan data barang masuk dan keluar secara <i>real time</i> .
2.	Jurnal SISFOKOM (Sistem Informasi dan Komputer), 10(3), 419-425	<p><b>Nama Artikel:</b> Rancang Bangun Sistem Informasi Manajemen Arsip Digital</p> <p><b>Nama Penulis:</b> Anisah, Delpiah Wahyuningsih, Ellya Helmud, Tedy Suwanda, Parlia Romadiana, dan Devi Irawan.</p>	Waterfall	Sistem informasi yang dihasilkan dapat memudahkan penelusuran dan <i>monitoring</i> terhadap arsip yang ada.
3.	Jurnal Sisfokom	<p><b>Nama Artikel:</b></p>	Waterfall	Sistem informasi yang dihasilkan dapat

	(Sistem Informasi Dan Komputer), 9(3), 365-372.	Rancang Bangun Sistem Informasi Manajemen Proyek Berbasis Web Pada PT Seatech Infosys  <b>Nama Penulis:</b> Doni Darmawan dan Anita Ratnasari.		digunakan untuk melakukan pembagian tugas dan <i>monitoring</i> proyek yang sedang dikerjakan oleh PT Seatech Infosys.
4.	JITK (Jurnal Ilmu Pengetahuan Dan Teknologi Komputer), 5(1), 15-22.	<b>Nama Artikel:</b> Rancang Bangun Sistem Informasi Akuntansi Penjualan dengan Model Waterfall Berbasis Java Desktop  <b>Nama Penulis:</b> Muthia, Nurul, Hilda Amalia, Ari Puspita, dan Ade Fitria Lestari.	Waterfall	Sistem informasi yang dihasilkan mampu menyajikan laporan penjualan dengan cepat dan tepat, sehingga dapat membantu pemilik dalam proses pengambilan keputusan.
6.	JIMP (Jurnal Informatika Merdeka Pasuruan), 4(1).	<b>Nama Artikel:</b> Rancang Bangun Sistem Informasi Persediaan Obat Pada Puskesmas Babelan I Kabupaten Bekasi  <b>Nama Penulis:</b> Miwan Kurniawan Hidayat.	Waterfall	Sistem informasi <i>inventory</i> yang dihasilkan dapat memberikan kelancaran dalam penginputan data, sehingga mampu memberikan informasi yang cepat dan akurat.
7.	Voteteknika (Vocational Teknik Elektronika dan Informatika), 9(4), 55-63.	<b>Nama Artikel:</b> Rancang Bangun Sistem Informasi Inventory Barang pada Master Dealer CV. Orbit Techno Regional Sentral Sumatra  <b>Nama Penulis:</b> Syahisro Mirajdandi, Dedy Irfan, dan Agariadne Dwinggo Samala.	Waterfall	Sistem informasi <i>inventory</i> yang dihasilkan dapat mempermudah proses pengelolaan dan pelaporan data barang, dari yang sebelumnya dilakukan manual menjadi berbasis digital.
8.	Jurnal Tekno Kompak, 17(1), 67-80.	<b>Nama Artikel:</b> Rancang Bangun Sistem Informasi	Waterfall	Sistem informasi <i>inventory</i> yang dihasilkan

		Inventory Barang Menggunakan Metode First In First Out  <b>Nama Penulis:</b> Nunu Nugraha Purnawan dan Slamet Rahayu. (2023)		menggunakan metode FIFO, dan mampu mempermudah proses permintaan barang, dan mengatasi kehilangan pencatatan data barang masuk dan keluar.
9.	JSiI (Jurnal Sistem Informasi), 8(1), 31-42.	<b>Nama Artikel:</b> Perancangan Sistem Informasi Persediaan Barang Pada Balai Besar Pengawas Obat Dan Makanan  <b>Nama Penulis:</b> Rehulina Tarigan, Budhy Raharjo.	Waterfall	Sistem informasi <i>inventory</i> berbasis <i>website</i> yang dihasilkan mampu mempercepat proses pembuatan laporan stok obat.
10.	Jurnal Ilmiah Matrik, 21(1), 1-10.	<b>Nama Artikel:</b> Pengembangan Sistem Informasi untuk Memfasilitasi Proses Adopsi Anjing Berbasis Web  <b>Nama Penulis:</b> Amelinda Chendra, Kristina G. Simanjuntak, Andree E. Widjaja, dan Suryasari.	RAD	Sistem informasi yang dihasilkan dapat membantu mengurangi kasus penelantaran anjing.

Tabel 2.4 merupakan kumpulan penelitian terdahulu yang digunakan sebagai referensi dari penyusunan penelitian ini. Berdasarkan penelitian terdahulu, peneliti melakukan eksplorasi mengenai metode pengembangan perangkat lunak yang digunakan. Kebanyakan penelitian terdahulu menggunakan metode Waterfall dalam melakukan pengembangan sistem informasi [39]–[45]. Dalam melakukan pengembangan sistem informasi *inventory*, juga ditemukan penelitian terdahulu yang membahas penggunaan metode manajemen inventaris FIFO (*First Expired First Out*) dalam

membantu menyelesaikan permasalahan pengelolaan stok barang dalam inventaris [44].

Dalam melakukan pengembangan sistem informasi *inventory*, penelitian ini mengadaptasi metode pengembangan perangkat lunak yang digunakan pada penelitian terdahulu. Pengembangan sistem informasi *inventory* dilakukan dengan mengikuti tahapan dalam metode Waterfall, dan menggunakan bahasa pemrograman HTML, CSS, dan PHP. Perbedaan yang cukup signifikan antara penelitian ini dengan penelitian terdahulu terletak pada metode manajemen inventaris yang digunakan. Pada penelitian ini digunakan metode manajemen inventaris FEFO (*First Expired First Out*) dalam melakukan perancangan sistem informasi *inventory*. Metode FEFO digunakan pada sistem informasi *inventory* dalam proses pemilihan produk yang akan dijual, sehingga dapat membantu dalam proses pengelolaan stok obat kedaluwarsa. Pada penelitian ini juga dilakukan pengembangan sistem informasi *inventory* berdasarkan teori *Inventory Management*, yang penerapannya berfokus pada *Inventory Monitoring*.

