

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Sistem Penyimpanan**

Sistem penyimpanan merupakan bagian integral dari proses pengolahan data barang yang berlokasi di dalam sebuah gudang. Sistem penyimpanan memiliki dampak yang signifikan pada suatu instansi, karena mampu memberikan kontribusi dalam menyelesaikan berbagai tantangan dalam pengolahan data barang, serta mempermudah proses pelaporan data barang yang tersedia [12].

#### **2.2 Penyimpanan Barang**

Penyimpanan barang merupakan kegiatan penempatan barang di dalam gudang sebelum melalui proses selanjutnya. Umumnya, aktivitas penyimpanan barang dilakukan berdasarkan karakteristik unik dari setiap barang, dengan persyaratan utama yang harus terpenuhi seperti keamanan, efisiensi, kemudahan, dan keterjangkauan dalam pencarian barang [13].

#### **2.3 HTML**

HTML merupakan bahasa pemrograman yang memiliki fleksibilitas dalam berbagai cara untuk menciptakan halaman website. Konten halaman website dapat berupa teks, grafik, atau media lainnya yang dapat ditampilkan melalui web browser [14].

#### **2.4 PHP**

PHP merupakan bahasa pemrograman yang dipakai untuk pengembangan aplikasi web. PHP digunakan sebagai bahasa server-side scripting, di mana sintaks dan perintah PHP dieksekusi oleh server. Namun, PHP juga dapat disematkan ke dalam halaman HTML biasa, sehingga memungkinkan integrasi antara kode PHP dan kode HTML dalam satu halaman yang sama. Hal ini memungkinkan penggunaan fitur-fitur PHP untuk memanipulasi dan menghasilkan konten dinamis di dalam halaman HTML [15].

## 2.5 Unified Modelling Language (UML)

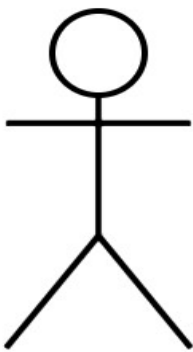
*Unified Modelling Language* adalah sebuah metode pemodelan visual yang digunakan sebagai alat perancangan sistem berorientasi objek. UML juga dapat dianggap sebagai bahasa standar untuk visualisasi, perancangan, dan dokumentasi sistem. UML sering kali disebut sebagai bahasa standar untuk membuat blueprint atau rancangan perangkat lunak. Dalam UML, para pengembang menggunakan notasi dan konvensi yang telah ditetapkan untuk menggambarkan struktur, relasi, dan interaksi antara komponen-komponen dalam sistem. Hal ini memungkinkan mereka untuk mengkomunikasikan desain dan pemahaman sistem secara efektif kepada para stakeholder yang terlibat dalam pengembangan perangkat lunak [16].

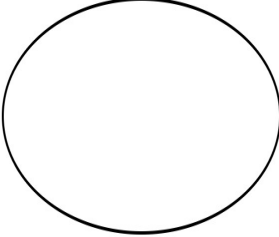

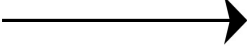
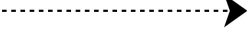
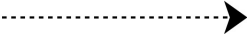
### 2.5.1 Use Case Diagram

*Use Case Diagram* merupakan salah satu diagram dalam UML (Unified Modeling Language) yang menggambarkan interaksi antara sistem dan aktor-aktor yang terlibat. Tujuan dari Use Case Diagram adalah untuk menggambarkan jenis-jenis interaksi antara pengguna sistem (aktor) dengan sistem itu sendiri. Diagram ini memberikan gambaran mengenai bagaimana aktor-aktor berinteraksi dengan sistem dan tindakan-tindakan yang dapat dilakukan dalam konteks penggunaan sistem [16].

Tabel 2.1. Simbol Use Case Diagram

Sumber [16]

Simbol	Keterangan
	<p><b>Actor :</b></p> <ul style="list-style-type: none"><li>• Aktor merujuk kepada individu, perangkat, atau bahkan sistem yang memiliki peran penting dalam menjalankan operasi.</li><li>• Aktor memberikan dan menerima informasi dari sistem.</li></ul>

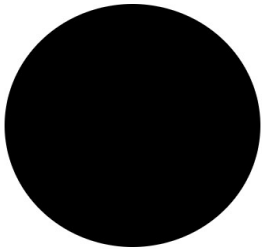

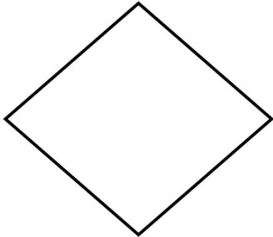
Simbol	Keterangan
	<p><b>Use Case :</b></p> <ul style="list-style-type: none"> <li>• Sistem menyediakan fungsionalitas yang memungkinkan unit-unit atau aktor-aktor saling bertukar pesan.</li> <li>• Dalam use case, fungsionalitas tersebut dinyatakan dengan kata kerja awal.</li> <li>• Menggambarkan fungsi dan kebutuhan dari perspektif pengguna.</li> </ul>
	<p><b>Association :</b></p> <ul style="list-style-type: none"> <li>• Metode yang digunakan untuk mengidentifikasi hubungan antara aktor spesifik dan use case tertentu.</li> </ul>
	<p><b>Generalisation :</b></p> <ul style="list-style-type: none"> <li>• Keterkaitan antara dua use case atau dua aktor.</li> </ul>
<p data-bbox="584 1581 687 1603">&lt;&lt;include&gt;&gt;</p> 	<p><b>Include :</b></p> <ul style="list-style-type: none"> <li>• Menemukan koneksi atau hubungan antara dua use case.</li> </ul>
<p data-bbox="584 1805 687 1827">&lt;&lt;extend&gt;&gt;</p> 	<p><b>Extend :</b></p> <ul style="list-style-type: none"> <li>• Jenis yang hanya berjalan dibawah kondisi tertentu .</li> </ul>


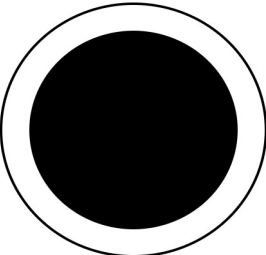
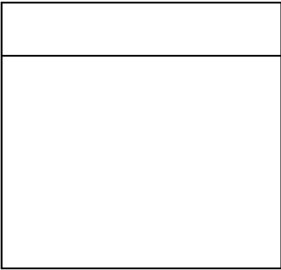
## 2.5.2 Activity Diagram

*Activity diagram* adalah sebuah diagram yang digunakan untuk memodelkan proses-proses yang terjadi dalam suatu sistem. Diagram ini menggambarkan urutan langkah-langkah atau aktivitas-aktivitas yang terjadi dalam sistem secara visual. Runtutan proses dalam sistem direpresentasikan secara vertikal dalam diagram aktivitas [17].

Tabel 2.2. Simbol Activity Diagram

Sumber [17]

Simbol	Keterangan
	<p><b>Initial State :</b></p> <ul style="list-style-type: none"><li>• Sebuah diagram aktivitas awal yang memiliki sebuah status awal.</li></ul>
	<p><b>Aktivitas :</b></p> <ul style="list-style-type: none"><li>• Aktivitas yang sedang dilakukan atau terjadi dalam sistem, umumnya diawali dengan kata kerja yang menggambarkan aktivitas tersebut.</li></ul>
	<p><b>Decision :</b></p> <ul style="list-style-type: none"><li>• Suatu titik atau kondisi dalam sistem, terdapat indikasi adanya kemungkinan transisi yang berbeda. Hal ini penting ketika sistem memiliki beberapa kemungkinan atau jalur alternatif.</li></ul>

Simbol	Keterangan
	<p><b>Join :</b></p> <ul style="list-style-type: none"> <li>Menggabungkan bagian-bagian yang telah dipisahkan menjadi satu alur yang utuh atau lengkap.</li> </ul>
	<p><b>End State :</b></p> <ul style="list-style-type: none"> <li>Notasi akhir digunakan untuk menunjukkan bahwa proses tersebut telah selesai atau berakhir.</li> </ul>
	<p><b>Swimlane :</b></p> <ul style="list-style-type: none"> <li>Activity diagram dapat dipecah menjadi kolom dan baris untuk membagi tanggung jawab objek-objek yang terlibat dalam suatu aktivitas.</li> </ul>

### 2.5.3 Relational Database Management System

*Relational Database Management System* adalah salah satu jenis *Database Management System* yang dirancang untuk menyediakan representasi visual atau diagram skema yang menggambarkan hubungan antara tabel-tabel di dalam sebuah database [18].

*Relational Database Management System* memiliki 4 jenis relasi yaitu :

- *One-To-One* adalah ketika setiap entitas di sebuah tabel dapat berhubungan dengan satu entitas dalam tabel lain pada basis data, dan ini berlaku dua arah atau timbal balik.
- *One-to-Many* adalah ketika setiap entitas dalam sebuah tabel dapat berhubungan atau berelasi dengan banyak entitas dalam tabel lain pada basis data, namun sebaliknya tidak berlaku.

- *Many-to-One* adalah ketika setiap entitas dalam sebuah tabel dapat berhubungan dengan satu entitas dalam tabel lain dalam database, namun sebaliknya tidak berlaku.
- *Many-to-Many* adalah ketika setiap entitas dalam sebuah tabel dapat berhubungan dengan banyak entitas dalam tabel lain pada basis data, dan sebaliknya juga berlaku [19].

## 2.6 *Black Box Testing*

*Black Box Testing* adalah metode pengujian yang dilakukan dengan melihat dan mengamati hasil dari pengeksekusian program serta memeriksa fungsi perangkat lunak [20].

### 2.6.1 Teknik-teknik *Black Box Testing*

Beberapa teknik umum yang digunakan untuk menguji perangkat lunak antara lain:

#### 1. *All pair testing*

Pengujian ini memanfaatkan teknik Pairwise Testing untuk menguji semua kemungkinan kombinasi dari pasangan input parameter yang ada [21].

#### 2. *Boundary value analysis*

Teknik ini berfokus pada pencarian *error* atau masalah dalam perangkat lunak dari perspektif eksternal atau luar ke dalam [22].

#### 3. *Cause-effect graph*

Teknik ini menggunakan grafik yang disebut Fault Tree yang menggambarkan hubungan antara efek yang terjadi dan penyebab dari error atau masalah yang terjadi dalam perangkat lunak [23].

#### 4. *Equivalence partitioning*

Teknik ini bekerja dengan membagi data input dari perangkat lunak menjadi beberapa partisi data yang setara atau setara secara fungsional [24].

#### 5. *Fuzzing*

Teknik untuk menemukan *bug* dalam perangkat lunak dengan mengirimkan data yang tidak valid, acak, atau tidak terduga sebagai input [25].

## 6. *Orthogonal array testing*

Teknik ini biasanya digunakan jika input memiliki ukuran kecil, namun menjadi lebih sulit diterapkan secara efisien jika digunakan dalam skala yang besar [26].

## 7. *State transition*

Teknik ini berguna untuk melakukan pengujian terhadap mesin dan navigasi antarmuka pengguna UI dalam bentuk grafik [27].

### **2.6.2 Manfaat Black Box Testing**

Terdapat beberapa manfaat dalam pengujian menggunakan Black Box Testing seperti:

1. Penguji dapat memahami pengguna dan melihat dari sudut pandang yang relevan saat melakukan pengujian perangkat lunak.
2. Black Box Testing dapat dengan cepat mengidentifikasi kekurangan sejak awal pengujian. Metode ini juga efektif dalam menguji sistem yang kompleks.
3. Black Box Testing memungkinkan penguji dan pengembang bekerja secara independen tanpa saling mengganggu dalam proses kerja [20].

## **2.7 Scrum**

Scrum adalah suatu kerangka kerja yang ringan dan membantu individu, tim, dan organisasi dalam menciptakan nilai melalui solusi adaptif terhadap masalah yang rumit. Dalam Scrum, diterapkan pendekatan bertahap untuk meningkatkan kemampuan analisa dan mengurangi kendala dalam pengembangan produk. Pendekatan ini memungkinkan tim untuk fokus pada tujuan yang jelas dan melakukan perubahan sesuai kebutuhan, sehingga mengoptimalkan kinerja secara efisien dan efektif. Dengan menggunakan Scrum, tim dapat merencanakan dan menjalankan proyek dengan lebih baik dan lebih terstruktur, sehingga menghasilkan produk yang lebih baik [6].

Setiap proses kontrol empiris didasarkan pada tiga pilar utama: transparansi, inspeksi, dan adaptasi. Ketiga pilar ini saling melengkapi dan memperkuat satu sama lain.



### 1. Transparansi

Transparansi adalah salah satu pilar utama dalam implementasi proses kontrol empiris. Hal ini berarti bahwa proses dan pekerjaan yang terjadi harus dapat dilihat dan dipahami oleh semua pihak yang terlibat, baik mereka yang melakukan pekerjaan maupun mereka yang menerima hasil pekerjaan.

### 2. Inspeksi

Inspeksi juga dapat membantu dalam mengidentifikasi masalah atau kekurangan dalam proses atau produk yang dapat segera ditangani dan diperbaiki. Penting bagi tim inspeksi untuk memiliki pemahaman yang mendalam tentang spesifikasi proyek dan standar kualitas yang harus dipenuhi. Mereka juga harus memiliki pengetahuan tentang proses produksi dan teknologi yang digunakan dalam proyek tersebut.

### 3. Adaptasi

Adaptasi juga diperlukan ketika ada perubahan dalam rencana proyek, spesifikasi, atau persyaratan pelanggan. Dalam hal ini, tim proyek harus mampu menyesuaikan diri dengan perubahan tersebut dan melakukan tindakan yang diperlukan untuk memenuhi persyaratan yang baru [6].

#### 2.7.1 Peran-Peran dalam Scrum

Terdapat tiga peran penting dalam Scrum yaitu sebagai berikut :

##### 1. *Product Owner*

Product owner merupakan orang yang bertanggung jawab atas pengelolaan kebutuhan dan kepentingan para pemangku kepentingan. Product Owner bertugas untuk merumuskan dan mengomunikasikan visi produk kepada tim pengembang, serta mengatur dan memprioritaskan backlog produk.

##### 2. *Scrum Master*

Scrum master adalah fasilitator dan pemimpin tim dalam menerapkan prinsip-prinsip dan praktik Scrum. Mereka bertanggung jawab untuk



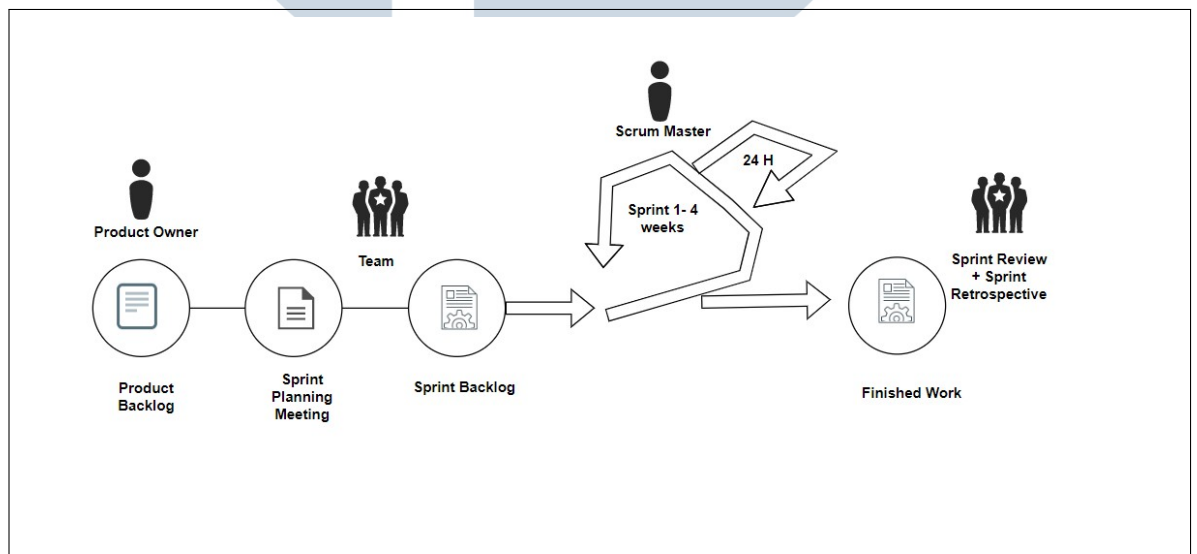
memastikan tim Scrum memahami dan menerapkan Scrum dengan benar, serta menghilangkan hambatan yang menghalangi kemajuan tim.

### 3. *Development Team*

Development team atau tim pengembang adalah kelompok orang yang bertanggung jawab langsung untuk mengembangkan dan menyampaikan produk yang dihasilkan. Mereka bekerja secara kolaboratif dalam mengerjakan tugas-tugas pengembangan, berkomunikasi secara terbuka, dan bertanggung jawab atas keberhasilan proyek secara keseluruhan [6].

## 2.7.2 Tahapan Scrum

Metode Scrum melibatkan serangkaian tahapan yang harus dilewati dalam proses pengembangan perangkat lunak, seperti:



Gambar 2.1. Tahapan Metode Scrum

### 1. *Product backlog*

Tahapan awal ini merupakan tanggung jawab dari Product Owner atau manajer. Secara sederhana, tahapan *product backlog* ini berfokus pada pembuatan daftar fitur-fitur yang akan dikembangkan dalam sistem dan memprioritaskan item-item dalam backlog berdasarkan kepentingan dan kebutuhan.

## 2. *Sprint planning*

Pada tahap perencanaan sprint, produk atau proyek teratas dalam product backlog akan diatur ulang menjadi sprint backlog. Tim Scrum juga bertanggung jawab untuk menentukan bagaimana mereka akan menyelesaikan tugas-tugas yang ada dalam sprint backlog.

## 3. *Daily Scrum*

Setelah cara dan batas waktu pengerjaan ditentukan dalam sprint planning, tahapan selanjutnya dalam metode Scrum adalah melaksanakan sprint. Proses ini melibatkan kegiatan daily scrum, di mana tim secara rutin bertemu untuk membahas kemajuan pekerjaan, mengidentifikasi hambatan, dan menyelaraskan upaya kolaboratif.

## 4. *Sprint Review*

Sprint review merupakan tahap terakhir dalam sprint. Pada tahap ini, produk atau proyek yang telah selesai dikerjakan akan dinilai dan dievaluasi. Produk tersebut akan diuji dan direview untuk memastikan bahwa telah memenuhi kriteria kualitas dan siap digunakan.

## 5. *Retrospective Process*

Scrum merupakan metode yang berulang. Proses-proses yang dilakukan oleh perusahaan, mulai dari pembuatan backlog hingga *review*, akan diingat dan dilakukan kembali dalam proses scrum proyek selanjutnya. Hal ini memastikan bahwa prinsip-prinsip dan praktik-praktik Scrum yang efektif dapat terus diterapkan dan disempurnakan dari waktu ke waktu [6].

### 2.7.3 Matriks Evaluasi

Matriks evaluasi berperan untuk mengetahui efektivitas penggunaan Scrum dalam sebuah pengembangan sistem. Berikut adalah beberapa contoh pengukuran dan metrik yang dapat digunakan dalam Scrum:

#### 1. Kecepatan Pengembangan

Ukuran yang menunjukkan seberapa banyak pekerjaan yang dapat diselesaikan oleh pengembang dalam setiap sprint. Kecepatan dihitung berdasarkan jumlah backlog item yang berhasil diselesaikan dalam satu

sprint. Ini memberikan gambaran tentang produktivitas pengembang dan dapat digunakan untuk merencanakan sprint berikutnya.

## 2. Tingkat Kepuasan Pelanggan

Ukuran yang menilai kepuasan pelanggan terhadap produk atau layanan yang dikembangkan oleh pengembang. Hal ini dapat diukur dengan menggunakan survei pelanggan, umpan balik dari pemangku kepentingan, atau melalui indikator kualitas produk seperti tingkat adopsi, peningkatan pendapatan, atau tingkat retensi pelanggan [28].

## 2.8 Penelitian Terdahulu

Penelitian terdahulu merupakan upaya peneliti untuk mencari perbandingan dan untuk menemukan inspirasi baru untuk penelitian. Berikut beberapa penelitian terdahulu yang digunakan dalam penelitian ini :

Penelitian pertama menghasilkan pengembangan sistem yang memperbaiki kegiatan operasional toko. Sistem ini memberikan keuntungan berupa informasi yang lebih cepat, akurat, dan optimal dalam penggunaan sistem komputerisasi. Penyelesaian masalah dalam pembuatan sistem ini dilakukan dengan memanfaatkan model UML, yang membantu dalam visualisasi, spesifikasi, pembangunan, dan dokumentasi sistem pengembangan perangkat lunak berbasis Object Oriented [29].

Penelitian kedua menghasilkan pengembangan sistem yang mempermudah proses pendataan barang masuk dan keluar. Sistem ini memiliki keunggulan dalam pencarian data yang diperlukan dan mengurangi penggunaan kertas yang berlebihan. Penyelesaian masalah dalam pembuatan sistem ini dilakukan melalui analisis kebutuhan sistem, yang mencakup analisis kebutuhan fungsional, analisis kebutuhan pengguna, analisis masukan dan keluaran, analisis sistem yang sedang berjalan, analisis sistem yang diusulkan, perancangan sistem, dan implementasi pada sistem [30].

Penelitian ketiga menghasilkan sistem yang mengelola data barang melalui website yang membuat pencarian data menjadi lebih cepat dan akurat. Penyelesaian masalah pada pembuatan sistem menggunakan analisis SWOT program. Analisa ini digunakan untuk mengukur kekuatan dan kelemahan dari sumber daya yang dimiliki, serta faktor eksternal juga internal di sebuah sistem tersebut [31].

Penelitian keempat menghasilkan sistem yang memiliki fungsi untuk mengeluarkan barang sesuai dengan urutan barang masuk sehingga barang tidak

disimpan lama dalam penyimpanan. Penyelesaian masalah pada pembuatan sistem diterapkan metode *FIFO* yang bertujuan untuk mengetahui urutan barang yang terlebih dahulu berada di dalam gudang [32].

Penelitian kelima menghasilkan sistem informasi yang dibuat memudahkan berbagai pihak untuk mengetahui informasi barang di dalam gudang. Penyelesaian masalah pada pembuatan sistem dengan cara analisa dan perancangan dengan UML (*Unified Modeling Language*) untuk menyesuaikan kebutuhan *stakeholder* [33].

Berdasarkan penelitian terdahulu, terdapat beberapa perbedaan yang akan dilakukan seperti berikut :

1. Memiliki fitur untuk memasukan foto barang.
2. Memiliki fitur untuk menghasilkan laporan dalam bentuk PDF dan Excel.

