

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Persediaan**

Persediaan adalah bahan atau barang yang dimiliki suatu perusahaan, yang disimpan dan akan digunakan untuk memenuhi tujuan tertentu, seperti untuk digunakan dalam proses produksi atau untuk dijual kembali [4]. Persediaan terdiri dari 3 jenis, yaitu persediaan bahan baku, persediaan bahan setengah jadi, dan persediaan barang jadi. Persediaan bahan baku dan bahan setengah jadi disimpan dalam suatu tempat yang telah ditentukan sebelum digunakan dalam proses produksi. Sedangkan persediaan barang jadi disimpan sebelum dijual atau dipasarkan [5]. Pelaksanaan proses produksi pada perusahaan yang menghasilkan barang jadi biasanya bergantung pada adanya persediaan. Oleh karena itu pengadaan persediaan perlu diperhatikan dengan melakukan penetapan jadwal dan jumlah pemesanan. Persediaan yang ada harus seimbang dengan kebutuhan karena akan berdampak pada biaya yang harus ditanggung perusahaan [6].

#### **2.2 Bahan Baku**

Pengertian bahan baku adalah suatu komponen yang digunakan untuk membuat barang jadi, dimana bahan baku tersebut pasti menempel menjadi satu dengan barang jadi [7]. Bahan-bahan baku diolah sedemikian rupa dalam proses produksi sehingga menghasilkan suatu produk jadi. Karena bahan baku merupakan bahan utama dalam menghasilkan produk, maka bahan baku menjadi salah satu faktor penentu hasil akhir produk, baik dari segi bentuk, kualitas, maupun harga jual.

#### **2.3 Sistem Informasi**

Sistem informasi dapat didefinisikan sebagai kegiatan mengumpulkan, memproses, menyimpan, menganalisis, menyebarkan informasi untuk suatu tujuan tertentu [8]. Dengan kata lain, sekumpulan subsistem melakukan fungsi pengolahan data, yaitu menerima masukan (*input*) berupa data, mengolahnya (*processing*), dan menghasilkan keluaran (*output*) berupa informasi sebagai dasar bagi pengambilan keputusan [9].

## 2.4 Sistem Informasi Persediaan

Sistem informasi persediaan atau sistem *inventory* didefinisikan sebagai sekumpulan kebijakan yang mengawasi dan mengatur tingkat persediaan, menentukan kapan dan berapa banyak stok bahan perlu dipesan [10]. Sistem informasi persediaan bahan baku mencatat proses penerimaan bahan baku dari *supplier*, ketersediaan bahan baku, pemakaian bahan baku, laporan aliran persediaan bahan baku, dan dokumentasi [11]. Sistem informasi persediaan dibangun untuk mempermudah pengontrolan stok bahan baku atau barang jadi yang disimpan oleh perusahaan.

## 2.5 PHP (*Hypertext Preprocessor*)

PHP adalah bahasa *server-side scripting* yang menyatu dengan HTML (*Hypertext Markup Language*) untuk membuat halaman suatu web yang dinamis [7]. PHP disisipkan di antara bahasa HTML, dieksekusi pada *web server*, lalu hasil yang dikirimkan ke *web browser* berupa HTML, dan kode PHP tidak akan terlihat [5]. PHP merupakan bahasa pemrograman *open source* dan sering digunakan dalam pembuatan sistem berbasis web.

## 2.6 CodeIgniter

*Framework* CodeIgniter adalah sebuah *framework* bahasa pemrograman PHP yang bersifat *open source*, *object oriented programming* (OOP), dan berkonsep MVC [12]. Konsep MVC memisahkan pengembangan sistem menjadi 3 jenis komponen, yaitu *view*, *model*, dan *controller*. Komponen pertama adalah *view*, yaitu bagian yang menangani tampilan, berupa file HTML, berfungsi untuk menerima dan merepresentasikan data kepada pengguna. Komponen kedua adalah *model*, yaitu bagian yang berhubungan langsung dengan *database* untuk memanipulasi data dan menangani validasi dari bagian *controller*. Komponen ketiga adalah *controller*, yaitu bagian yang mengatur hubungan *model* dan *view*, berfungsi untuk menerima permintaan data dari pengguna kemudian menentukan apa yang akan diproses oleh sistem [13].

## 2.7 MariaDB

MariaDB merupakan salah satu contoh *Database Management System* (DBMS) yang banyak digunakan. DBMS sendiri merupakan sistem yang digunakan

untuk pembuatan dan pengelolaan data dengan skala yang besar [14]. MariaDB merupakan salah satu *database server* pengembangan dari MySQL. Setelah MySQL diakuisisi oleh Oracle pada 2010, pengembang awal MySQL tidak dapat dengan leluasa menguasai MySQL sehingga dibangunlah MariaDB dengan performa yang bagus dan kompatibel dengan MySQL [14].


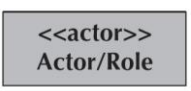
## 2.8 Unified Modeling Language

Hingga tahun 1995 konsep objek diimplementasikan dalam berbagai cara berbeda oleh para pengembang dengan metodologi dan notasi mereka sendiri. Pada tahun 1995, Rational Software menyatukan tiga pemimpin industri, yakni Grady Booch, Ivar Jacobson, dan James Rumbaugh, untuk menciptakan pendekatan tunggal pengembangan sistem berorientasi objek. Mereka bekerja dengan orang lain untuk membuat seperangkat teknik diagram yang dikenal sebagai *Unified Modeling Language* (UML). Tujuan UML adalah untuk menyediakan kumpulan kosakata umum dari istilah-istilah *object oriented* dan teknik diagram yang cukup kaya untuk memodelkan proyek pengembangan sistem apapun mulai dari analisis hingga implementasi. Pada November 1997, *Object Management Group* (OMG) secara resmi menerima UML sebagai standar semua pengembang objek [3]. Versi UML yang digunakan saat ini adalah UML 2.5.


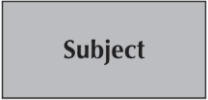

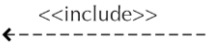
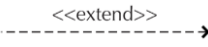

### 2.8.1 Use Case

*Use case* adalah cara formal yang menggambarkan bagaimana sistem bisnis berinteraksi dengan aktor dan proses bisnis. Pada dasarnya, *use case diagram* adalah tinjauan *high-level* dari proses bisnis dalam sistem informasi bisnis yang mewakili seluruh dasar untuk sistem berorientasi objek [3]. Dengan *use case diagram*, pengguna sistem dapat mengerti fungsi-fungsi sistem yang dimaksud tanpa harus mengetahui cara mengembangkan sistem.

Tabel 2.1 Elemen *Use Case Diagram*

Keterangan	Simbol
<p><b>Actor</b></p> <ul style="list-style-type: none"> <li>Seseorang atau sistem yang memperoleh manfaat dan bersifat eksternal terhadap subjek.</li> <li>Digambarkan sebagai <i>stick figure</i> atau persegi panjang (jika aktor bukan manusia) dengan tulisan &lt;&lt;actor&gt;&gt; di dalamnya.</li> <li>Diberi label dengan perannya.</li> <li>Dapat diasosiasikan dengan aktor lain menggunakan asosiasi spesialisasi/<i>superclass</i>, dilambangkan dengan panah khusus.</li> <li>Ditempatkan di luar batas subjek.</li> </ul>	 <p>Actor/Role</p> 

Tabel 2.1 Elemen *Use Case Diagram* (lanjutan)




Keterangan	Simbol
<b>Use Case</b> <ul style="list-style-type: none"> <li>• Merepresentasikan bagian utama dari fungsi sistem.</li> <li>• Dapat <i>extend</i> ke <i>use case</i> lain.</li> <li>• Dapat <i>include use case</i> lain.</li> <li>• Ditempatkan di dalam batas sistem.</li> <li>• Diberi label dengan frasa kata kerja – kata benda deskriptif.</li> </ul>	
<b>Subject Boundary</b> <ul style="list-style-type: none"> <li>• Nama subjek ditulis di dalam atau di atas simbol.</li> <li>• Merepresentasikan lingkup subjek, seperti sistem atau proses bisnis.</li> </ul>	
<b>Association Relationship</b> <ul style="list-style-type: none"> <li>• Menghubungkan <i>actor</i> dengan <i>use case</i> yang berinteraksi dengannya.</li> </ul>	
<b>Include Relationship</b> <ul style="list-style-type: none"> <li>• Merepresentasikan penyertaan fungsi dari suatu <i>use case</i> di dalam <i>use case</i> yang lain.</li> <li>• Panah ditarik dari <i>use case</i> asal ke <i>use case</i> yang disertakan.</li> </ul>	
<b>Extend Relationship</b> <ul style="list-style-type: none"> <li>• Merepresentasikan ekstensi dari suatu <i>use case</i> untuk memasukkan perilaku opsional.</li> <li>• Panah ditarik dari <i>use case</i> ekstensi ke <i>use case</i> dasar.</li> </ul>	
<b>Generalization Relationship</b> <ul style="list-style-type: none"> <li>• Merepresentasikan <i>use case</i> khusus ke yang lebih umum.</li> <li>• Panah ditarik dari <i>use case</i> khusus ke <i>use case</i> dasar.</li> </ul>	

Sumber : Dennis et al, 2015, p. 122.






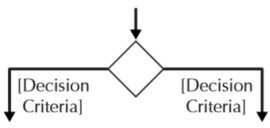
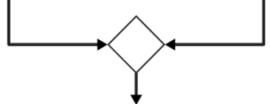



## 2.8.2 Activity Diagram

*Activity diagram* adalah suatu model yang biasanya digunakan untuk memodelkan segala jenis proses, mulai dari alur kerja bisnis *high-level* yang melibatkan banyak *use case* berbeda, hingga *use case* individual yang spesifik, serta untuk memodelkan perilaku dalam proses bisnis yang tidak bergantung pada objek [3].

Tabel 2.2 Elemen *Activity Diagram*

Keterangan	Simbol
<b>Action</b> <ul style="list-style-type: none"> <li>• Perilaku yang sederhana dan tidak dapat diuraikan.</li> <li>• Diberi label dengan namanya.</li> </ul>	
<b>Activity</b> <ul style="list-style-type: none"> <li>• Merepresentasikan serangkaian <i>action</i>.</li> <li>• Diberi label dengan namanya.</li> </ul>	
<b>Object Node</b> <ul style="list-style-type: none"> <li>• Merepresentasikan objek yang terhubung dengan serangkaian <i>object flow</i>.</li> <li>• Diberi label dengan nama kelasnya.</li> </ul>	

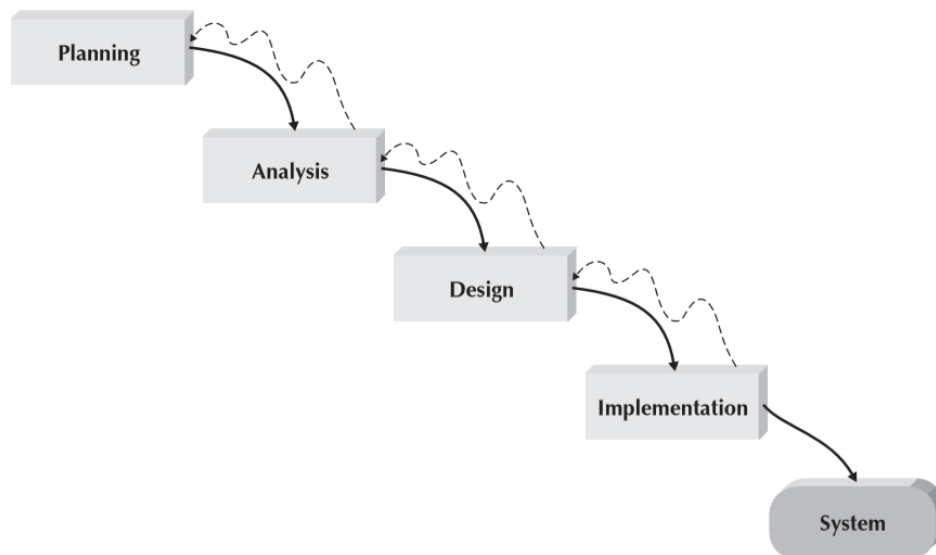
Tabel 2.2 Elemen *Activity Diagram* (lanjutan)

Keterangan	Simbol
<b>Control Flow</b> <ul style="list-style-type: none"> <li>Memperlihatkan urutan eksekusi.</li> </ul>	
<b>Object Flow</b> <ul style="list-style-type: none"> <li>Memperlihatkan aliran dari suatu objek dari satu <i>activity/action</i> ke <i>activity/action</i> lain.</li> </ul>	
<b>Initial Node</b> <ul style="list-style-type: none"> <li>Menggambarkan awal dari serangkaian <i>action</i> atau <i>activity</i>.</li> </ul>	
<b>Final-Activity Node</b> <ul style="list-style-type: none"> <li>Digunakan untuk menghentikan semua <i>control flow</i> dan <i>object flow</i> dalam suatu <i>activity</i> (atau <i>action</i>).</li> </ul>	
<b>Final-Flow Node</b> <ul style="list-style-type: none"> <li>Digunakan untuk menghentikan <i>control flow</i> atau <i>object flow</i> tertentu.</li> </ul>	
<b>Decision Node</b> <ul style="list-style-type: none"> <li>Merepresentasikan kondisi pengujian untuk memastikan bahwa <i>control flow</i> atau <i>object flow</i> hanya berada di satu jalur.</li> <li>Diberi label dengan kriteria keputusan untuk melanjutkan ke jalur tertentu.</li> </ul>	
<b>Merge Node</b> <ul style="list-style-type: none"> <li>Digunakan untuk menyatukan kembali jalur <i>decision</i> yang berbeda yang dibuat menggunakan <i>decision node</i>.</li> </ul>	
<b>Fork Node</b> <ul style="list-style-type: none"> <li>Digunakan untuk membagi perilaku menjadi serangkaian <i>activity</i> (atau <i>action</i>) yang paralel.</li> </ul>	
<b>Join Node</b> <ul style="list-style-type: none"> <li>Digunakan untuk menyatukan kembali serangkaian <i>activity</i> (atau <i>action</i>) yang paralel.</li> </ul>	
<b>Swimlane</b> <ul style="list-style-type: none"> <li>Digunakan untuk memecah <i>activity diagram</i> menjadi baris dan kolom untuk menetapkan <i>activity/action</i> individual untuk individu atau objek yang bertanggung jawab menjalankan <i>activity</i> (atau <i>action</i>).</li> <li>Diberi label dengan nama individu atau objek yang bertanggung jawab.</li> </ul>	

Sumber : Dennis et al, 2015, p. 132.

## 2.9 System Development Life Cycle (SDLC)

*System Development Life Cycle* (SDLC) adalah proses memahami bagaimana sistem informasi dapat mendukung kebutuhan bisnis dengan merancang suatu sistem, membangunnya, dan mengirimkannya kepada pengguna [3]. Terdapat empat fase dalam *System Development Life Cycle*, yaitu fase perencanaan, fase analisis, fase desain, dan fase implementasi.



Gambar 2.1 Tahap-Tahap *System Development Life Cycle (SDLC)*  
Sumber : Dennis et al, 2015, p. 7.

### 2.9.1 Fase Perancangan

Fase perencanaan atau *planning* adalah fase untuk memahami mengapa suatu sistem harus dibangun dan menentukan bagaimana membangunnya. Fase ini terdiri dari dua langkah. Pertama adalah inisiasi proyek, tahap untuk mengidentifikasi nilai bisnis dari sistem dan mencatat ide atau permintaan untuk sistem, kemudian permintaan tersebut disetujui. Tahap kedua adalah manajemen proyek, tahap merencanakan proyek atau pengembangan sistem [3].

### 2.9.2 Fase Analisis

Fase analisis adalah fase untuk menyelidiki segala hal terkait sistem yang akan dikembangkan, seperti siapa yang akan menggunakan sistem, apa yang bisa dilakukan sistem, dan kapan digunakan. Terdapat tiga tahap dalam fase ini, yaitu strategi analisis, *requirements gathering*, dan proposal sistem yang menggambarkan *business requirements* yang harus dipenuhi oleh sistem [3].

### 2.9.3 Fase Desain

Fase desain adalah fase menentukan bagaimana sistem akan beroperasi, misalnya membuat desain antarmuka, *form*, laporan, fungsi-fungsi yang akan digunakan, dan *database*. Hasil desain menjadi spesifikasi sistem yang akan diimplementasikan [3].

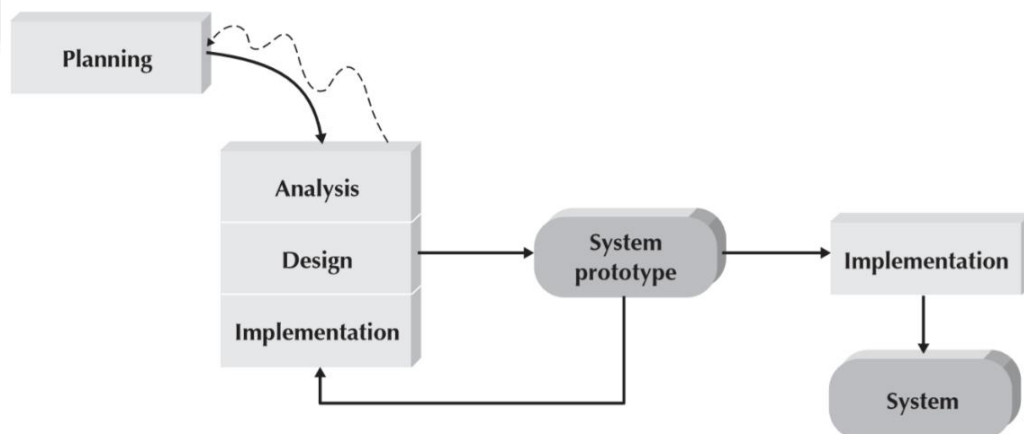


#### 2.9.4 Fase Implementasi

Fase implementasi adalah fase dimana sistem dibangun dan dilakukan pengujian. Setelah sistem selesai dibangun, dilakukan instalasi sistem dan para pengguna diajarkan cara menggunakan sistem tersebut. Kemudian dibuat rencana dukungan untuk sistem [3].

#### 2.10 Prototyping

*Prototype* merupakan satu versi dari sebuah sistem yang dikembangkan yang dapat memberikan ide atau gambaran bagi para pengembang dan calon pengguna mengenai bagaimana sistem tersebut akan berfungsi [11]. Metodologi *prototyping* melakukan tiga fase secara bersamaan, yaitu fase analisis, desain, dan implementasi, dan ketiganya dilakukan berulang kali dalam satu siklus hingga suatu sistem selesai dibangun. Prototipe pertama biasanya ditujukan kepada pengguna dan sponsor proyek untuk memberikan komentar atau *feedback*. Komentar-komentar ini digunakan untuk melakukan *reanalyze*, *redesign*, dan *reimplementation* pada prototipe selanjutnya yang menyediakan beberapa fitur tambahan. Proses berlanjut dalam satu siklus hingga *stakeholder* menerima dan setuju bahwa prototipe tersebut sudah menyediakan fungsi yang layak untuk digunakan dalam organisasi [3].



Gambar 2.2 Metodologi *Prototyping*  
Sumber : Dennis et al, 2015, p. 10.

## 2.11 Penelitian Terdahulu

Tabel 2.3 Penelitian Terdahulu

No.	Judul	Penulis	Tahun	Publikasi	Deskripsi
1	Perancangan Sistem Informasi Persediaan Barang Pada CV. Bintang Lima Furniture	Fery Wongso	2017	Jurnal Ilmu Komputer dan Bisnis, Vol. 8, No. 2, Hal. 1977-1987	Penelitian ini membahas perancangan sistem persediaan barang yang memiliki pencatatan stok barang yang akurat sehingga pengecekan dapat dilakukan secara otomatis oleh sistem dan informasi yang didapat mempermudah pembelian barang kepada supplier.
2	Perancangan Sistem Informasi Persediaan Bahan Baku Di UD. XY, Tulungagung	Retno Astuti, Reza Ashari, dan Mas'ud Effendi	2018	Jurnal Teknologi Industri Pertanian, Vol. 29, No. 2, Hal. 162-174	Penelitian ini membahas perancangan sistem informasi persediaan bahan baku yang dapat mengelola data <i>input</i> dan dijadikan <i>output</i> laporan persediaan. Sistem menghasilkan informasi daftar bahan baku, pasokan bahan baku, penggunaan serta ketersediaan bahan baku di dalam gudang.
3	Sistem Pengendalian Persediaan Bahan Baku, <i>Inventory</i> dan Produksi pada <i>Home Industry</i> Mamake dengan Metode <i>Reorder Point</i> berbasis Web	Wirantika Rahma Putri, Irma Permata Sari	2018	Jurnal Multinetics, Vol. 4, No. 2, Hal. 22-27	Penelitian ini membahas hasil perancangan dan implementasi sistem persediaan bahan baku yang bisa menampilkan stok bahan baku yang tersedia dan mengelola data pembelian bahan baku ke <i>supplier</i> .
4	Perancangan Sistem Informasi Berbasis Web (Studi Kasus PT. Continental Panjipratama)	Budi Utami Fahnun, Handoko Dwi Hartono, Yuli Karyanti	2014	Jurnal Ilmiah FIFO, Vol. 6, No. 1, Hal. 8-14	Penelitian ini membahas perancangan sistem berbasis web yang dapat membantu proses pembelian, penjualan, dan pengecekan persediaan. Data-data disimpan di dalam sistem dan terus dibaharui. Melalui sistem yang dirancang, staf <i>marketing</i> , <i>purchasing</i> , dan produksi bisa saling berinteraksi sehingga stok bahan baku akan terus dibaharui dan memperlancar prosedur penjualan barang pada perusahaan.



Tabel 2.3 Penelitian Terdahulu (lanjutan)

No.	Judul	Penulis	Tahun	Publikasi	Deskripsi
5	Sistem Informasi Impor Bahan Baku Berbasis Web Sebagai Upaya Minimalisasi Biaya Persediaan	Indria Widyastuti, Arfan Hamdi	2017	<i>Indonesian Journal on Computer and Information Technology</i> , Vol. 2, No. 2, Hal. 55-64	Penelitian ini membahas pengembangan sistem informasi yang dapat mengontrol persediaan bahan baku dan membantu proses pembelian bahan baku. Dengan adanya sistem ini, jumlah bahan baku yang disimpan di gudang tidak akan berlebih namun akan selalu tersedia ketika proses produksi dilaksanakan. Pembelian bahan baku yang terkontrol membuat biaya yang dikeluarkan dapat diminimalisasi dan jadwal produksi dapat terlaksana sesuai perencanaan.