

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Teori Umum

2.1.1 Perancangan

Menurut (Nasril & Saputra, 2016) Perancangan adalah proses pengembangan spesifikasi sistem baru berdasarkan rekomendasi hasil analisis sistem”. Dalam tahap perancangan, tim kerja harus merancang dalam berbagai kertas dengan kebutuhan pengguna (*end user*) melalui alat perancangan sistem yang terstandarisasi. Kertas kerja yang dimaksud memuat berbagai uraian mengenai *input*, proses, dan *output* dari sistem yang akan diusulkan.

Sedangkan menurut (Waluyo & Fatich, 2017) Perancangan sistem adalah proses menyusun atau mengembangkan sistem informasi yang baru. Dalam tahap ini harus dapat dipastikan bahwa semua persyaratan untuk menghasilkan sistem informasi dapat dipenuhi. Hasil sistem yang dirancang harus sesuai dengan kebutuhan pemakai untuk mendapatkan informasi.

2.1.2 Sistem

Menurut (Rahmawati, Andini, & Zefriyenni, 2016) Sistem berasal dari bahasa Yunani yaitu *systema* yang berarti kesatuan atau sekumpulan. Terdapat dua pendekatan dalam mendefinisikan sistem yaitu dengan menekankan pada prosedurnya dan menekankan pada komponen atau elemennya. Berikut definisi sistem dari dua pendekatan tersebut:

1. Yang mengarah pada prosedurnya.

Suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

2. Yang mengarah pada komponen dan elemennya.

Kumpulan elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

Sedangkan menurut (Lestari, Eko, & Heru, 2017) Secara umum sistem merupakan kumpulan objek atau elemen yang saling berinteraksi untuk mencapai satu tujuan. Sistem sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Dan pendapat lain dari sistem yang penulis kutip yaitu menurut (Rosliana, Herlawati, & Supriyatna, 2015) Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.

2.1.3 Informasi

Menurut (Haryanta, Rochman, & Setyaningsih, 2017) Informasi adalah data yang telah diklasifikasikan atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah data dari bentuk tak berguna menjadi berguna bagi penerimanya. Sumber informasi adalah data. Sedangkan menurut (Lestari et al., 2017) Informasi merupakan data yang dirubah menjadi bentuk yang lebih

berguna dan berarti bagi pihak yang menerimanya. Sedangkan data adalah suatu kenyataan yang menggambarkan suatu kejadian-kejadian dari kesatuan nyata. Suatu sistem yang kurang mendapat informasi akan menjadi luruh, kerdil dan akhirnya berakhir.

Dan menurut (Rosliana et al., 2015) Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tak berguna menjadi berguna bagi yang menerimanya.

2.1.4 Sistem informasi

Menurut (Kinanti, 2017) Sistem informasi adalah cara yang terorganisir untuk mengumpulkan, memasukan, dan memproses data dan menyimpan, mengelola, mengontrol dan melaporkannya sehingga dapat mendukung suatu perusahaan maupun organisasi untuk mencapai tujuan. Sedangkan menurut (Lestari et al., 2017) Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategis dari suatu organisasi untuk dapat meyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan. Dan menurut (Rahmawati et al., 2016) Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

2.1.5 Persediaan

Menurut (Aristiyanto, Putri, & Adi, 2016) persediaan dapat didefinisikan sebagai barang yang berupa bahan baku, barang setengah jadi, barang jadi, atau *parts* yang disimpan untuk kontinuitas proses produksi dalam memenuhi kebutuhan pelanggan. Oleh karena itu, persediaan merupakan salah satu hal yang sangat penting dalam keberhasilan suatu perusahaan. Sedangkan menurut (Tamodia, 2013) pengertian persediaan merupakan barang-barang yang dimiliki untuk kemudian dijual atau digunakan dalam proses produksi atau dipakai untuk keperluan *non* produksi dalam siklus kegiatan yang normal.

Dan pendapat lain mengenai persediaan menurut (Irwadi, 2015) Persediaan adalah sejumlah barang jadi, bahan baku, dan barang dalam proses yang dimiliki perusahaan dengan tujuan untuk dijual atau diproses lebih lanjut. Berdasarkan pengertian di atas dapat disimpulkan bahwa pengertian persediaan adalah barang yang dibeli atau diproduksi oleh perusahaan dengan tujuan untuk dijual kepada pelanggan atau pembeli.

2.1.6 Fungsi Persediaan

Menurut (Salangka, 2015) Fungsi persediaan adalah sebagai berikut:

1. Fungsi *Decoupling* adalah persediaan yang memungkinkan perusahaan dapat memenuhi permintaan pelanggan tanpa tergantung pada *supplier*.
2. Fungsi *Economic Lot Sizing*, persediaan ini perlu mempertimbangkan penghematan atau potongan pembelian, biaya pengangkutan per unit menjadi lebih murah dan sebagainya.

3. Fungsi Antisipasi, apabila perusahaan menghadapi fluktuasi permintaan yang dapat diperkirakan dan diramalkan berdasarkan pengalaman atau data-data masa lalu yaitu permintaan musiman.

2.1.7 Jenis-Jenis Persediaan

Jenis-jenis persediaan yang pada umumnya dikelompokkan menjadi beberapa bagian dan menurut (Salangka, 2015) persediaan dapat dikelompokkan ke dalam empat jenis, yaitu:

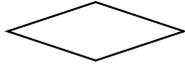
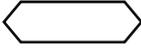
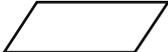
1. *Fluctuation stock*, merupakan persediaan yang dimaksudkan untuk menjaga terjadinya fluktuasi permintaan yang tidak diperkirakan sebelumnya, dan untuk mengatasi bila terjadi kesalahan/penyimpangan dalam prakiraan penjualan, waktu produksi, atau pengiriman barang.
2. *Anticipation stock*, merupakan persediaan untuk menghadapi permintaan yang dapat diramalkan pada musim permintaan tinggi, tetapi kapasitas produksi pada saat itu tidak mampu memenuhi permintaan. Persediaan ini juga dimaksudkan untuk menjaga kemungkinan sukarnya diperoleh bahan baku sehingga tidak mengakibatkan terhentinya produksi.
3. *Lot-size inventory*, merupakan persediaan yang diadakan dalam jumlah yang lebih besar daripada kebutuhan saat itu. Persediaan dilakukan untuk mendapatkan keuntungan dari harga barang (berupa diskon) karena membeli dalam jumlah yang besar, atau untuk mendapatkan penghematan dari biaya pengakutan per unit yang lebih rendah.
4. *Pipeline inventory*, merupakan persediaan yang dalam proses pengiriman dari tempat asal ke tempat dimana barang tersebut akan digunakan.

Misalnya, barang yang dikirim dari pabrik menuju tempat penjualan, yang dapat memakan waktu beberapa hari atau minggu.

2.1.8 Flowchart

Menurut (Rahman & Julianto, 2015) *flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. Berikut tabel dari bentuk *flowchart* yang biasa digunakan.

Tabel 2. 1 Flowchart

| Simbol | Fungsi |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
|  | Permulaan sub program |
|  | Perbandingan, pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya |
|  | Penghubung bagian-bagian <i>flowchart</i> yang berada pada satu halaman |
|  | Penghubung bagian-bagian <i>flowchart</i> yang berada pada halaman berbeda |
|  | Permulaan/akhir program |
|  | Arah aliran program Proses |
|  | Proses inisialisasi/pemberian harga awal |
|  | Proses penghitung/ proses pengolahan data |
|  | Proses <i>input/output</i> data |

Sumber: (Rahman & Julianto, 2015)

2.1.9 UML (*Unified Modeling Language*)

Menurut (Rahmawati et al., 2016) UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek. Secara fisik UML adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG (*Object Management Group*). OMG mengajukan proposal pada tahun 1996 agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 UML diakomodasikan oleh OMG sehingga sampai saat ini UML telah memberikan kontribusinya yang cukup besar dalam metodologi berorientasi objek dan hal-hal yang terkait dalamnya.

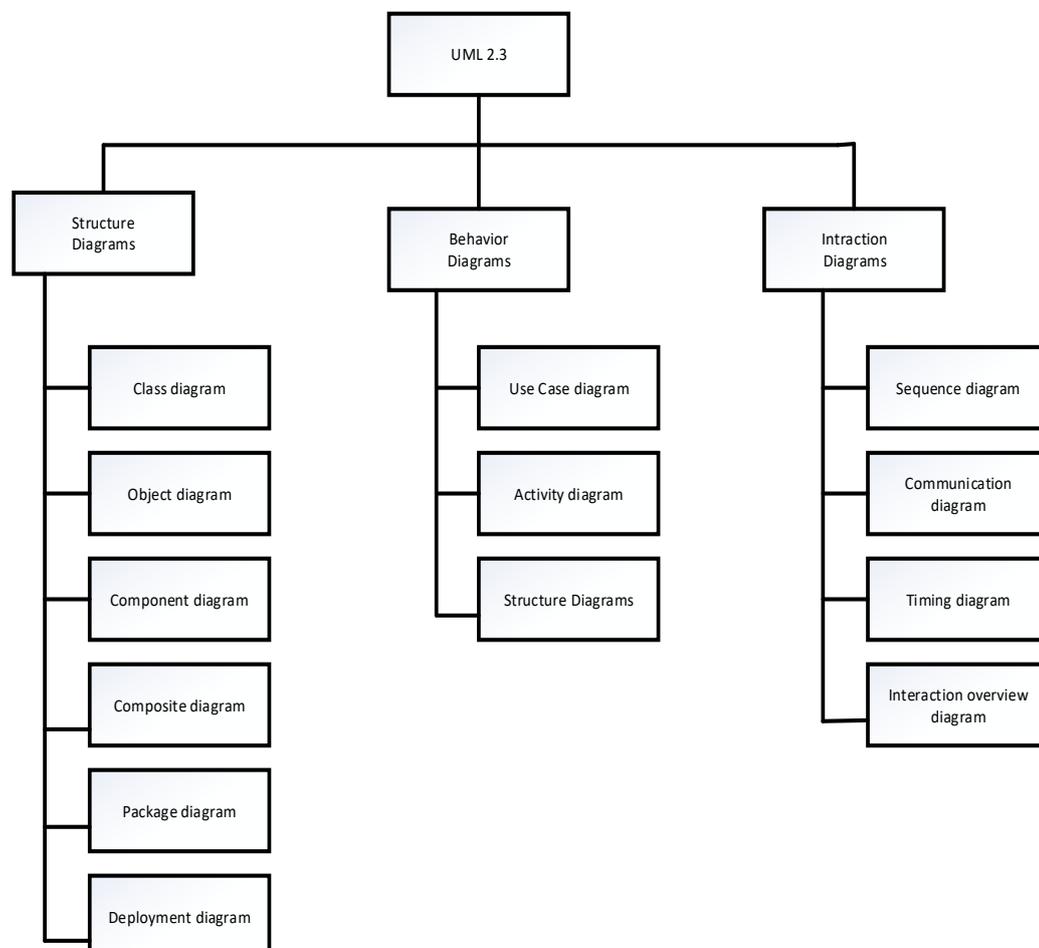
Sedangkan menurut (Kesumaningtyas, 2017) UML (*Unified Modelling Language*) adalah sebuah bahasa untuk menentukan visualisasi, konstruksi dan mendokumentasikan artifact (bagian dari informasi yang digunakan atau dihasilkan dalam suatu pembuatan perangkat lunak. Artifact dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan permodelan.

Dan menurut (Rahman & Julianto, 2015) *Unified modelling language* (UML) merupakan salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang sangat berorientasi objek UML dapat digunakan pada

setiap bahasa pemrograman, tapi UML akan lebih cocok pada setiap bahasa pemrograman yang berorientasi objek.

2.1.10 Diagram UML

Menurut (Syahriani, 2017) UML adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Berikut ini adalah pembagian kategori dan macam-macam diagram.



Sumber: (Syahriani, 2017)

Gambar 2. 1 Diagram UML 2.3

Dan pada penelitian ini penulis mengambil 4 jenis diagram UML untuk perancangan sistem yang akan digunakan yaitu sebagai berikut:

1. *Use Case Diagram*

Menurut (Urva & Siregar, 2015) *Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case diagram*, yaitu:

Tabel 2. 2 *Use Case Diagram*

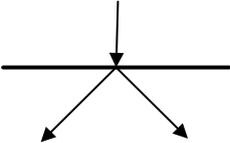
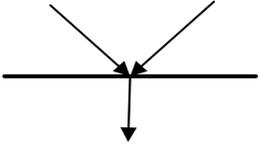
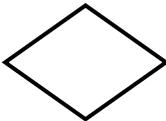
| Gambar | Keterangan |
|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> |
|  | Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> . |
|  | Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data. |
|  | Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor |
|  | <i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program. |
|  | <i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi |

Sumber: (Urva & Siregar, 2015)

2. Diagram Aktivitas (*Activity Diagram*)

Menurut (Aprianti & Maliha, 2016) *Activity Diagram* menggambarkan *workflow* (aliran kerja) dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Dan menurut (Urva & Siregar, 2015) *Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* yaitu:

Tabel 2. 3 *Activity Diagram*

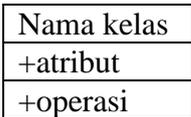
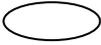
| Gambar | Keterangan |
|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas. |
|  | <i>End point</i> , akhir aktifitas. <i>Activites</i> , |
|  | <i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis. |
|  | <i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu. |
|  | <i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi. |
|  | <i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> . |
|  | <i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa. |

Sumber: (Urva & Siregar, 2015)

3. *Class Diagram* (Diagram Kelas)

Menurut (Urva & Siregar, 2015) *Class Diagram* Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Dan menurut (Aprianti & Maliha, 2016) *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan di buat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Simbol-simbol yang ada pada *class diagram* ditunjukkan yaitu:

Tabel 2. 4 *Class Diagram*

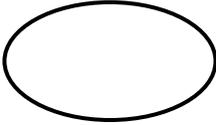
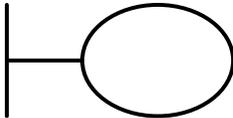
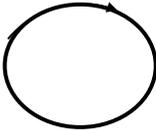
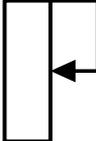
| Simbol | Diskripsi |
|----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| Kelas  | Kelas pada struktur <i>system</i> |
| Antarmuka/ <i>interface</i>  Nama <i>interface</i> | Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek. |
| Asosiasi/ <i>association</i>  | Relasi antarkelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i> . |
| Asosiasi berarah/ <i>directed association</i>  | Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |
| Generalisasi  | Relasi antarkelas dengan makna generalisasi- spesialisasi (umum khusus). |
| Kebergantungan/ <i>dependensy</i>  | Relasi antarkelas dengan makna kebergantungan antarkelas. |
| Agregasi/ <i>agregation</i>  | Relasi antarkelas dengan makna semua bagian |

Sumber: (Aprianti & Maliha, 2016)

4. Diagram Urutan (*Sequence Diagram*)

Menurut (Urva & Siregar, 2015) *Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* yaitu:

Tabel 2. 5 *Sequence Diagram*

| Gambar | Keterangan |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data. |
|  | <i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>formentry</i> dan <i>form</i> cetak. |
|  | <i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek. |
|  | <i>Message</i> , simbol mengirim pesan antar <i>class</i> . <i>Recursive</i> , |
|  | <i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri. |
|  | <i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi. |
|  | <i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> . |

Sumber: (Urva & Siregar, 2015)

2.2 Tinjauan Teori Khusus

2.2.1 PHP

Menurut (Jamaludin & Sugiarto, 2017) PHP merupakan kependekan dari *Hypertext Preprocessor*. PHP tergolong sebagai perangkat lunak *open source* yang diatur dalam aturan *general purpose license (GPL)*". Pemrograman PHP sangat cocok dikembangkan dalam lingkungan *web*, karena PHP bisa dilekatkan pada *script* HTML atau sebaliknya. PHP tergolong juga sebagai bahasa pemrograman yang berbasis *server (server side scripting)*.

Sedangkan menurut (Rahman & Julianto, 2015) PHP adalah pemrograman interpreter yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. Berbeda dengan html yang *source* kodenya di tampilkan di *website*, *source code* PHP tidak di tampilkan di halaman muka suatu *website* karena PHP di olah dan diproses di *server*, PHP mampu berjalan di berbagai sistem operasi seperti windows, Linux, Mac OS, dll. PHP memiliki kedinamisan dalam hal *database* yang bisa dihubungkan dengan PHP seperti MySQL, Oracle, MS Access, PostgreSQL. Namun untuk pemrograman website yang paling sering digunakan adalah MySQL. PHP sampai sekarang sudah mengalami perkembangan yang pesat dan sudah mencapai PHP 5.5. Untuk mengawali kode dalam PHP menggunakan kode `<?>` dan diakhiri tanda `?>`. Dan menurut (Sidik, Sutarman, & Marlenih, 2013) PHP adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan php harus menggunakan *web server*.

2.2.2 HTML (*Hypertext Markup Language*)

Menurut (Jamaludin & Sugiarto, 2017) Pengertian HTML adalah bahasa pemrograman dasar yang dipakai untuk menampilkan informasi pada halaman *web*. HTML menampilkan informasi dalam bentuk *hypertext* dan sesuai dengan namanya. Bahasa ini menggunakan tanda (*markup*) disebut dengan *tag* untuk menandai perintah-perintahnya. Sedangkan menurut (Sugiyani & Risdiyani, 2017) HTML (*Hyper Text Markup Language*) adalah sekumpulan simbol-simbol atau tag-tag yang dituliskan dalam sebuah *file* yang digunakan untuk menampilkan halaman pada *web browser*. Tag-tag HTML selalu diawali dengan `<x>` dan diakhiri dengan `</x>` dimana x tag HTML itu seperti b, i, u, dan lain-lain.

Dan menurut (Qotimah, 2017) HTML merupakan bahasa yang digunakan untuk menulis halaman *web*. Biasanya mempunyai ekstensi “.htm”, “.html”, atau “.shtml”. HTML tersusun atas tag-tag, digunakan untuk menentukan tampilan dari dokumen HTML yang diterjemahkan oleh *browser*. Tag HTML tidak *case sensitive*, jadi bisa menggunakan `<HTML>` atau `<html>`. Keduanya menghasilkan *output* yang sama.

2.2.3 CSS (*Cascading Style Sheet*)

Menurut (Jamaludin & Sugiarto, 2017) Dalam teknik pembuatan *layout* HTML terbagi menjadi beberapa tahapan sejarah yang berkembang dari waktu ke waktu seperti *Frame* adalah cara lama yang sekarang ini sudah jarang digunakan, kemudian berkembang ke teknologi tabel yang disukai oleh sebagian besar *designer web* yang ada, dan kini berubah menjadi teknologi CSS yang merupakan metode baru yang sekarang ini menjadi standar dalam suatu

pembuatan *web*. CSS (*Cascading Style Sheet*) secara sederhana adalah sebuah metode yang digunakan untuk mempersingkat penulisan tag HTML seperti *font*, *color*, *text*, *table*, menjadi lebih ringkas sehingga tidak terjadi pengulangan penulisan. *Cascading Style Sheet* (CSS) adalah suatu kumpulan kode-kode untuk memformat, yang mengendalikan tampilan isi dalam suatu halaman *web*". Dengan menggunakan CSS, maka dalam pengaturan tampilan website akan lebih efisien. Karena CSS dapat dibuat terpisah dengan halaman HTML. Sehingga satu dokumen CSS dapat digunakan berulang kali pada halaman lainnya.

2.2.4 Adobe Dreamweaver

Menurut (Sidik et al., 2013) Adobe Dreamweaver adalah perangkat lunak terkemuka untuk desain *web* yang menyediakan kemampuan visual yang intuitif termasuk pada tingkat kode, yang dapat digunakan untuk membuat dan mengedit website HTML serta aplikasi mobile seperti *smartphone*, *tablet*, dan perangkat lainnya. Dengan adanya fitur *layout Fluit Grid* yang dirancang khusus untuk memungkinkan lintas *platform*, maka akan membuat *layout* adaptif atau dapat menyesuaikan dengan *browser* yang dipakai.

Sedangkan menurut (Jamaludin & Sugiarto, 2017) Adobe Dreamweaver adalah perangkat lunak untuk mendesain *web* yang menyediakan kemampuan visual yang intuitif termasuk pada tingkat kode, yang dapat digunakan untuk membuat dan mengedit website HTML serta aplikasi *mobile* seperti *smartphone*, *tablet*, dan perangkat lainnya. Adobe Dreamweaver dikembangkan oleh perusahaan *software* Adobe Inc. Saat ini versi yang beredar adalah Adobe Dreamweaver CS6 yang memiliki banyak fitur, antara lain:

- a. *Layout Fluid Grid*
- b. Peningkatan *support jQuery Mobile*
- c. Transisi CSS3 dan HTML5
- d. Fitur *Live View* diperbarui
- e. Mendukung integrasi CMS
- f. Isyarat kode yang lebih spesifik

2.2.5 MySQL

Menurut (Nasril & Saputra, 2016) MySQL merupakan RDBMS atau *server database* yang mengelola *database* dengan cepat menampung dalam jumlah sangat besar dan dapat diakses oleh banyak *user*. Sedangkan pendapat lain menurut (Sidik et al., 2013) MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi *web*. Contoh DBMS lainnya adalah: PostgreSQL (*freeware*), SQL Server, MS Access dari Microsoft, DB2 dari IMB, Oracle dan Oracle Crop, Dbase, FoxPro, dsb. MySQL adalah sistem manajemen basis data relasi yang bersifat terbuka atau *open source*. Sistem manajemen basis data ini adalah hasil pemikiran dari Michael “Monty” Widenius, David Axmark, dan Allan Larson pada tahun 1995. Tujuan awal ditulisnya program MySQL adalah untuk mengembangkan aplikasi *web*. MySQL menggunakan bahasa standar SQL (*Structure Query Language*) sebagai bahasa interaktif dalam mengelola data. Perintah SQL sering juga disebut *Query*. MySQL menawarkan berbagai keunggulan dibandingkan database server lain. Berikut ini adalah beberapa keunggulan MySQL:

1. Mampu menangani jutaan *user* dalam waktu yang bersamaan.

2. Mampu menampung lebih dari 50.000.000 *record*.
3. Sangat cepat mengeksekusi perintah.

Dan menurut (Chrystanti & Wardati, 2011) MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multiuser*. MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. MySQL yang *free software* bebas digunakan untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensi, yang berada di bawah lisensi GNU/GPL (*general public license*).

2.2.6 XAMPP

Menurut (Sidik et al., 2013) XAMPP merupakan paket PHP berbasis *open source* yang dikembangkan oleh komunitas *Open Source*, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan pearl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Pearl. Program ini tersedia dalam GNU (*General Public License*) dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis. Sedangkan menurut (Sugiyani & Risdiyani, 2017) XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MySQL di komputer *local*". XAMPP berperan sebagai *server web* pada komputer. XAMPP juga dapat disebut sebuah *panel server virtual*, yang dapat membantu anda melakukan *preview* sehingga dapat

meemodifikasi *website* tanpa harus *online* atau terakses dengan *internet*. Dan menurut (Nasril & Saputra, 2016) XAMPP (X Apache MySQL PHP Perl) merupakan paket PHP dan Mysql berbasis *open source* yang dapat digunakan sebagai *tool* pembantu pengembangan aplikasi berbasis PHP.

2.2.7 Database

Menurut (Rahman & Julianto, 2015) Data merupakan fakta mengenai suatu objek seperti manusia, benda, peristiwa, konsep, keadaan dan sebagainya yang dapat dicatat dan mempunyai arti secara implisit. Data dapat dinyatakan dalam bentuk angka, karakter atau simbol, sehingga bila data dikumpulkan dan saling berhubungan maka dikenal dengan istilah basis data (*database*). Sedangkan menurut (Qotimah, 2017) Basis data atau *database* adalah sekumpulan data yang saling berhubungan yang menjelaskan aktivitas suatu organisasi atau dapat juga didefinisikan sebagai sekumpulan data yang saling berhubungan dan menjadi bagian dari suatu *Database Management System*.

Elemen penting di dalam sistem basis data adalah basis data sebagai inti dari sistem basis data, perangkat lunak (*software*) sebagai perancangan dan pengelolaan basis data, perangkat keras (*hardware*) sebagai pendukung operasi pengolahan data, manusia (*brainware*) yang mempunyai elemen penting dalam sistem tersebut. Basis data terdiri dari satu atau lebih tabel yang saling berhubungan. Anda atau *user* mempunyai wewenang untuk mengakses data tersebut, baik untuk menambah, mengubah atau menghapus data yang ada di dalam tabel tersebut. Saat ini banyak perangkat lunak *database* yang menggunakan SQL sebagai perintah untuk mengakses data. Beberapa diantaranya:

DB2, Ingress, Informix, ORACLE, Ms. Acces, MySQL, PostgreSQL, Rdb, Sybase. Dalam Sistem Informasi *E-procurement* ini kami menggunakan MySQL. Dan menurut (Hendini, 2016) *Database* juga merupakan kumpulan data yang umumnya menggambarkan aktifitas-aktifitas dan pelakunya dalam suatu organisasi. Sistem *database* merupakan sistem komputer yang digunakan untuk menyimpan dan mengelola data tersebut.

2.2.8 Web

Menurut (Nasril & Saputra, 2016) *Web* adalah layanan yang didapat oleh pemakai komputer yang terhubung ke *internet*, sedangkan menurut (Sidik et al., 2013) *Web* adalah salah satu aplikasi yang berisikan dokumen – dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan *protocol HTTP (hypertext transfer protocol)* dan untuk mengaksesnya menggunakan perangkat lunak yang disebut *browser*.

2.2.9 Analisis SWOT

Menurut (Ruhmana & Saputra, 2013) Analisis Swot adalah identifikasi berbagai factor secara sistematis untuk merumuskan strategi perusahaan. Analisis ini didasarkan pada logika yang dapat memaksimalkan kekuatan (*Strengths*) dan peluang (*Opportunities*), namun secara bersamaan dapat meminimalkan kelemahan (*Weaknesses*) dan ancaman (*Threats*). Proses pengambilan keputusan strategi selalu berkaitan dengan pengembangan misi, tujuan, strategis, dan kebijakan perusahaan. Dengan demikian, perencanaan strategis (*strategic planner*) harus menganalisis faktor-faktor strategis perusahaan (kekuatan, kelemahan, peluang, dan ancaman) dalam kondisi yang ada saat ini.

Sedangkan menurut Analisis SWOT adalah suatu cara mengidentifikasi berbagai faktor secara sistematis dalam rangka merumuskan strategi perusahaan. Analisis SWOT mempertimbangkan faktor lingkungan internal *strengths* dan *weaknesses* serta lingkungan eksternal *opportunities* dan *threats* yang dihadapi dunia bisnis. Analisis SWOT didahului dengan indentifikasi posisi perusahaan melalui evaluasi nilai faktor eksternal. Analisis SWOT terdiri dari empat faktor, yaitu:

1. *Strengths* (kekuatan)

Merupakan kondisi kekuatan yang terdapat dalam organisasi, proyek atau konsep bisnis yang ada. Kekuatan yang dianalisis merupakan faktor yang terdapat dalam tubuh organisasi, proyek atau konsep bisnis itu sendiri.

2. *Weakness* (kelemahan)

Merupakan kondisi kelemahan yang terdapat dalam organisasi, proyek atau konsep bisnis yang ada. Kelemahan yang dianalisis merupakan faktor yang terdapat dalam tubuh organisasi, proyek atau konsep bisnis itu sendiri.

3. *Opportunities* (peluang)

Merupakan kondisi peluang berkembang di masa datang yang terjadi. Kondisi yang terjadi merupakan peluang dari luar organisasi, proyek atau konsep bisnis itu sendiri. Misalnya kompetitor, kebijakan pemerintah, kondisi lingkungan sekitar.

4. *Threats* (ancaman)

Merupakan kondisi yang mengancam dari luar. Ancaman ini dapat mengganggu organisasi, proyek atau konsep bisnis itu sendiri.

2.2.10 SDLC (*System Development Life Cycle*)

Menurut (Lestari et al., 2017) Pengertian SDLC adalah suatu proses berkelanjutan dari planning, analisis, desain dan implementasi. Yang pada setiap prosesnya dilakukan perbaikan secara bertahap. Sedangkan menurut (Setyorini & Hakim, 2016) pola sirkulasi dari siklus hidup sistem, masalah didefinisikan dalam tahap perencanaan dan analisis, kemudian solusi alternatif diidentifikasi dan dievaluasi dalam tahap desain. Tahap selanjutnya, solusi terbaik diimplementasikan dan digunakan. Selama tahap penggunaan, umpan balik informasi dikumpulkan untuk melihat seberapa baik sistem dapat memecahkan masalah yang dihadapi.

Dan menurut (Amborowati & Marco, 2016) Dalam membangun suatu rekayasa piranti lunak, diperlukan tahap-tahap. Sistem yang secara luas digunakan adalah *System Development Life Cycle* (SDLC), yang meliputi beberapa tahap, yaitu:

1. Rekayasa Sistem

Karena *software* merupakan bagian dari suatu sistem, maka dimulai dengan penetapan semua sistem elemen dan mengalokasikan beberapa bagiannya ke dalam usulan pada *software* kemudian menggabungkan semua level sistem dengan melakukan pengkajian dari level atas dalam pendesainan dan analisis.

2. Analisis Kebutuhan *Software*

Merupakan proses mengerti tentang domain informasi, fungsi, kinerja, dan tatap muka pada *software*.

3. Desain

Pada desain, prinsipnya adalah mengubah kebutuhan menjadi *software* yang layak dari segi kualitas sebelum proses pengkodean.

4. Pengkodean

Proses pengkodean yaitu mengubah ke dalam bentuk yang dapat dibaca oleh mesin.

5. Pengetesan

Proses yang memastikan semua kalimat dalam program telah dilakukan pengetesan sehingga memberikan input sesuai dengan yang diinginkan.

6. Pemeliharaan

Software akan mengalami perubahan setelah dikirim ke pengguna, maka proses pemeliharaan dilakukan dengan menerapkan setiap langkah daur hidup sebelumnya disertai dengan perbaikan

2.3 Penelitian Terdahulu

Berikut ini adalah kesimpulan beberapa jurnal penelitian terdahulu yang dipakai dalam pembuatan dan penyusunan skripsi ini, yaitu:

1. Peneliti (Hendini, 2016). Jurnal Khatulistiwa Informatika, Vol. Iv, No. 2 Desember 2016 dengan judul Pemodelan Uml Sistem Informasi Monitoring Penjualan Dan Stok Barang (Studi Kasus: Distro Zhezha Pontianak). Prosedur sistem dengan menggunakan program UML (*Unified Modeling Language*). Kesimpulan Dengan adanya sistem informasi monitoring

penjualan dan stok barang ini, mempermudah pelaku usaha dalam memantau atau mengetahui penjualan dan stok barang di tiap cabang.

2. Peneliti (Setyorini & Hakim, 2016). Jurnal Teknologi Informasi dan Komunikasi, ISSN:2087-0868, Volume 7 Nomor 2 September 2016 dengan judul Rancang Bangun Sistem Informasi Persediaan Gudang Untuk Analisis Kebutuhan Barang (Studi Kasus Pada Dinas Penerangan Jalan Dan Pengelolaan Reklame Kota Semarang). Prosedur sistem dengan menggunakan program UML (*Unified Modeling Language*). Kesimpulan Sistem Informasi Persediaan Gudang Dinas Penerangan Jalan dan Pengelolaan Reklame Kota Semarang membantu Kepala Sie Pemeliharaan dalam menentukan kebutuhan barang pada tahun anggaran berikutnya.
3. Penulis (Haryanta et al., 2017). Jurnal Sisfotek Global ISSN : 2088 – 1762 Vol. 7 No. 1 / Maret 2017 dengan judul Perancangan Sistem Informasi Perencanaan Dan Pengendalian Bahan Baku Pada Home Industri. Prosedur sistem dengan menggunakan program UML (*Unified Modeling Language*). Kesimpulan Rancangan sistem informasi perencanaan dan pengendalian bahan baku yang akan dibuat penulis lebih mengutamakan pada penginputan data dalam perhitungan bahan baku sehingga menghasilkan perhitungan bahan baku yang lebih jelas dan akurat serta meminimalisir kesalahan pada proses perhitungan bahan baku.