

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Teori Umum

2.1.1. Sistem Informasi

Menurut Yakub (2012) dalam (Nugrahanti, 2015), sistem merupakan gabungan dari beberapa elemen yang saling terintegrasi dan memiliki maksud yang sama yaitu untuk tercapainya sebuah tujuan, terdiri dari material, uang, mesin, sumber daya manusia, dan informasi. Hal itu membuat sebuah tujuan tertentu berhasil tercapai yang telah ditentukan dengan pihak manajemen ataupun pemilik dengan adanya sumber daya yang saling bekerja sama. Sedangkan yang dimaksud dengan informasi, yaitu hasil pengolahan data yang didapatkan dari gabungan elemen sistem yang terbentuk menjadi sesuatu yang mudah dimengerti dan merupakan pengetahuan relevan yang dibutuhkan oleh manusia agar dapat menambah pemahaman yang ada pada fakta-fakta (Bustanur Rahmad & Setiady, 2014).

Sistem informasi dapat diartikan sebagai gabungan dari komponen-komponen yang memiliki hubungan antara satu sama lain yang memproses, mengumpulkan, menyimpan, dan mendistribusikan informasi agar dapat memberikan dukungan pada saat keputusan akan diambil dan pengawasan pada suatu organisasi (Laudon & Laudon, 2015: 16). Dalam definisi lainnya sistem informasi merupakan sebuah cara yang terorganisir dalam pengumpulan,

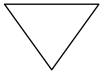
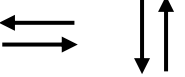
pemasukkan, dan pemrosesan data dan melakukan penyimpanan, pengelolaan, pengontrolan, dan pelaporannya sehingga dapat memberikan dukungan bagi perusahaan atau organisasi untuk tercapainya suatu tujuan (Tantra, 2012: 2). Sistem informasi juga merupakan gabungan pada informasi, orang, prosedur kerja dan teknologi informasi dalam memperoleh suatu tujuan (Heryanto et al., 2014). Sedangkan sistem informasi dibuat untuk menyederhanakan pengelolaan data dan informasi serta membantu dalam pencarian data dan informasi yang dibutuhkan (Fitriana, Moengin, & Riana, 2016).

Berdasarkan konsep sistem informasi dari berbagai sumber diatas, maka peneliti merumuskan sistem informasi adalah proses perubahan data menjadi informasi yang diproses secara komputerisasi dan disimpan pada database, di mana informasi yang dihasilkan berguna bagi manajemen perusahaan sebagai dasar pengambilan keputusan manajemen. Sistem informasi juga memiliki keunggulan menghindari terjadinya *human error*, membuat pekerjaan lebih cepat dan praktis dengan tingkat keakuratan yang tinggi. Selain kelebihan pada sistem informasi juga terdapat kelemahan, yaitu harus mengeluarkan investasi pada awal penerapan sistem tersebut seperti pada pembelian *hardware*, membeli sistem pada perusahaan yang menyediakan jasa pembuatan sistem atau merekrut *programmer freelance*, dan penambahan biaya *maintanance* ataupun penambahan fitur lain yang diperlukan.

2.1.2. Aliran Sistem Informasi

Zefriyenni dan Santoso (2015) dalam (Sukrianto, 2017), untuk mengetahui masalah-masalah yang terdapat pada suatu sistem maka dibutuhkan Aliran sistem informasi. Dari aliran sistem informasi tersebut dapat disimpulkan bahwa sistem informasi itu layak untuk dipergunakan atau sudah tidak layak lagi, masih secara manual atau sudah terkomputerisasi, apabila sudah tidak layak dipergunakan maka butuh dilakukan pergantian pada pengolahan data agar dapat memberikan hasil berupa informasi yang akurat, cepat dan juga membuat keputusan yang lebih bagus. Berikut simbol-simbol dari Aliran Sistem Informasi (ASI):

Tabel 2.1 Simbol-Simbol Pada Aliran Sistem Informasi (ASI)

No	Nama	Gambar	Keterangan
1	Proses komputerisasi		Untuk proses pengolahan data secara komputerisasi
2	Penghubung		Digunakan untuk menghubungkan sambungan aliran
3	Dokumen		Digunakan untuk operasi input
4	Arsip		Merupakan arsip data yang dihasilkan
5	Proses manual		Untuk proses pengolahan data secara manual
6	Aliran sistem		Untuk arah pengaliran data Proses
7	Basis data		Untuk media penyimpanan secara terkomputerisasi
8	Pita kertas		Untuk menampilkan input/output menggunakan pita kertas
9	Display		Untuk menampilkan output kelayar monitor

2.2. Tinjauan Teori Khusus

2.2.1. Inventory

Menurut Herjanto (2007) *inventory* merupakan barang/ bahan yang disimpan dan dipergunakan dalam memenuhi satu tujuan, seperti ketika akan digunakan pada proses perakitan atau produksi, dijual lagi, dan untuk suku cadang pada sebuah mesin atau peralatan (Mersiana & Purwandari, 2017). Dalam pengertian lainnya, *inventory* merupakan suatu rangkaian pengendalian dan kebijaksanaan yang mengawasi tingkat persediaan dan menetapkan tingkat pada persediaan yang harus diawasi, berapa persen jumlah pesanan yang harus dipesan dan kapan persediaan wajib diisi kembali (Rahmadi & Yusmiarti, 2016). *Inventory* juga memiliki arti sebagai sekumpulan barang yang disimpan agar dapat dijual/ digunakan pada periode atau masa mendatang (Fatmawati & Munajat, 2018).

Berdasarkan konsep *inventory* dari berbagai sumber diatas, maka dapat disimpulkan *inventory* adalah pengendalian persediaan stok barang yang digunakan untuk proses produksi, penjualan, dsb.

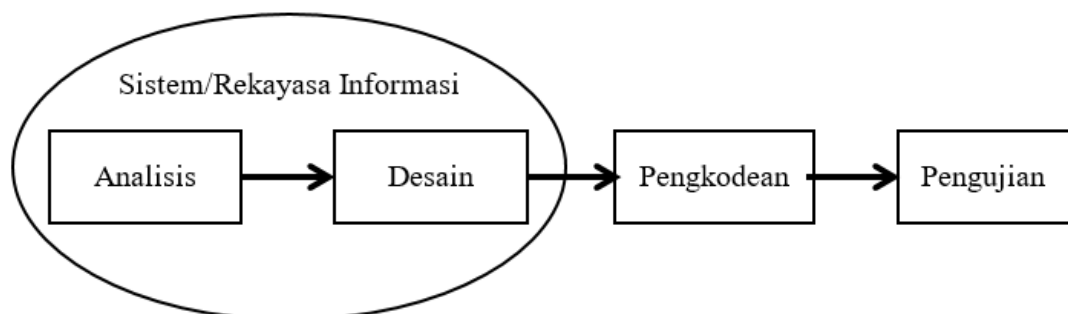
2.2.2. Sistem Informasi Inventory

Sistem informasi *inventory* barang adalah suatu sistem yang diciptakan untuk mengetahui jumlah barang yang terdapat di gudang (Pahlevi, Mulyani, & Khoir, 2018). Pengertian lain dari sistem informasi *inventory* lainnya, yaitu sebagai suatu sistem perangkat lunak yang berfungsi untuk melakukan proses inventarisasi seperti pencatatan data dari barang masuk sampai dengan barang keluar (Mersiana & Purwandari, 2017).

Berdasarkan konsep sistem informasi *inventory* dari berbagai sumber diatas, maka dapat diambil kesimpulan bahwa sistem informasi *inventory* merupakan sistem yang mempunyai fungsi agar dapat membantu proses *inventory* seperti memonitor jumlah stok barang dari proses masuknya barang hingga barang tersebut dikeluarkan.

2.2.3. SDLC Model *Waterfall*

Dalam A.S & Shalahuddin (2011: 24) SDLC (*System Development Live Cycle*) adalah proses untuk mengubah/ mengembangkan sebuah *system software* dengan mempergunakan model-model maupun metodologi yang dipergunakan manusia dalam mengembangkan beberapa *system software* sebelumnya (berdasarkan *best practice* datau berbagai cara yang telah diuji baik). Ada beberapa model pada SDLC salah satunya adalah model *waterfall* yang sering juga disebut *classic life cycle* (alur hidup klasik) atau model sekuensial linier (*sequential linear*), model tersebut juga memberikan ketersediaan pada pendekatan *life cycle* pada *software* secara berurutan atau sekuensial mulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*).



Sumber: A.S & Shalahuddin (2011: 27)

Gambar 2.1 Ilustrasi model *waterfall*

Penjelasan gambar:

1. Analisis kebutuhan perangkat lunak

Rangkaian tindakan dalam mengumpulkan kebutuhan yang dilakukan dengan intensif untuk menjelaskan keperluan *software* agar bisa dimengerti bagaimana *software* diperlukan oleh *user*.

2. Desain

Desain *software* merupakan rangkaian tindakan beberapa langkah yang berfokus kepada arsitektur perangkat lunak , desain *development* program *software* hingga representasi antarmuka, struktur data, dan prosedur pengkodean. Langkah ini mentranslasi keperluan *software* dari langkah analisis keperluan pada representasi desain supaya bisa diimplementasikan menjadi program untuk langkah berikutnya.

3. Pengkodean

Proses dimana desain wajib sudah ditranslasikan pada program *software*, yang nantinya akan menghasilkan program komputer sama seperti desain yang sudah dikerjakan dalam langkah desain.

4. Pengujian

Pengujian berfokus kepada *software* mulai dari dari segi *logic* hingga fungsionalnya dan memastikan apakah seluruh bagian telah di uji. Hal tersebut dilakukan agar dapat meminimalkan kesalahan (*error*) dan memastikan *output* yang dikeluarkan sudah seperti yang diinginkan.

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Pada langkah pemeliharaan/ pendukung bisa mengulang langkah proses pengembangan bermula dari analisis spesifikasi dalam perubahan *software* yang telah ada, namun tidak dalam pembuatan *software* baru.

2.2.4. *Unified Modeling Language (UML)*

Menurut Bunafit Nugroho (2010) dalam (Mersiana & Purwandari, 2017), *unified modeling language* merupakan pengolahan dalam mendesain perangkat lunak (*software*) sebelum melakukan pengkodean (*coding*). Sedangkan dalam A.S & Shalahuddin (2011: 113) UML merupakan suatu bahasa standar yang banyak dipakai pada dunia *industry* untuk membuat analisis dan desain, mendefinisikan *requirement*, dan juga mendeskripsikan arsitektur pada pemrograman *OOP (Object Oriented Programming)*.

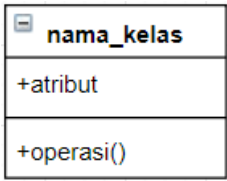
Berikut adalah beberapa diagram yang terdapat dalam UML:

- ***Class Diagram***

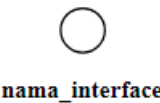
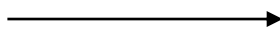
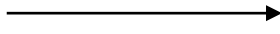
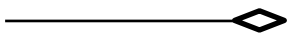
Diagram kelas atau *class diagram* mendeskripsikan struktur sistem dari sisi pengertian kelas-kelas yang nantinya dibuat pada pembangunan sistem.

Berikut adalah simbol-simbol yang terdapat dalam *class diagram*:

Tabel 2.2 *Class Diagram*

No	Simbol	Deskripsi
1	Kelas 	Kelas yang terdapat dalam struktur sistem

Tabel 2.2 Lanjutan

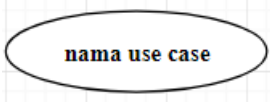
2	Antarmuka/ <i>interface</i> 	Seperti konsep dari <i>interface</i> pada <i>Object Oriented Programming</i> .
3	Asosiasi/ <i>association</i> 	Makna umum pada relasi antar kelas, asosiasi biasanya juga diikuti dengan <i>multiplicity</i>
4	Asosiasi berarah / <i>directed association</i> 	Makna kelas pada relasi antar kelas yang dipergunakan oleh kelas lainnya, asosiasi biasanya dapat diikuti dengan <i>multiplicity</i>
5	Generalisasi 	Makna generalisasi-spesialisasi (umum khusus) pada relasi antar kelas
6	Kebergantungan/ <i>dependency</i> 	Makna kebergantungan antar kelas pada relasi antar kelas
7	Agregasi/ <i>aggregation</i> 	Makna semua-bagian (<i>whole-part</i>) pada relasi antar kelas

- **Use Case Diagram**

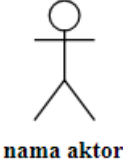
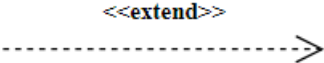
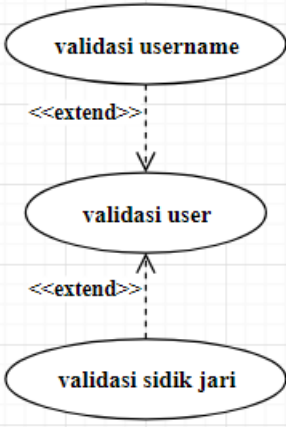
Use case atau diagram *use case* adalah pemodelan pada kelakuan (*behaviour*) sistem informasi yang akan dibuat, maupun mendeskripsikan suatu interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Berikut adalah simbol-simbol yang terdapat dalam diagram *use case*:

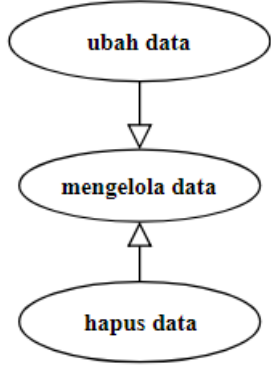
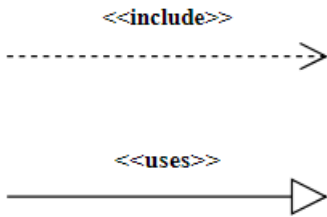
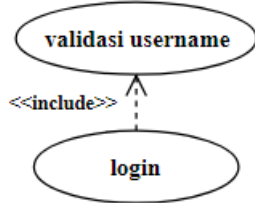
Tabel 2.3 *Use Case*

No	Simbol	Deskripsi
1	<i>Use case</i> 	Sebuah fungsi yang terdapat pada sistem sebagai aktor atau unit-unit yang saling bertukar pesan antar unit, biasanya dinyatakan dengan memakai kata kerja pada awal frase nama <i>use case</i>

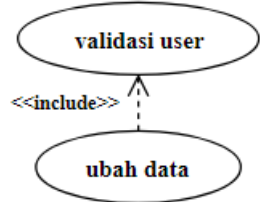
Tabel 2.3 Lanjutan

2	Aktor/ <i>actor</i>		Proses, manusia, atau sistem lain yang berkomunikasi dengan sistem informasi yang akan dibuat pada luar sistem informasi yang akan dibuat, meskipun simbol dari aktor merupakan gambar manusia, biasanya diwujudkan dengan kata benda pada awal frase nama aktor
3	Asosiasi/ <i>association</i>		Hubungan antara <i>use case</i> dengan aktor yang ikut kedalam <i>use case</i> atau <i>use case</i> yang mempunyai interaksi pada aktor
4	Ekstensi/ <i>extend</i>		Tambahan realasi <i>use case</i> pada suatu <i>use case</i> di mana <i>use case</i> yang ditambahkan bisa berdiri sendiri meski tidak ada <i>use case</i> tambahan, hal ini seperti prinsip <i>inheritance</i> pada <i>OOP</i>, biasanya <i>use case</i> tambahan mempunyai nama depan yang sama dengan <i>use case</i> yang ditambahkan, contohnya:  arah dari panah mengarah pada <i>use case</i> yang ditambahkan
5	Generalisasi/ <i>generalization</i>		Hubungan generalisasi dan spesialisasi (umum-khusus)

Tabel 2.3 Lanjutan

		antara dua buah use case di mana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:  arah panah mengarah terhadap <i>use case</i> yang menjadi generalisasinya (umum)
6	Menggunakan/ <i>include</i>/ <i>uses</i> 	Relasi <i>use case</i> tambahan pada suatu <i>use case</i> dimana <i>use case</i> yang sudah ditambahkan membutuhkan <i>use case</i> ini agar dapat menjalankan fungsinya/ sebagai syarat dijalankan <i>use case</i> ini ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>: • <i>Include</i> merupakan <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, contohnya pada <i>case</i> berikut:  • <i>Include</i> merupakan <i>use case</i> yang ditambahkan akan selalu menjalankan pemeriksaan

Tabel 2.3 Lanjutan

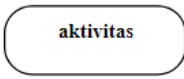
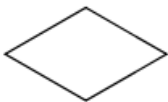
		apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, contoh pada <i>case</i> berikut:  <pre> graph BT A([ubah data]) -.-> <<include>> B([validasi user]) </pre> Kedua contoh diatas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang diperlukan.
--	--	--

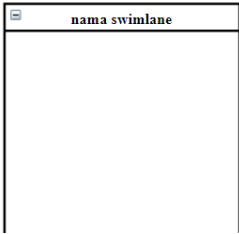
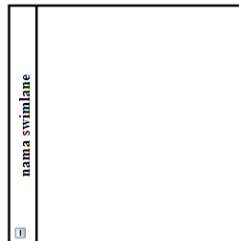
- **Activity Diagram**

Diagram aktivitas atau *activity diagram* menjelaskan aliran kerja (*workflow*) atau aktivitas pada suatu sistem/ proses bisnis.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.4 Activity Diagram

No	Simbol	Deskripsi
1	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3	Pecabangan/ decision 	Asosiasi pecabangan di mana jika ada pilihan aktivitas lebih dari satu
4	Penggabungan/ join 	Asosiasi penggabungan di mana lebih dari satu aktivitas digabungkan menjadi satu

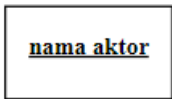
5	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6	Swimlane  atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

- **Sequence Diagram**

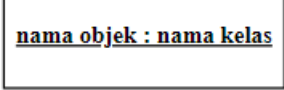
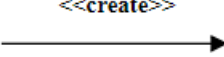
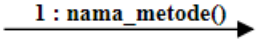
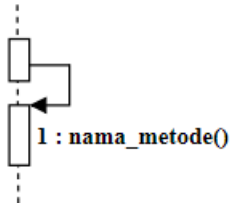
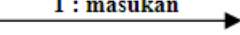
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek.

Berikut adalah simbol-simbol yang ada pada diagram sekuen:

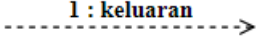
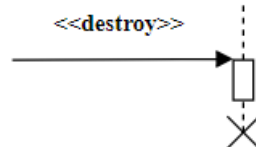
Tabel 2.5 *Sequence Diagram*

No	Simbol	Deskripsi
1	Aktor  nama aktor atau  tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor

Tabel 2.5 Lanjutan

2	Garis hidup/ <i>lifeline</i> 	Merupakan alur hidup suatu objek
3	Objek 	Merupakan objek yang berinteraksi pesan
4	Waktu aktif 	Merupakan keadaan aktif dan berinteraksi pesan pada objek
5	Pesan tipe <i>create</i> 	Merupakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6	Pesan tipe <i>call</i> 	Merupakan sebuah objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  Arah panah mengarah pada objek yang memiliki operasi/ metode, karena ini memanggil operasi/ metode, maka operasi/ metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
7	Pesan tipe <i>send</i> 	Menyatakan bahwa sebuah objek masukkan/ mengirimkan data/ informasi pada objek lainnya, arah panah mengarah pada objek yang dikirim

Tabel 2.5 Lanjutan

8	Pesan tipe <i>return</i> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9	Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

2.2.5. Database

Database merupakan sebuah dasar untuk memastikan kualitas dari informasi yang tepat waktu, akurat, dan relevan serta memberikan informasi (Fahrissal, Pohan, & Nasution, 2018). Dalam pengertian lainnya, database adalah gabungan data yang memiliki hubungan satu sama lain yang dikelompokkan berdasarkan struktur atau skema tertentu, sehingga bisa dimanfaatkan lagi dengan mudah dan cepat (Nugrahanti, 2015).

Fathansyah (2009) dalam (Bustanur Rahmad & Setiady, 2014) menyatakan bahwa penyusunan *database* dipakai agar dapat mengatasi permasalahan yang ada dalam data, diantaranya yaitu:

1. Inkonsistensi dan redudansi data

Kalau sekumpulan file dan program aplikasi dibuat oleh *software developer* yang tidak sama dalam waktu yang telah lewat cukup lama, maka memungkinkan terdapat bagian-bagian data yang mengalami redudansi pada file-file yang berbeda. Tersimpannya data yang terjadi berkali-kali dalam beberapa file juga bisa mengakibatkan inkonsistensi (tidak konsisten).

2. Kesulitan Pengaksesan

Dibutuhkan suatu data saat mencetak data-data, akan tetapi program yang sudah tertulis belum tersedia untuk menghasilkan data itu maka timbul kesulitan tersebut, dan solusi dari masalah tersebut akan mengarah pada Sistem Manajemen Basis Data yang dapat memperoleh data dengan cara langsung menggunakan bahasa yang mudah digunakan dan familiar.

3. Isolasi data untuk standarisasi

Apabila data terdistribusi pada file-file berbentuk format yang berbeda, maka hal tersebut dapat menjadi sulit dalam penulisan program aplikasi untuk mendapatkan dan melakukan penyimpanan data, oleh sebab itu data harus berada pada satu *database* yang dibuat satu format agar mudah dalam pembuatan program aplikasinya.

4. Masalah keamanan atau *Security*

Para pengguna sistem *database* sebagian diberikan izin agar dapat mengakses seluruh data, contohnya pada data yang berhubungan dengan gaji karyawan hanya dapat diakses bagian personalia dan keuangan. Keamanan tersebut bisa diatur dengan program yang dibuat oleh pemrogram atau fasilitas keamanan dari sistem operasi.

5. Masalah Integrasi (Kesatuan)

Database memiliki isi file yang berhubungan satu sama lain, yang menjadi masalah utama yaitu bagaimana hubungan antara file itu terjadi. Walaupun diketahui bahwa file A berhubungan dengan file B, tetapi secara teknis terdapat file kunci yang menghubungkan kedua file tersebut.

6. Masalah data *independence* (kebebasan data)

Aplikasi yang dibuat menggunakan bahasa yang diciptakan dari Sistem Manajemen *Database*, segala sesuatu yang terjadi dalam struktur file, ketika akan memeriksa data cukup dengan utility USE, menambahkan data hanya memerlukan APPEND, hal tersebut merupakan perintah-perintah yang terdapat pada paket Sistem Manajemen Basis Data bebas terhadap basis data. Perubahan apa saja pada *database*, seluruh perintah akan menjadi stabil tanpa memerlukan perubahan.

2.2.6. *Website*

Yuhefizar et al (2009) mengemukakan bahwa *website* merupakan keseluruhan dari sekumpulan halaman web yang ada pada suatu domain yang terdapat informasi, domain tersebut adalah nama *unique* yang dipunyai oleh suatu institusi agar dapat dihubungkan dengan internet. Teks yang digunakan sebagai media penghubung disebut *hypertext*, sedangkan relasi antara sebuah halaman web dengan halaman web yang berbeda disebut *hyperlink* (Mersiana & Purwandari, 2017).

2.2.7. HTML

Henderson (2009) dalam (Pahlevi et al., 2018) menyatakan *Hyper Text Markup Language* (HTML) adalah bahasa yang dipakai dalam mendeskripsikan struktur halaman web dengan fungsi untuk memublikasikan dokumen online.

2.2.8. PHP

PHP (*Hypertext Preprocessor*) merupakan suatu skrip bahasa pemrograman yang dirancang dengan tujuan agar dapat membangun aplikasi web dalam bentuk skrip yang diproses dan ditempatkan dalam server, dengan hasil yang akan dikirimkan pada klien, sebagai pengguna yang memakai browser yang secara khusus didesain untuk membentuk web dinamis (Bari & Kasmawi, 2016). Sedangkan Kurniawan (2010) dalam (Pahlevi et al., 2018) menyatakan pengertian lain dari, yaitu PHP merupakan bahasa pemrograman *web server-side* yang memiliki sifat *open source*, yaitu skrip yang menyatu dengan HTML yang terdapat dalam *server*.

2.2.9. CSS

CSS (*Cascading Style Sheet*) dalam Henderson (2009) adalah bahasa yang menampilkan tampilan pada halaman web seperti layout, warna, dan font, sehingga dengan memakai CSS seorang *programmer* bisa menciptakan halaman web yang dapat beradaptasi dengan bermacam-macam ukuran layar (Pahlevi et al., 2018).

2.2.10. Bootstrap

Spurlock (2013:1) menyatakan bahwa *bootstrap* merupakan suatu *framework* untuk CSS yang berbentuk produk *open source* yang diciptakan oleh Jacob Thornton dan Mark Otto yang pada mulanya dibuat untuk standarisasi bagian *front end* bagi seluruh *programmer* pada perusahaannya (Pahlevi et al., 2018).

2.2.11. *JavaScript*

JavaScript merupakan bahasa pemrograman web yang memiliki sifat *client side programming language* yaitu tipe bahasa pemrograman yang pengolahannya dilakukan oleh *client* yang bertujuan menuju pada *web browser* seperti Mozilla Firefox, Google Chrome, Opera Mini dan sebagainya (Pahlevi et al., 2018).

2.2.12. *JQuery*

Beighley (2010:8) menjelaskan *JQuery* merupakan *open source add-on* dari *library JavaScript* yang menekankan kepada interaksi antar HTML dan *JavaScript*. *JQuery* juga merupakan kode *JavaScript* yang sudah tertulis dan cukup menambahkan satu atau dua baris kode untuk memanggil *JQuery* (Pahlevi et al., 2018).

2.2.13. *Ajax*

Ajax engine merupakan suatu objek *JavaScript* atau *function* yang dipanggil pada saat informasi dibutuhkan dari server, dalam *Ajax* terdapat *layer* yang sebut juga *Ajax engine* agar dapat menangani interaksi. Proses yang dilakukan mesin *AJAX (AJAX Engine)* yaitu melakukan *parsing data*, menerima *respons* dari server, dan melakukan sedikit perubahan untuk kebutuhan dalam menampilkan informasi. Oleh sebab itu proses melakukan pengiriman sedikit informasi, jika dibandingkan pada model aplikasi web tradisional, maka tampilan *UI* dengan cepat terupdate dan pengguna dapat melakukan pekerjaannya dengan cepat (Fitriya et al., 2015).

2.2.14. *Dreamweaver*

Dreamweaver merupakan suatu HTML *editor* profesional dalam merancang web secara visual dan mengelola halaman web atau situs. Saat ini ada perangkat lunak dari perkumpulan Adobe yang sekarang ini banyak dipergunakan dalam merancang suatu situs web dengan kemampuan dalam menyunting koding maupun dalam pembuatan aplikasi web dengan memakai bahasa-bahasa pemrograman web, antara lain: HTML, JavaScript, CSS, PHP, dll (Kadir, Abdul, 2003) (Linda, 2016).

2.2.15. XAMPP

Anhar (2010) dalam (Mersiana & Purwandari, 2017) menyatakan XAMPP merupakan sebuah *software* yang memberi dukungan pada sistem-sistem operasi dan merupakan gabungan dari program-program, sebagai server berupa *localhost*, yang mencakup program *MySQL database*, *Apache HTTP Server*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman Perl dan PHP. Singkatan dari nama XAMPP, yaitu dari X (empat *operating system*), APACHE, MySQL, PHP dan Perl.

2.2.16. *MySQL*

MySQL merupakan satu dari berbagai jenis basis data *server* yang memakai SQL (*Structured Query Language*) menjadi bahasa dasar dalam pengaksesan basis datanya yang termasuk jenis *Relational Database Management System* (RDBMS), agar perumpamaan seperti tabel, baris dan kolom dipergunakan dalam MySQL. MySQL sangat terkenal pada lingkungan *software developer* dikarenakan MySQL adalah database server yang *open source* dan cepat. Bukan hanya itu, MySQL juga

mendapat dukungan dari komunitas maupun perusahaan yang memadai sehingga MySQL dapat menjadi basis data yang disenangi dan masuk dalam kategori *database* yang handal (Bari & Kasmawi, 2016).

2.3. Penelitian Terdahulu

Sebagai bahan pertimbangan dalam penelitian ini, maka penulis mencantumkan beberapa penelitian yang diambil dari beberapa jurnal ilmiah, yaitu:

Tabel 2.6 Penelitian Terdahulu

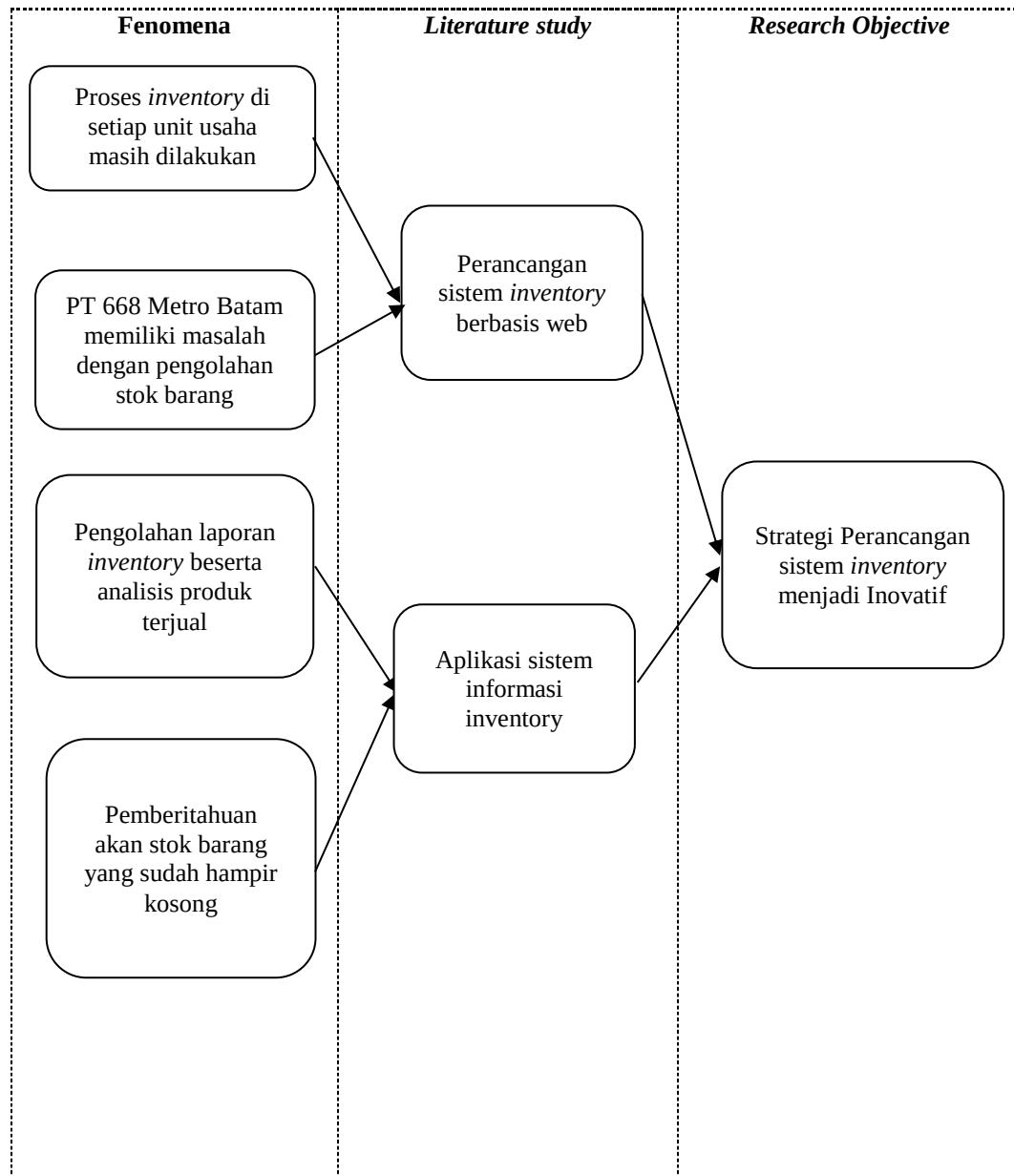
No.	Judul & Penulis	Variabel Penelitian	Jenis penelitian	Hasil
1	<i>Design of a Computerized Inventory Management System for Supermarkets</i> (Opeyimi A., Fatoba, & Blessing O., 2013)	<i>Inventory Management System</i>	Kualitatif	Sistem manajemen persediaan yang terkomputerisasi untuk memesan dan memperbarui persediaan yang dirancang dan untuk mengimplementasikan solusi yang mungkin pada barang yang dipasok.
2	<i>Information system design of inventory control spare parts maintenance (valuation class 5000) (case study: Plant kw)</i> (Fitriana et al., 2016)	<i>Information system design of inventory control</i>	Kualitatif	Mengembangkan sistem informasi yang dapat mendukung pemeliharaan dari pengendalian <i>inventory</i> suku cadang KW plant.
3	<i>Web based Multi Product Inventory Optimization using Genetic Algorithm</i> (P. & Iyakutti, 2011)	<i>Inventory Optimization</i>	Kualitatif	Mengoptimalkan tingkat produksi dan meminimalkan biaya penyimpanan <i>inventory</i>
4	Sistem Inventori Barang dengan Teknologi AJAX (Fitriya et al., 2015)	Sistem Inventori Barang	Kualitatif	Sistem inventori barang dengan menggunakan AJAX

Tabel 2.6 Lanjutan

5	Aplikasi Sistem <i>Inventory</i> Berbasis Web pada PT. Kreasinar Inticipa Nuansa (Linda, 2016)	Sistem <i>Inventory</i> Berbasis Web	Kualitatif	Sistem <i>inventory</i> berbasis web yang dapat menyajikan informasi yang dibutuhkan oleh pengguna
6	Sistem Informasi Pengolahan Data Persediaan Barang Menggunakan Metode Object Oriented di PT Livaza Teknologi Indonesia Jakarta (Pahlevi et al., 2018)	Sistem Informasi Pengolahan Data Persediaan Barang	Kualitatif	Aplikasi berbasis web untuk mengelola data <i>inventory</i> barang di PT Livaza Teknologi Indonesia Jakarta
7	Perancangan Sistem <i>Inventory</i> Barang pada UD. Minang Dewi Berbasis Website (Fahrisal et al., 2018)	Sistem <i>Inventory</i> Barang	Kualitatif	Penelitian ini dapat memperoleh informasi yang dibutuhkan setiap saat untuk melakukan kebijaksanaan dalam pengambilan keputusan secara cepat dan tepat.

2.4. Kerangka Pemikiran

Kerangka pemikiran dari penelitian ini adalah sebagai berikut:



Gambar 2.2 Kerangka pemikiran