

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Penjualan**

Penjualan merupakan kegiatan menjual barang dagangan yang melibatkan dua atau lebih individu. Barang dagangan yang dimaksud dapat berupa barang secara fisik ataupun jasa. Penjualan dilakukan untuk mencapai suatu tujuan tertentu yang diinginkan. Penjualan juga merupakan suatu transaksi yang memiliki tujuan untuk mendapatkan suatu keuntungan, dan merupakan suatu jantung dari suatu perusahaan (Himayati, 2008:123). Pengertian lain dari penjualan adalah pemindahan hak milik atas barang atau pemberian jasa dari penjual kepada pembeli. Pemindahan hak milik barang atau pemberian jasa ini dilakukan dengan sejumlah harga yang telah disepakati antara pihak pembeli dan penjual (Rangkuti, 2009:206) [10].

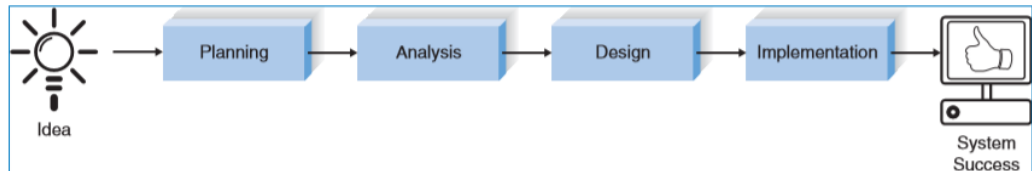
#### **2.2 Sistem Informasi Penjualan**

Sistem Informasi Penjualan merupakan suatu sistem yang mengatur pembuatan pernyataan penjualan. Kegiatan yang ada pada sistem informasi penjualan akan dijelaskan melalui langkah-langkah yang meliputi urutan kegiatan sejak penjual menerima pesanan dari pembeli, pengecekan stok barang dan dilanjutkan dengan pengiriman barang. Tidak lupa juga langkah-langkah tersebut pun juga disertai dengan pembuatan faktura tau nota serta pencatatan atas penjualan barang. Pencatatan atas penjualan barang biasanya dicatat setiap hari dengan tujuan agar penjual mengetahui keuntungannya per hari [3].

#### **2.3 System Development Life Cycle (SDLC)**

*System Development Life Cycle (SDLC)* merupakan proses mendesain, merancang, membangun, dan mengimplementasikan sebuah sistem. Tahapan dari

*SDLC* ada empat tahapan. Dimulai dari tahap perancangan, analisis, desain, sampai dengan implementasi. Dalam penerapannya, tahapan pada *SDLC* tidak selalu berurutan, namun tahapan *SDLC* dalam sebuah *project* selalu memiliki keempat tahapan tersebut.



Gambar 2. 1 *The System Development Life Cycle*

Sumber: A. Dennis, B. H. Wixom, and D. Tegarden, *Systems analysis and design: An objectoriented approach with UML*, 5th ed. USA: Wiley, 2015.

Tahap pertama adalah tahap perencanaan. Pada tahap perencanaan, penting sekali untuk memahami mengapa sebuah sistem informasi harus dikembangkan. Dalam tahap ini juga, perancang sistem harus menentukan cara tim untuk membangunnya. Tahap perencanaan terdiri dari dua proses. Proses pertama adalah identifikasi cara mengatur *cashflow*, menciptakan sistem yang dapat memiliki *business value*. Proses kedua dilakukan saat *project* sudah diterima, lalu dilanjutkan ke *project management*. Hasil dari proses tersebut yaitu *project plan* dimana *project plan* berfungsi untuk mendeskripsikan bagaimana tim akan mengembangkan sistem.

Tahap kedua adalah tahap analisis. Pada tahap analisis, perancang sistem harus memahami siapa yang nantinya akan menggunakan sistem tersebut. Hal lain yang harus diperhatikan adalah kemampuan sistem serta tempat dan waktu sistem tersebut akan digunakan. Dalam tahap ini, tim akan melakukan investigasi sistem saat ini, identifikasi peluang untuk meningkatkan kemampuan sistem, serta akan dilakukan pengembangan fitur untuk sistem yang baru.

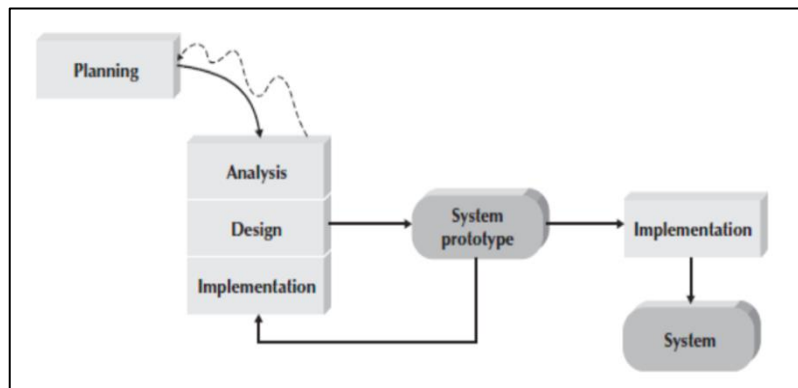
Tahap ketiga adalah tahap desain. Pada tahap desain dilakukan penentuan cara sistem beroperasi nantinya. Mulai dari *software*, *hardware*, *network*

*infrastructure*, tampilan sistem dan segala hal yang berkaitan dengan sistem. Pada tahap ini, sistem akan diuji agar tidak terjadi kesalahan sistem.

Tahap terakhir adalah tahap implementasi. Tahap implementasi terdiri dari tiga proses. Proses pertama adalah *system construction*. Sistem akan dirancang dan diuji kelengkapan dan kesesuaiannya dengan rencana awal agar sistem dapat berjalan sesuai dengan desain yang telah ditentukan. Proses kedua adalah sistem melakukan instalisasi. Proses instalisasi merupakan proses dimana sistem yang lama akan dimatikan dan diganti dengan yang sistem baru. Proses ketiga adalah proses dimana tim analisis membuat perencanaan untuk mendukung kerja sistem [4].

#### **2.4 Prototyping**

Tujuan dari dilakukannya sebuah *Prototyping* adalah untuk mengumpulkan berbagai informasi dari calon pengguna sehingga sistem yang dibuat tidak melenceng dari tujuan pembuatan sistem. *Prototype* memberikan gambaran awal dari sistem yang akan dibuat nantinya. *Prototyping* dapat diimplementasikan pada pengembangan sistem yang kecil maupun besar. *Prototyping* dibuat dengan harapan agar proses pengembangan sistem dapat berjalan dengan baik, sistematis serta dapat diselesaikan dalam jangka waktu yang telah ditentukan. Peran dari pengguna sangat berpengaruh saat proses pembuatan *prototype*. Jika pengguna berperan aktif, maka akan menguntungkan seluruh pihak yang terlibat dalam sistem. Baik itu pimpinan perusahaan, pengguna sendiri itu sendiri maupun pengembang sistem [5]. Gambar 2.1 berikut ini adalah gambar dari tahapan *prototyping*.



Gambar 2. 2 Metodologi *Prototyping*

Sumber: A. Dennis, B. H. Wixom, and D. Tegarden, Systems analysis and design: An objectoriented approach with UML, 5th ed. USA: Wiley, 2015.

Tahap-tahap pengembangan model *Prototype* menurut Rogeer S. Pressman dibagi menjadi 4 tahapan. Tahap pertama adalah *Communication*. Pada tahap *communication*, pengembang sistem dan pengguna berdiskusi untuk menentukan tujuan dari sistem yang akan dibuat. Tahap kedua adalah *Quick Plan, Modelling and Quick Design*.

Tahap kedua ini akan dilakukan setelah gambaran sistem sudah diketahui secara umum. Proses *Quick Design* berisi proses perancangan *user interface* dari sistem. Tahap ketiga yaitu *Construction of Prototype*. Tahap ketiga ini dilakukan setelah diketahui tujuan umum dan rancangan dari sistem. Maka, setelah itu *prototype* akan mulai dikerjakan.

Tahap terakhir adalah *Deployment Delivery and Feedback*. Tahap ini dilakukan setelah *prototype* sudah terbentuk. Selanjutnya, *prototype* akan diserahkan kepada pengguna untuk di evaluasi. Pada tahap ini pengguna dapat memberikan *feedback* mengenai kelayakan dari apa yang sudah dibuat. Jika terjadi hal yang tidak sesuai, maka pengembang sistem akan memperbaiki *prototype* tersebut. Seiring dengan dilakukannya evaluasi *prototype* sistem oleh pengguna, tahap *Communication* beserta tahapan-tahapan lainnya kembali diulang sampai dengan kepuasan pengguna terhadap sistem tercapai [6].



## 2.5 Unified Modelling Language (UML)

*Unified Modelling Language (UML)* merupakan bahasa standar yang digunakan untuk mendokumentasikan dan menjelaskan proses dalam membangun *software*. *UML* merupakan metodologi dalam mengembangkan sistem *software* berbasis *Object Oriented* [7]. *UML* memberikan standar penulisan yang meliputi konsep dalam proses bisnis, skema pada *database*, dan komponen-komponen yang diperlukan dalam membuat sebuah sistem. Dalam merancang *UML* dikategorikan dalam 3 jenis yaitu *use case diagram*, *activity diagram*, dan *class diagram* [4].




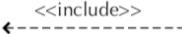


### 2.5.1 Use Case Diagram

*Use Case Diagram* merupakan skema sebagian atau seluruh *actor* dan *use case* dengan tujuan mengetahui interaksi yang ada dalam sistem tersebut. *Use case diagram* menjelaskan dan memberikan gambaran fungsionalitas dari sebuah sistem. *Use case* menggambarkan hal yang akan dibuat dalam sistem. Elemen-elemen dan penjelasan yang ada pada *user case diagram* terdapat dalam Tabel 2.1 [4].

Tabel 2. 1 Elemen-elemen *Use Case Diagram*

No.	Keterangan	Simbol
1.	<p><i>Actor</i></p> <ul style="list-style-type: none"><li>• Merupakan seseorang atau sistem yang memperoleh manfaat dari dan memiliki posisi di luar subjek.</li><li>• Digambarkan sebagai <i>stick figure</i> atau persegi panjang dengan &lt;&lt;actor&gt;&gt; didalamnya.</li><li>• <i>Label</i> diberikan sesuai dengan perannya.</li><li>• Dapat dikaitkan dengan <i>actor</i> lain menggunakan asosiasi khusus atau <i>superclass</i> dan dilambangkan dengan panah yang berongga.</li><li>• Ditempatkan di luar batas subjek.</li></ul>	 Actor/Role 

Tabel 2. 2 Elemen-elemen *Use Case Diagram* (Lanjutan)

No.	Keterangan	Simbol
2.	<p><i>Use Case</i></p> <ul style="list-style-type: none"> <li>• Merepresentasikan fungsionalitas sistem.</li> <li>• Dapat <i>extend</i> dengan <i>use case</i> lainnya.</li> <li>• Dapat <i>include</i> dengan <i>use case</i> lainnya.</li> <li>• <i>Label</i> diberikan dengan menggunakan <i>descriptive verb-noun phrase</i>.</li> <li>• Terletak di dalam <i>subject boundary</i>.</li> </ul>	
3.	<p><i>Subject Boundary</i></p> <ul style="list-style-type: none"> <li>• Nama subjek terletak didalam atau diatas.</li> <li>• Merepresentasikan <i>scope</i> subjek</li> </ul>	
4.	<p><i>Association Relationship</i></p> <ul style="list-style-type: none"> <li>• Menghubungkan interaksi antara <i>actor</i> dan <i>use case</i>.</li> </ul>	
5.	<p><i>Include Relationship</i></p> <ul style="list-style-type: none"> <li>• Merupakan penyertaan fungsi dari satu <i>use case</i> dengan <i>use case</i> yang lainnya.</li> <li>• Memiliki tanda panah yang memiliki arah dari <i>use case</i> ekstensi ke <i>use case</i> dasar</li> </ul>	
6.	<p><i>Extend Relationship</i></p> <ul style="list-style-type: none"> <li>• Berfungsi untuk memasukkan perilaku <i>optional</i>.</li> <li>• Panah berarah dari <i>use case</i> khusus ke <i>use case</i> dasar.</li> </ul>	
7.	<p><i>Generalization Relationship</i></p> <ul style="list-style-type: none"> <li>• Mewakikan <i>use case</i> khusus ke <i>use case</i> umum.</li> <li>• Tanda panah mengarah ke <i>use case</i> dasar.</li> </ul>	

Sumber: A. Dennis, B. H. Wixom, and D. Tegarden, Systems analysis and design: An objectoriented approach with UML, 5th ed. USA: Wiley, 2015.








## 2.5.2 Activity Diagram

*Activity Diagram* merepresentasikan rangkaian alur aktivitas atau proses bisnis dalam sistem yang dirancang. *Activity Diagram* juga memiliki fungsi untuk


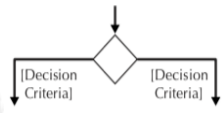
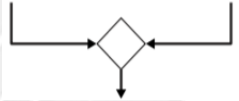



mendeskripsikan aktivitas yang dibentuk dalam suatu proyek atau operasi. *Activity Diagram* berupa *flow chart* yang berfungsi untuk menunjukkan alur dari sebuah sistem. Model ini memberikan informasi seputar hal-hal yang dibutuhkan untuk dapat membangun dan merancang sebuah sistem . Elemen-elemen dari *Activity Diagram* beserta penjelasan terdapat pada Tabel 2.2 [4].

Tabel 2. 3 Elemen-elemen *Activity Diagram*

No.	Keterangan	Simbol
1.	<p><i>Action</i></p> <ul style="list-style-type: none"> <li>Merupakan bagian dari perilaku sederhana yang tidak bisa didekomposisi.</li> <li><i>Label</i> diberikan sesuai dengan namanya.</li> </ul>	
2.	<p><i>Activity</i></p> <ul style="list-style-type: none"> <li>Berfungsi untuk merepresentasikan rangkaian aktivitas.</li> <li><i>Label</i> diberikan sesuai dengan namanya.</li> </ul>	
3.	<p><i>Object node</i></p> <ul style="list-style-type: none"> <li>Berfungsi untuk merepresentasikan sebuah objek yang berkaitan dengan <i>object flow</i>.</li> </ul>	
4.	<p><i>Control flow</i></p> <ul style="list-style-type: none"> <li>Menunjukkan urutan eksekusi.</li> </ul>	
5.	<p><i>Object flow</i></p> <ul style="list-style-type: none"> <li>Memberikan petunjuk alur objek dari suatu <i>activity</i> atau <i>action</i> ke <i>activity</i> atau <i>action</i> yang lainnya.</li> </ul>	
6.	<p><i>Initial node</i></p> <ul style="list-style-type: none"> <li>Menggambarkan <i>activity</i> atau <i>action</i> dimulai.</li> </ul>	
7.	<p><i>Final activity node</i></p> <ul style="list-style-type: none"> <li>Berfungsi untuk menghentikan semua <i>control flow</i> dan <i>object flow</i> dalam sebuah <i>activity</i> atau <i>action</i>.</li> </ul>	

Tabel 2.2 Elemen-elemen *Activity Diagram* (Lanjutan)

No.	Keterangan	Simbol
8.	<p><i>Final flow node</i></p> <ul style="list-style-type: none"> <li>Berfungsi untuk menghentikan <i>control flow</i> atau <i>object flow</i> tertentu.</li> </ul>	
9.	<p><i>Decision node</i></p> <ul style="list-style-type: none"> <li>Berfungsi untuk merepresentasikan suatu kondisi untuk memastikan bahwa <i>control flow</i> dan <i>object flow</i> hanya berada pada satu jalur.</li> <li><i>Label</i> diberikan dengan kriteria <i>decision</i> untuk melanjutkan ke jalur tertentu.</li> </ul>	
10.	<p><i>Merge node</i></p> <ul style="list-style-type: none"> <li>Berfungsi untuk menyatukan kembali jalur <i>decision</i> berbeda yang telah dibuat menggunakan <i>decision node</i>.</li> </ul>	
11.	<p><i>Swimlane</i></p> <ul style="list-style-type: none"> <li>Berfungsi untuk memecah <i>activity diagram</i> menjadi barisan dan kolom untuk menempatkan <i>action</i> atau <i>activity</i> individual dengan individu atau objek yang mengeksekusinya.</li> <li><i>Label</i> diberikan dengan nama individu atau objek yang memiliki tanggung jawab di bagian tersebut.</li> </ul>	

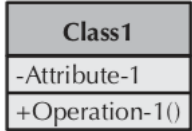
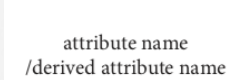
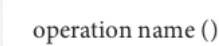
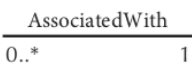
Sumber: A. Dennis, B. H. Wixom, and D. Tegarden, Systems analysis and design: An objectoriented approach with UML, 5th ed. USA: Wiley, 2015.

### 2.5.3 Class Diagram




*Class Diagram* merupakan model statis yang menggambarkan struktur dari suatu sistem dengan cara menunjukkan sistem *class*, *attribute*, *operation* dan hubungan antar objek. *Class Diagram* berfungsi untuk memvisualisasikan struktur *class* dari suatu sistem. Elemen-elemen dari *Class Diagram* beserta penjelasan terdapat pada Tabel 2.3 [4].



Tabel 2. 4 Elemen-elemen *Class Diagram*

No.	Keterangan	Simbol
1.	<p><i>Class</i></p> <ul style="list-style-type: none"> <li>• Merepresentasikan seseorang, tempat atau hal yang perlu disimpan sistem.</li> <li>• Memiliki nama yang diketik tebal (<i>bold</i>) dan terletak di bagian tengah atas.</li> <li>• Memiliki daftar <i>attribute</i> di kompartemen tengah.</li> <li>• Memiliki daftar <i>operation</i> di kompartemen bawah.</li> <li>• Tidak menunjukkan operasi yang tersedia pada semua <i>class</i> secara jelas.</li> </ul>	
2.	<p><i>Attribute</i></p> <ul style="list-style-type: none"> <li>• Merepresentasikan <i>property</i> yang menjelaskan keadaan suatu objek.</li> <li>• Dapat diturunkan dari <i>attribute</i> lain dengan cara menempatkannya pada garis miring sebelum nama <i>attribute</i>.</li> </ul>	
3.	<p><i>Operation</i></p> <ul style="list-style-type: none"> <li>• Merepresentasikan tindakan atau fungsi yang dapat dilakukan oleh suatu <i>class</i>.</li> <li>• Dapat dikategorikan sebagai <i>constructor</i>, <i>query</i>, <i>update operation</i>.</li> <li>• Terdapat tanda kurung yang berisi parameter atau informasi yang diperlukan untuk melakukan <i>operation</i>.</li> </ul>	
4.	<p><i>Association</i></p> <ul style="list-style-type: none"> <li>• Merepresentasikan hubungan antar beberapa <i>class</i> atau <i>class</i> dengan <i>class</i> itu sendiri.</li> <li>• <i>Label</i> diberikan dengan menggunakan frasa kata kerja atau nama pemeran.</li> <li>• Dapat terletak di antara satu <i>class</i> atau lebih.</li> </ul>	

Tabel 2. 5 Elemen-elemen *Class Diagram* (Lanjutan)

No.	Keterangan	Simbol
5.	<i>Generalization</i> <ul style="list-style-type: none"> <li>Mewakili sebuah hubungan antar beberapa <i>class</i>.</li> </ul>	
6.	<i>Aggregation</i> <ul style="list-style-type: none"> <li>Merepresentasikan bagian logis dari hubungan antara beberapa <i>class</i> atau <i>class</i> dengan dirinya sendiri.</li> <li>Merupakan bentuk khusus dari <i>association</i>.</li> </ul>	
7.	<i>Composition</i> <ul style="list-style-type: none"> <li>Merepresentasikan bagian fisik dari hubungan antara beberapa <i>class</i> atau <i>class</i> dengan dirinya sendiri.</li> <li>Merupakan bentuk khusus dari <i>association</i>.</li> </ul>	

Sumber: A. Dennis, B. H. Wixom, and D. Tegarden, *Systems analysis and design: An object oriented approach with UML*, 5th ed. USA: Wiley, 2015.

## 2.6 Business Process Model and Notation (BPMN)

Menurut *Object Management Group*, BPMN merupakan standar untuk melakukan pemodelan proses bisnis yang menyediakan notasi untuk memudahkan penjelasan proses bisnis yang terjadi dalam *Business Process Diagram* (BPD). BPMN juga menyediakan cara berkomunikasi untuk proses bisnis yang terjadi antara personil manajemen, analisis bisnis, dan pengembang sistem sehingga memudahkan mereka mendefinisikan dan menganalisis proses bisnis yang bersifat umum dan pribadi. Ada 3 jenis elemen utama dari BPMN, yaitu; *flow objects*, *connecting objects*, *swim lanes* (*Object Management Group* (OMG), 2016). BPMN dirancang dengan tujuan membuat pemodelan pada proses bisnis dan menciptakan bisnis proses *end-to-end*. Pada BPMN terdapat 3 sub-model (Von Rosing, Von Scheel, & Scheer, 2014), yaitu:

### a. Proses termasuk di dalamnya;

- Proses Bisnis *Private non-executable* yang merupakan model untuk organisasi tertentu. Tujuan proses bisnis ini dimodelkan adalah untuk dieksekusi menurut eksekusi semantik BPMN.









- Proses Bisnis *Private executable* yang merupakan proses pribadi yang dimodelkan untuk mendokumentasikan perilaku proses pada tingkat modeler.
- Proses Bisnis Publik (*public process*) yang merupakan model untuk menjelaskan interaksi antar proses atau *participant*. Proses publik dapat dimodelkan secara terpisah atau digabung untuk menunjukkan *flow* dari sebuah pesan.

b. Koreografi yang merupakan proses interaksi yang mewakili satu atau lebih set pertukaran pesan. Pesan yang dimaksud melibatkan dua atau lebih *participants*.







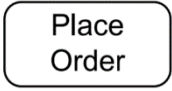
c. Kolaborasi yang merupakan model untuk menunjukkan interaksi antara dua atau lebih entitas bisnis. Dalam kolaborasi biasanya terdiri dari dua *pools* atau lebih yang mewakili *participant*. Pesan yang ditukar antar *participant* ditunjukkan dengan *flow* pesan yang menghubungkan antara dua *pools* atau objek yang ada di dalamnya. Diantara *pools* dapat menampilkan elemen koreografi karena hal ini membagi antara dua *flow* pesan antar *pools*. Kolaborasi dapat berisi kombinasi *pools*, proses, dan koreografi.

BPMN tidak digunakan sebagai model data, akan tetapi BPMN digunakan untuk menjelaskan *flow* daripada data atau pesan yang terjadi dalam sebuah kegiatan. Peraturan bisnis dalam perusahaan, pengambilan keputusan, laporan bisnis tidak termasuk dalam BPMN. Secara singkat, BPMN ini sebuah metode yang digunakan untuk memberi gambaran mengenai proses bisnis dalam bentuk diagram yang menyerupai *flowchart*. BPMN dibuat dengan tujuan agar orang dapat lebih mudah memahami semua bagian yang ada di dalam proses bisnis. Penjelasan aktivitas yang ada pada BPMN dijelaskan dengan menggunakan kata kerja [8]. BPMN sendiri memiliki beberapa elemen-elemen yang penting. Berikut adalah elemen-elemen yang paling sering digunakan dalam BPMN dan penjelasannya yang dapat dilihat pada Tabel 2.4

Tabel 2. 6 Elemen-elemen BPMN

No.	Keterangan	Simbol
1.	<p><i>Start Event</i></p> <ul style="list-style-type: none"> <li>Simbol untuk mengawali suatu proses.</li> </ul>	
2.	<p><i>Intermediate Event</i></p> <ul style="list-style-type: none"> <li>Simbol yang memengaruhi alur dari proses tapi tidak akan memulai atau menghentikan proses secara langsung.</li> </ul>	
3.	<p><i>End Event</i></p> <ul style="list-style-type: none"> <li>Simbol untuk mengakhiri suatu proses.</li> </ul>	
4.	<p><i>Start Timer</i></p> <ul style="list-style-type: none"> <li>Simbol spesifikasi tanggal dan waktu untuk memulai proses tersebut.</li> </ul>	
5.	<p><i>Intermediate Timer</i></p> <ul style="list-style-type: none"> <li>Simbol untuk menunjukkan berapa lama waktu selama proses itu dilakukan.</li> </ul>	
6.	<p><i>Start Message</i></p> <ul style="list-style-type: none"> <li>Simbol pesan untuk men-trigger sebuah proses.</li> </ul>	
7.	<p><i>Intermediate Message</i></p> <ul style="list-style-type: none"> <li>Simbol pesan untuk menunjukkan sesuatu yang terjadi selama proses.</li> </ul>	
8.	<p><i>End Message</i></p> <ul style="list-style-type: none"> <li>Simbol pesan untuk mengakhiri sebuah proses.</li> </ul>	

Tabel 2. 7 Elemen-elemen BPMN (Lanjutan)

No.	Keterangan	Simbol
9.	<p><i>Exclusive Gateway</i></p> <ul style="list-style-type: none"> <li>Simbol untuk menunjukkan adanya 1 keputusan yang akan diambil dalam proses.</li> </ul>	
10.	<p><i>Inclusive Gateway</i></p> <ul style="list-style-type: none"> <li>Simbol untuk menunjukkan keputusan dimana ada lebih dari 1 kemungkinan hasil yang terjadi.</li> </ul>	
11.	<p><i>Parallel Gateway</i></p> <ul style="list-style-type: none"> <li>Simbol untuk menunjukkan adanya dua aktivitas yang akan dilakukan bersamaan.</li> </ul>	
12.	<p><i>Swimlane</i></p> <ul style="list-style-type: none"> <li>Merupakan partisi sub-dalam Proses. Kegunaannya untuk mengelompokkan aktivitas.</li> </ul>	
13.	<p><i>Sequence Flow</i></p> <ul style="list-style-type: none"> <li>Menunjukkan urutan aktivitas yang akan dilakukan dalam proses tersebut.</li> </ul>	
14.	<p><i>Association</i></p> <ul style="list-style-type: none"> <li>Menghubungkan satu aktivitas dengan aktivitas lainnya.</li> </ul>	
15.	<p><i>Activity</i></p> <ul style="list-style-type: none"> <li>Merepresentasikan suatu kegiatan yang dilakukan dalam proses tersebut.</li> </ul>	

Sumber: Weske, Mathias. (2012). Business Process Management: Concept, Languages, Architectures. German: Springer.

## **2.7 Hypertext Markup Language (HTML)**

HTML merupakan *Markup language* untuk *World Wide Web*. Bahasa HTML mendefinisikan format suatu dokumen WWW dan memungkinkan *hypertext link* bergabung dengan dokumen tersebut. Dalam perkembangan HTML terdapat penambahan-penambahan yang dikenal dengan sebutan *Dynamic HTML*. Dengan adanya perkembangan bahasa lanjutan ini memungkinkan HTML menampilkan informasi yang bersifat statis serta dinamis [9].

## **2.8 PHP Hypertext Preprocessor (PHP)**

PHP merupakan singkatan dari *Hypertext Preprocessor*. PHP adalah bahasa berbentuk *script* yang ditempatkan dan diproses di dalam *server*. Hasil dari proses itu yang dikirim kepada *client*, tempat pemakai menggunakan *browser*. Asal usul PHP dimulai pada tahun 1994. Saat itu, Rasmus Lerdorf membuat sejumlah *script* Perl yang mempunyai kemampuan untuk mengamati siapa saja yang melihat daftar riwayat hidupnya. Rasmus Lerdorf kemudian mengembangkan *script* tersebut menjadi *tool* yang dinamakan dengan “Personal Home Page”. Paket yang telah diproses itu menjadi asal mula PHP. Pada tahun 1995, Rasmus menciptakan PHP/FI versi 2. Pada versi inilah *programmer* dapat meletakkan kode terstruktur di dalam tag HTML. Kode PHP juga dapat melakukan komunikasi dengan *database* dan juga melakukan perhitungan-perhitungan yang kompleks (Kadir, 2008:2). PHP adalah bahasa pemrograman berbentuk *script* yang memiliki fungsi-fungsi dalam membentuk sistem yang sistematis [10].

## **2.9 Cascading Style Sheet (CSS)**

CSS merupakan singkatan dari *Cascading Style Sheet*. Kegunaan CSS adalah untuk memberi nilai estetika pada penampilan HTML atau menentukan penampilan elemen-elemen yang ada pada halaman HTML, contohnya seperti merubah warna dan ukuran teks, menentukan posisi foto, dan merubah warna dan ukuran *background*.



Cara penulisan kode dalam CSS juga bermacam-macam. Penulisannya dibagi menjadi 3 jenis yaitu *inline*, *internal* dan *external*. Ketiganya bisa digunakan sesuai dengan apa yang diperlukan oleh *programmer* [11]. Berikut ini adalah penjelasan dari ketiga jenis penulisan tersebut:

1. *Inline*

Penulisan kode CSS dengan metode *inline* ini dilakukan langsung pada *tag* HTML yang ingin diberi *style* dengan menggunakan atribut *style*.

2. *Internal*

Penulisan kode CSS dengan metode *internal* dilakukan di dalam *tag style* yang terletak dalam *tag head*. Metode *internal* sangat disarankan untuk pengujian *style* saat *programmer* hanya mempunyai satu saja halaman *web*.

3. *External*

Penulisan kode CSS dengan metode *external* dilakukan dengan cara membuat *file* CSS yang terpisah dari file HTML itu sendiri. Cara menggunakan CSS nya adalah dengan memanggilnya dalam *tag head*. *File* CSS memiliki ekstensi *.CSS* misalnya *namafilekamu.CSS*.

## 2.10 JavaScript

*JavaScript* merupakan bahasa pemrograman yang pada mulanya dikembangkan oleh *Netscape Communication*. Kini, *javascript* sudah menjadi standar untuk hampir seluruh *web browser* yang ada. *JavaScript* juga termasuk bahasa pemrograman seperti halnya *Python*, *C#* dan *Visual Basic*.

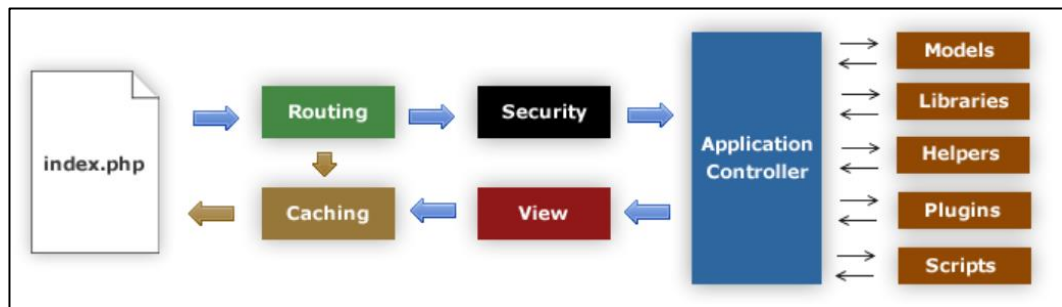
*JavaScript* adalah bahasa pemrograman *scripting*, hal ini menjadikan *javascript* tidak membutuhkan proses kompilasi. Jenis bahasa ini memungkinkan program hanya berisi sebaris kode yang terdiri atas sebuah fungsi saja. *Javascript* tidak seperti bahasa pemrograman *Java*. *Java* sendiri merupakan sebuah *file* program yang harus mempunyai tempat atau bagian untuk menambahkan *namespace* dan deklarasi untuk nama *class* [12].

## 2.11 MySQL

*MySQL* merupakan sebuah *software* untuk sistem manajemen *database SQL*. *MySQL* sendiri *multiuser* dan sudah diinstalasi sekitar 6 juta di seluruh penjuru dunia. *MySQL AB* membuat *MySQL* tersedia sebagai *software* yang tidak dipungut biaya dengan lisensi GNU *General Public License* (GPL). Namun, mereka juga menjual lisensi komersial jika terjadi masalah penggunaan dari *MySQL* yang tidak cocok dengan lisensi GPL. Beda halnya dengan *Apache* dimana *Apache* merupakan *software* yang dikembangkan oleh komunitas umum. Hak cipta atas *source code Apache* dimiliki oleh penulisnya masing-masing. *MySQL* dimiliki dan disponsori oleh sebuah perusahaan komersial yang berasal dari Swedia yaitu *MySQL AB*. Hak cipta atas seluruh *source code MySQL* dipegang penuh oleh *MySQL AB*. Pendiri *MySQL AB* yaitu David Axmark, Allan Larsson, dan Michael "Monty" Widenius [13].

## 2.12 CodeIgniter

*CodeIgniter* merupakan sebuah *framework* yang digunakan untuk membuat sebuah *website*. *CodeIgniter* sendiri disusun dengan menggunakan bahasa PHP. Di dalam *CodeIgniter* terdapat beberapa macam *class* yang berbentuk *library* dan *helper*. *Library* dan *helper* berfungsi untuk membantu *programmer* dalam mengembangkan *website* yang mereka inginkan (Riyanto, 2011). Pada mulanya *CodeIgniter* dikembangkan oleh *EllisLab* sebagai sebuah *framework* PHP yang *open source*. *Framework* ini telah dirancang dengan dukungan konsep MVC (*Model, View, Controller*). Konsep MVC ini digunakan untuk membedakan antara kode tampilan, sehingga pemrosesan kode bisa lebih mudah diatur karena telah dipecah menjadi beberapa bagian. Gambar 2.2 dibawah ini menjelaskan cara kerja dari *CodeIgniter*.



Gambar 2. 3 Application Flow Chart CodeIgniter

Sumber: [https://www.codeigniter.com/user\\_guide/overview/appflow.html](https://www.codeigniter.com/user_guide/overview/appflow.html) diakses pada 15 April 2020.

Selain itu, *CodeIgniter* juga telah menyediakan berbagai *library* yang siap untuk dipakai. *Library* tersebut disediakan dengan tujuan agar memudahkan proses pembuatan aplikasi *website* menjadi lebih efektif dan efisien[14].

## 2.13 Penelitian Terdahulu

Tabel 2. 8 Penelitian Terdahulu

No.	Judul	Tahun	Pengarang	Metode Penelitian	Hasil Penelitian	Kesimpulan
1.	Rancang Bangun Sistem Informasi Penjualan dan Pembelian Alat Tulis Kantor ada CV. Putra Mandiri [2].	2020	Rifky Permana, Daning Nur Sulistyowati, Ani Oktarini Sari, Tika Adilah Mutiara	<ul style="list-style-type: none"> <li>• Observasi</li> <li>• Wawancara</li> <li>• Studi Pustaka</li> </ul>	Implementasi dari sistem informasi penjualan yang dibuat sangat membantu proses penjualan dan pembelian alat tulis kantor pada CV. Putra Mandiri. Sistem ini mencakup pengelolaan data pelanggan, <i>supplier</i> , transaksi serta barang yang dijual.	Dengan adanya implementasi Microsoft Visual Basic 6.0, maka proses transaksi penjualan dan pembelian pada CV. PUTRA MANDIRI lebih mudah. Keakuratan data pun lebih terjamin. Sistem yang dikembangkan ini juga dapat meningkatkan efektifitas dan efisiensi pengolahan data. Keamanan penyimpanan data dengan media komputer pun lebih aman. Kemudahan untuk mengubah dan mencari kembali data-data yang diperlukan jauh lebih mudah. Laporan harian, mingguan dan bulanan dapat dibuat lebih mudah.

Tabel 2. 9 Penelitian Terdahulu (Lanjutan)

No.	Judul	Tahun	Pengarang	Metode Penelitian	Hasil Penelitian	Kesimpulan
2.	Sistem Informasi Penjualan Berbasis Web pada UD Dwi Surya Aluminium dan Kaca Yogyakarta [16].	2019	Fanny Fatma Wati , Uswatun Khasanah	<ul style="list-style-type: none"> <li>• Observasi</li> <li>• Wawancara</li> <li>• Studi Pustaka</li> <li>• <i>Prototyping</i></li> </ul>	<p>Pada <i>website</i> yang sudah dibuat untuk kemudahan proses jual beli toko Surya Aluminium dan Kaca Yogyakarta, dihasilkan beberapa fitur untuk beberapa pengguna <i>website</i> yang dibagi menjadi pengunjung <i>web</i>, <i>member</i>, <i>driver</i>, <i>admin</i>. Setiap pengguna memiliki akses yang berbeda tergantung dengan perannya.</p> <p><i>Website</i> ini menyediakan layanan <i>chatting online</i> agar informasi mengenai toko maupun <i>website</i> dapat dijawab dan dijadikan <i>feedback</i> untuk toko. Halaman <i>website</i> juga dirancang sedemikian rupa agar pembeli dapat memberikan testimoni dari layanan yang mereka terima.</p>	Dengan sistem penjualan <i>online</i> berbasis <i>web</i> yang dibuat ini mampu mempermudah proses pembelian tanpa adanya halangan waktu serta mampu menjangkau seluruh wilayah. Sistem informasi penjualan ini dapat mengurangi biaya yang harus dikeluarkan untuk promosi produk. Sistem ini juga mempermudah pengolahan data seperti <i>update data</i> .
3.	Sistem Informasi Penjualan Sandal berbasis Web [17].	2018	Reza Fahlevi Ahmad, Novrini Hasti	<ul style="list-style-type: none"> <li>• Pendekatan Sistem</li> <li>• <i>Prototyping</i></li> </ul>	Setelah sistem berhasil diimplementasi, toko Cucko Bandung ini merasa terbantu sekali. Laporan toko yang tadinya dicatat secara manual dan sering hilang kini terintegrasi dengan adanya sistem laporan. Laporan yang dimaksud yaitu laporan stok, penjualan serta pendapatan dari toko.	<ol style="list-style-type: none"> <li>1. Sistem penjualan <i>online</i> ini dapat mempertahankan usaha dan bersaing dengan perusahaan lain.</li> <li>2. Dengan adanya sistem ini, data- data perusahaan menjadi terkomputerisasi.</li> <li>3. Media informasi pemesanan dan penjualan lebih informatif dan inovatif.</li> </ol>

Tabel 2. 10 Penelitian Terdahulu (Lanjutan)

No.	Judul	Tahun	Pengarang	Metode Penelitian	Hasil Penelitian	Kesimpulan
4.	Perancangan Sistem Penjualan Alat Tulis Kantor (ATK) berbasis <i>Web</i> [18].	2017	Aristotelianto	<ul style="list-style-type: none"> <li>• Observasi</li> <li>• Studi Pustaka</li> <li>• Metode <i>Waterfall</i></li> </ul>	Pemesanan merupakan hal yang sangat penting dalam proses toko. Maka dari itu, pada hasil penelitian ini dikembangkan fitur status pemesanan agar mudah untuk memantau pesanan yang dibuat. Selain itu juga ada fitur forum untuk pelanggan. Fitur-fitur umum pada proses toko juga tersedia seperti pengolahan data transaksi, data pengguna <i>website</i> , data produk. Ada juga halaman untuk mengetahui cara membayar barang-barang yang sudah dipesan	<ol style="list-style-type: none"> <li>1. Sistem yang terkomputerisasi memberi kemudahan bagi pelanggan untuk membeli ATK secara <i>online</i> tanpa harus memakan waktu.</li> <li>2. <i>Website</i> ini mampu meningkatkan penjualan dan mencakup pasar yang lebih luas.</li> <li>3. Kemudahan mendapatkan informasi bagi pelanggan tentang produk alat tulis kantor yang tersedia bagi pelanggan.</li> </ol>
5.	Sistem Informasi Penjualan Alat Tulis Kantor berbasis <i>Website</i> pada CV. Sumber Rezeki Jakarta [19].	2016	Yuly Rachmawati, Laila Septiana, Susana Dwi Yulianti	<ul style="list-style-type: none"> <li>• Studi Literatur</li> <li>• Observasi</li> </ul>	Hasil dari implementasi sistem yang dibuat adalah CV. Sumber Rezeki Jakarta memberi kemudahan dengan dibuatnya fitur-fitur yang mendukung proses operasional toko. Calon pembeli dapat membuat akun dan memesan barang sesuai dengan kategori yang diinginkan. Ada halaman konfirmasi pembayaran untuk memberi tahu bahwa pelanggan tersebut sudah membayar. Dari sisi <i>admin</i> , <i>admin</i> memiliki akses untuk mengelola data akun, produk, pelanggan, konfirmasi pembayaran serta pembuatan laporan di dalam <i>website</i> .	<ol style="list-style-type: none"> <li>1. Dengan adanya sistem informasi berbasis <i>website</i> dapat menghemat biaya promosi dan dapat melakukan transaksi secara online.</li> <li>2. Sistem pemesanan produk-produk secara <i>online</i> pada CV. Pelanggan tidak harus datang ke toko untuk memesan produk.</li> <li>3. Mampu memperluas pasar serta meningkatkan pendapatan perusahaan.</li> </ol>



Perbandingan penelitian terdahulu yang ada pada tabel 2.5 diatas membahas tentang perancangan sistem informasi penjualan berbasis *web*. Penelitian terdahulu juga dibuat dengan tujuan untuk melakukan pencatan data yang dibutuhkan toko agar dapat disimpan terpusat serta mengurangi kehilangan data-data penting. Data yang disimpan dalam *website* bertujuan untuk diolah menjadi informasi yang dapat meningkatkan kinerja perusahaan terutama dalam hal pengambilan keputusan. Sistem yang dirancang pada perbandingan penelitian terdahulu diatas tersebut, dirancang karena perusahaan masih melakukan proses penjualan dan pencatatan secara manual. Hal ini seringkali menyebabkan data yang dikelola tidak akurat serta tidak ada sistem yang terintegrasi untuk membantunya proses dalam perusahaan. Sehingga proses dalam toko dapat memakan waktu yang cukup lama dibandingkan apabila sudah ada sistem yang terintegrasi.

Kendala yang dialami toko Metro Stationary serupa dengan yang dialami perusahaan yang tertera pada tabel diatas, dimana pencatatan data dan laporan masih dilakukan secara manual. Perbedaan dari sistem penjualan yang akan dirancang untuk keperluan toko Metro Stationary dengan perusahaan yang tertera namanya pada tabel diatas adalah toko Metro Stationary akan menggunakan sistem untuk mengestimasi biaya kirim dari kurir pribadi milik toko. Estimasi biaya kirim akan diukur sesuai dengan harga yang ditentukan dikalikan dengan jarak yang ada pada *Google Maps*. Tujuannya supaya ada biaya pasti untuk pengiriman dari kurir milik toko. Sistem yang dirancang untuk toko Metro Stationary juga dapat memberikan laporan keuangan. Fitur-fitur lainnya yang dirancang adalah fitur pemesanan *online*, pengecekan stok, pengecekan status pengiriman, dan perhitungan kasir.