

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Penelitian terkait bertujuan untuk mengambil beberapa referensi jurnal terkait yang digunakan dalam mendukung penelitian yaitu:

1. Penelitian yang dilakukan oleh Imam Solikin (2017) dengan judul **APLIKASI MENGHITUNG HARGA POKOK PERSEDIAAN MENGGUNAKAN METODE *AVERAGE***. Berdasarkan penelitian yang dilakukan, dijelaskan bahwa Penelitian ini telah menghasilkan sebuah Aplikasi dapat menghitung harga pokok persediaan menggunakan metode *Average*. Aplikasi dapat melakukan pencatatan transaksi pembelian persediaan barang dan mencetak laporan transaksi pembelian persediaan barang.
2. Peneliti yang dilakukan oleh Amiennia Rektiani (2015) dengan judul **APLIKASI PENCATATAN PERSEDIAAN, PENJUALAN, DAN PESANAN DENGAN METODE PERPETUAL**. Berdasarkan penelitian yang dilakukan, dijelaskan bahwa Aplikasi dapat melakukan pencatatan transaksi penjualan dan pesanan. Aplikasi dapat menghitung dan menampilkan jumlah stok barang melalui kartu persediaan dengan metode *Average* secara otomatis dan dapat menampilkan laporan berupa jurnal dan buku besar secara otomatis.
3. Penelitian yang dilakukan oleh Julianto Simatupang (2017) dengan judul **PERANCANGAN SISTEM INVENTORI BARANG PADA TOKO NICHOLAS JAYA MENGGUNAKAN METODE FIFO**. Berdasarkan

penelitian yang dilakukan bahwa Sistem inventori pada Toko Nicholas Jaya dengan menggunakan komputer merupakan suatu pilihan yang tepat apabila dibandingkan dengan cara manual. Jika dibandingkan dengan sistem manual maka dengan memakai sistem komputer mempunyai kelebihan sebagai berikut:

- a. Dapat menghasilkan informasi data persediaan barang dengan tepat dan cepat.
- b. Dapat menghasilkan informasi yang tepat waktu sesuai dengan permintaan yang diinginkan.

II.2. Uraian Teoritis

Penelitian yang dilakukan dengan cara langsung pada PT. Karet Deli yaitu :

II.2.1. Aplikasi

Menurut Imam Solikin (2017) Aplikasi merupakan penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi *output*. Aplikasi merupakan penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.

II.2.2. Akuntansi

Menurut Amiennia Rektiani (2015) Menurut *American Institute of Certified Public Accountants* (AICPA), akuntansi adalah suatu kegiatan jasa. Fungsinya adalah menyediakan data kuantitatif, terutama yang mempunyai sifat keuangan, dari kesatuan usaha ekonomi yang dapat digunakan dalam pengambilan keputusan-keputusan ekonomi dalam memilih alternatif dari suatu keadaan.

II.2.3. Prediksi Penjualan

Menurut Ryan Naufal, Dkk (2017) Prediksi Penjualan adalah pemikiran terhadap suatu besaran penjualan, misalnya permintaan terhadap satu atau beberapa produk penjualan pada periode yang akan datang. Pada hakekatnya prediksi hanya merupakan suatu perkiraan (*guess*), tetapi dengan menggunakan teknikteknik tertentu, maka prediksi menjadi lebih dari sekedar perkiraan. Prediksi dapat diartikan perkiraan yang ilmiah (*educated guess*).

II.2.4. Metode *Moving Average*

Menurut Imam Solikin (2017) Pada metode rata-rata penentuan harga perolehan persediaan tidak didasarkan pada harga persediaan yang pertama masuk atau yang terakhir masuk melainkan diantara keduanya. Harga perolehan persediaan akhir dan harga pokok penjualan dihitung berdasarkan harga rata-rata persediaan yang siap dijual. Metode rata-rata mengasumsikan barang yang ada untuk dijual memiliki harga yang sama untuk setiap unit-nya. Dalam metode ini,

persediaan barang terdiri dari dua transaksi atau lebih dengan harga yang berbeda maka digunakan persamaan untuk menghitung harga perolehan rata-rata per unit.

Rumus harga perolehan rata-rata per unit :

$$\text{Harga perolehan rata-rata} = \frac{\text{Harga perolehan persediaan siap jual}}{\text{Jumlah persediaan siap dijual}}$$

Harga perolehan rata-rata per unit dimana harga tersebut dipengaruhi oleh jumlah barang yang diperoleh pada masing-masing harga.

II.2.5. PHP (*Hypertext Preprocessor*)

Menurut Yusi Ardi Binarso (2014 : 76) PHP (*Hypertext Preprocessor*) merupakan bahasa *scripting* yang tergabung menjadi satu dengan *HTML* dan dijalankan pada *server side* atau semua perintah yang diberikan akan secara penuh dijalankan pada *server*, sedangkan yang dikirimkan ke klien (*browser*) hanya berupa hasilnya saja.

Sintak dan semantik *PHP* memiliki kesamaan dengan bahasa *C*, *JAVA*, *PERL*, dengan menambahkan tambahan tag `<?php` sebagai pembuka dan diakhiri dengan `?>`, pasangan kedua kode inilah yang berfungsi sebagai tag kode *PHP*. Berdasarkan tag inilah pihak *server* dapat memahami kode *PHP* dan kemudian memprosesnya, hasilnya dikirim ke *browser*.

II.2.6. Pengertian XAMPP

Menurut Yusi Ardi Binarso (2014) XAMPP adalah sebuah software web server Apache yang didalamnya sudah tersedia database server MySQL dan mendukung PHP programming. XAMPP merupakan singkatan dari X(untuk empat sistem operasi), Apache, MySQL, PHP.

II.2.7. Pengertian Database

Menurut Sarifudin (2014 : 3) *Database* (basis data) adalah kumpulan *file-file* yang mempunyai kaitan antara satu *file* dengan *file* yang lain sehingga membentuk data untuk menginformasikan satu perusahaan dan instansi. Bila terdapat *file* yang tidak dapat dipadukan atau dihubungkan dengan *file* yang lainnya, berarti *file* tersebut bukanlah kelompok dari satu database, melainkan membentuk satu database sendiri.

Database juga merupakan landasan bagi pembuatan dan pengembangan program aplikasi. Oleh sebab itu, database harus dibuat sedemikian rupa sehingga pembuatan program lebih mudah dan cepat. Database merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi bagi para pemakai. Penerapan database dalam sistem informasi disebut dengan Sistem data-base. Sistem database adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya. Database terdiri dari data yang akan digunakan atau diperuntukkan terhadap banyak *user*, dari masing-masing *user* akan menggunakan data tersebut sesuai dengan tugas dan fungsinya.

II.2.8. *MySQL*

Menurut Sarifudin (2014 : 2) *MySQL* adalah sebuah sistem manajemen database. Database adalah sekumpulan data yang terstruktur. Data-data itu dapat suatu daftar belanja yang sangat sederhana sampai ke galeri lukisan atau banyaknya jumlah informasi pada jaringan perusahaan. Untuk menambah, mengakses dan memproses data yang tersimpan pada database komputer, kita membutuhkan manajemen database seperti *MySQL*. *MySQL* adalah sebuah sistem manajemen database yang saling berhubungan. Sebuah hubungan database dari data yang tersimpan pada tabel yang terpisah daripada menyimpan semua data pada ruang yang sangat besar. Hal ini menambah kecepatan dan fleksibilitas. Tabel-tabel tersebut dihubungkan oleh hubungan yang sudah didefinisikan mengakibatkan akan memungkinkan untuk mengkombinasikan data dari beberapa tabel sesuai dengan keperluan. *MySQL* adalah ‘Open Source Software’. ‘*Open Source*’ maksudnya program tersebut memungkinkan untuk dipakai dan dimodifikasi oleh siapa saja. Semua orang bisa mendownload *MySQL* dari Internet dan memakainya tanpa membayar sepeser pun. Seseorang dapat mempelajari ‘*Source Code*’ dan dapat mengubahnya sesuai dengan kebutuhan mereka. *MySQL* menggunakan GPL (GNU General Public License). *MySQL* adalah suatu perangkat lunak database relasi (Relational Database Management System atau RDBMS), seperti halnya *ORACLE*, *Postgresql*, *MS SQL*, dan sebagainya. Berdasarkan riset dinyatakan bahwa di platform web, dan baik untuk kategori open source maupun umum, *MySQL* adalah database yang paling banyak dipakai. Menurut perusahaan pengembangnya, *MySQL* telah terpasang di sekitar

3 juta komputer. Puluhan hingga ratusan ribu situs mengandalkan *MySQL* bekerja siang malam memompa data bagi para pengunjungnya.

II.2.9. UML (*Unified Modelling Language*)

Menurut (Windu Gatta, 2017 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

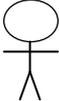
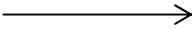
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *Use case* Diagram

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 II.1. Simbol *Use Case*

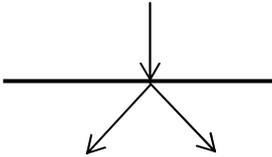
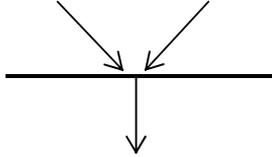
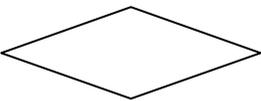
Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang 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 mengidkasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<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 : Windu Gatta : 4-6)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu:

Tabel II.2. Simbol *Activity Diagram*

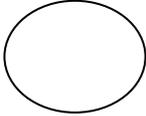
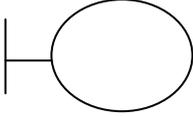
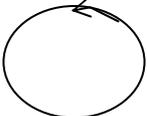
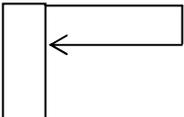
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, 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</i> diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gatta : 6)

3. Diagram Urutan (*Sequence Diagram*)

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 II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>EntityClass</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 formentry 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> , 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 : Windu Gatta : 7)

4. *Class Diagram* (Diagram Kelas)

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.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

Class diagram secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.4. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Windu Gatta : 8-9)