



BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Beberapa penelitian yang relevan diantaranya adalah penelitian yang telah dilakukan oleh Rian Hardi Ramdani dengan judul “Rancang Bangun Sistem Informasi Penjualan Furniture Berbasis Web”. Sistem dibangun dengan menggunakan metode pengembangan *System Develompment Life Cycle*. Sistem ini dibuat menggunakan bahasa pemrograman PHP dan penyimpanan *database* menggunakan MySQL. Tujuan pembuatan sistem ini adalah membantu CV. Morodadi Furniture dalam mengelola data master, pemesanan barang, dan transaksi penjualan secara online sehingga dapat memudahkan konsumen maupun pemilik mendapat informasi secara *realtime*. Hasil yang didapat dari penelitian yang sudah dilakukan adalah terciptanya sistem informasi penjualan furnitur berbasis website[4].

Penelitian lain terkait dengan sistem informasi pemesanan dilakukan oleh Eko Nurcahyo dengan judul “Sistem Informasi Penjualan Mebel Furniture Berbasis Website”. Dalam penelitian tersebut penulis membuat sistem yang sebelumnya berjalan dengan cara penawaran langsung kepada konsumen dan juga mengikuti pameran-pameran mebel maupun interior diubah menjadi penjualan secara online atau *e-commerce*. Dalam rancangan sistem ini penulis menggunakan model perancangan berbasis objek DAD (Data Alir Diagram), ERD (*Entity Realtionship Diagram*) untuk mengatur Struktur Basis Data, Struktur Tabel, dan Relasi antar Tabel. Tujuan pembuatan sistem ini adalah memperluas penyebaran wilayah penjualan dan mempermudah akses pengelolaan data barang, laporan penjualan barang, pemesanan barang, dan laporan ketersediaan barang sehingga tidak terjadi kekeliruan dalam mengakses data[5].

Penelitian lain yang dilakukan oleh Elisa Nendes dan Dian Anubhakti dengan judul “Analisa dan Desain Sistem Informasi Penjualan Furniture Berbasis *e-commerce* Studi Kasus PT. Anya Living”. Permasalahan yang ada pada Anya Living saat ini yaitu Anya Living kesulitan mengetahui barang yang dikembalikan oleh pelanggan, hal ini disebabkan karena data retur masih menjadi satu dengan dokumen transaksi dan dokumen pembayaran. Dari permasalahan yang ada, penulis melakukan pengumpulan data dengan hasil berupa pembuatan Sistem

Informasi Berbasis *E-Commerce*. Sistem ini dibuat menggunakan bahasa pemrograman PHP dan penyimpanan *database* menggunakan MySQL. Tujuan pembuatan sistem memudahkan pelanggan dalam bertransaksi dan memudahkan perusahaan dalam mengelola data penjualan barang[6].

Penelitian yang dilakukan oleh penulis memiliki beberapa perbedaan diantaranya yaitu sistem dilengkapi dengan fitur diskusi untuk membuat pemesanan. Sistem juga dilengkapi dengan telegram yang digunakan untuk menginformasikan progress pemesanan. Sistem dibangun menggunakan tahapan-tahapan pengumpulan data yaitu meneliti berbagai literatur, wawancara, dan observasi. Metode yang digunakan dalam pengembangan sistem informasi pemesanan yaitu metode *prototype*. Sistem dibuat menggunakan bahasa pemrograman PHP dan MySQL sebagai *database*.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Menurut Kadir (2018), sistem informasi merupakan beberapa komponen yang melakukan proses data menjadi informasi yang dilakukan untuk mencapai suatu sasaran atau tujuan. Komponen yang dimaksud yaitu manusia, komputer, teknologi informasi, dan prosedur kerja. Sedangkan menurut Kristanto (2018) menyimpulkan bahwa sebuah sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang akan mengolah data menggunakan perangkat keras dan perangkat lunak tersebut[7].

2.2.2 Pemesanan

Menurut Susanti dan Prabowo (2014), Pemesanan adalah aktivitas harus dilakukan konsumen sebelum proses membeli. Pemesanan juga merupakan proses, pembuatan, cara pemesanan kepada orang lain. Hal ini biasanya dilakukan pada saat transaksi jual beli. Langkah-langkah pemesanan adalah melakukan kontak langsung dengan penjual dan konsumen akan memesan barang yang ingin dibeli. Setelah barang yang dipesan ada, selanjutnya konsumen melakukan pembayaran. Pesanan yang dilakukan saat ini tidak hanya barang sudah diterima, tetapi konsumen masih dapat mememesannya. Perintah ini dapat dilakukan dengan berbagai cara, baik secara lisan maupun di dunia maya[8].

2.2.3 *Kitchen set*

Kitchen set merupakan sebuah bentuk dapur yang di desain secara multifungsi sebagai tempat penyimpanan alat-alat dapur. *Kitchen set* dikatakan multifungsi karena memiliki fasilitas penyimpanan, fasilitas pencucian, dan fasilitas kerja. Fasilitas penyimpanan contohnya kabinet, fasilitas pencucian contohnya bak cuci piring dan yang termasuk fasilitas kerja yaitu kompor. Bagian-bagian dari *kitchen set* yaitu kabinet atas, kabinet bawah, rak terbuka, lemari kabinet laci, lemari sudut, kitchen sink, dan kompor[2].

2.2.4 **Rekayasa Web**

Menurut Janer Simarmata rekayasa web adalah subdisiplin dari rekayasa perangkat lunak yang membantu menyediakan metodologi untuk merancang, mengembangkan, memelihara dan melibatkan aplikasi web. Tujuan utama dari rekayasa web adalah kesuksesan dalam mengatur kompleksitas, serta keanekaragaman pengembangan aplikasi web. Sedangkan menurut Pressman rekayasa web merupakan bagian dari software engineering yang dikhususkan dalam pembuatan suatu sistem aplikasi berbasis web berkualitas tinggi[9].

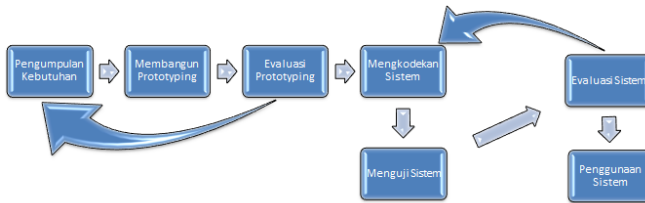
2.2.5 **Rekayasa Perangkat Lunak**

Rekayasa Perangkat Lunak (RPL) adalah suatu disiplin ilmu yang mengulas semua aspek produksi perangkat lunak, dimulai dari tahap awal yaitu communication, requirements capturing (analisa kebutuhan pengguna), specification (menentukan spesifikasi dari kebutuhan pengguna), desain, coding, testing hingga maintenance (pemeliharaan sistem) setelah digunakan. Tujuan dari RPL adalah memiliki banyak kesamaan dengan disiplin ilmu rekayasa yang lain. Ilmu rekayasa perangkat lunak akan selalu berusaha menghasilkan keluaran (output) yang memiliki performa tinggi serta waktu penyelesaian singkat dan hemat[10].

A. Metode Pengembangan Sistem

Sistem yang akan dibuat menggunakan metode Prototype. Menurut Ogedebe, dkk (2012) *prototyping* merupakan metode pengembangan perangkat lunak yang berupa model fisik kerja sistem dan berfungsi sebagai versi awal dari sistem. Dengan metode prototyping akan dihasilkan sistem antara pengembang dan pengguna agar dapat berinteraksi dalam proses kegiatan pengembangan sistem informasi. Prototype akan dihilangkan atau ditambahkan pada bagiannya sehingga

sesuai dengan perencanaan dan analisis yang dilakukan oleh pengembang sampai dengan uji coba dilakukan secara simultan seiring dengan proses pengembangan[11].



Gambar 2. 1 Tahapan Prototype

Berikut merupakan tahap-tahap metode *prototype*[3] :

- a. Tahap Pengumpulan Kebutuhan
Pengembang dan pelanggan akan mengidentifikasi masalah dan kebutuhan dari sistem yang akan dibangun.
- b. Tahap Membangun *Prototyping*
Dilakukan dengan cara menjelaskan keinginan pelanggan kepada pengembang untuk pembuatan perancangan sementara sistem meliputi fitur-fitur menu dan tampilan output maupun input
- c. Tahap Evaluasi *Prototyping*
Pelanggan melakukan proses pengecekan terhadap *prototype* yang sudah dibangun. Tujuannya mengetahui apakah sistem yang dibangun sesuai dengan kebutuhan atau belum. Jika belum maka akan dilakukan pengecekan dan perbaikan kembali. Jika sudah, maka bisa menambahkan fitur baru jika perlu dan menyempurnakan tampilan output maupun input.
- d. Tahap Mengkodekan Sistem
Apabila *prototype* pada tahap sebelumnya sudah disetujui oleh pelanggan maka akan dilakukan proses koding yaitu menerjemahkan kedalam bahasa pemrograman
- e. Tahap Pengujian Sistem
Jika sudah menjadi sebuah perangkat lunak maka akan dilakukan pengujian apakah perangkat lunak tersebut layak digunakan atau belum. Pengujian bisa dilakukan melalui *Black Box*, *White Box*,

Pengujian arsitektur, maupun Basis path dengan tujuan meminimalisir kesalahan yang ada.

f. Tahap Evaluasi Sistem

Dilakukan evaluasi sistem oleh pelanggan untuk memastikan apakah sistem yang dibangun sesuai dengan keinginan atau belum. Jika sudah sesuai dengan keinginan maka sistem sudah dapat digunakan. Namun apabila belum sesuai keinginan maka pengembang harus memperbaiki ketidaksesuaian ke tahap sebelumnya.

g. Tahap Penggunaan Sistem

Jika sistem yang dibangun berhasil melewati tahap evaluasi dengan baik maka sistem tersebut sudah dapat digunakan.

B. Tools / Alat Bantu

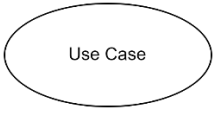
1. UML (*Unified Modelling Language*)



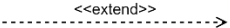
UML (*Unified Modeling Language*) adalah salah satu standarisasi bahasa pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, mengembangkan dan dokumentasi dari sistem perangkat lunak [12].

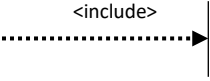
a. *Use Case Diagram*

Use case atau diagram *use case* merupakan pemodelan untuk kegiatan atau *behavior* sistem informasi yang akan dibuat. *Use case* menjelaskan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada *use case diagram* [12].

Tabel 2. 1 Simbol-Simbol *Use Case*

No.	Simbol	Nama	Keterangan
1		<i>Use Case</i>	Unit-unit yang saling berinteraksi antara sistem atau aktor; biasanya

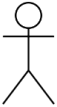
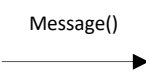

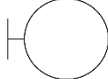

			dinyatakan dengan menggunakan kata kerja.
2 .		Aktor / <i>Actor</i>	Menunjukkan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri
3 .		Asosiasi / <i>association</i>	Menunjukkan komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4 .		Ekstensi / <i>extend</i>	Menunjukkan relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan tersebut


6		<i>include / uses</i>	<i>Menspesifikasikan bahwa perilaku use case merupakan bagian dari use case lain</i>
---	---	-----------------------	--

b. *Sequence Diagram*

Sequence diagram menggambarkan perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Berikut adalah symbol-symbol yang ada pada *sequence diagram*[12] :

Tabel 2. 2 Simbol-Symbol *Sequence Diagram*

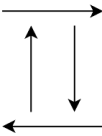

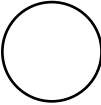
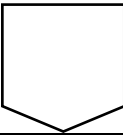
No	Simbol	Nama	Keterangan
1.	 Actor	Aktor / <i>actor</i>	Menggambarkan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri
2.		<i>A message</i>	Menggambarkan pengiriman pesan
3.		<i>A focus of control & A life line</i>	Menggambarkan temoat mulai dan berakhirnya message
4.		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dalam form
5.		<i>Control Class</i>	Menghubungkan <i>boundary</i> dengan tabel.

6.		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
----	---	---------------------	--

c. *Flowchart*


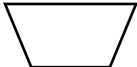
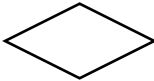
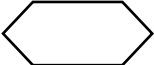
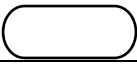



Flowchart adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma. Berikut adalah simbol-simbol yang ada pada *flowchart* [13] :

Tabel 2. 3 *Flowchart Direction Symbols*


No	Simbol	Nama	Keterangan
1		<i>Arus atau flow</i>	Menyatakan jalan arus suatu proses
2		<i>Communication Link</i>	Menyatakan transisi suatu data dari satu lokasi ke lokasi lainnya
3		<i>Connector</i>	Menyatakan sambungan dari suatu proses ke proses lainnya dalam lembar yang sama.
4		<i>Off Line Connector</i>	Menyatakan simbol masuk dan keluarnya suatu prosedur dalam lembar/halaman yang berbeda


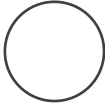



Tabel 2. 4 *Flowchart Processing symbols*

No	Simbol	Nama	Keterangan
1.		<i>Proses</i>	Menggambarkan setiap pengolahan yang

			dilakukan oleh komputer.
2.		<i>Manual</i>	Menyatakan suatu tindakan yang tidak dilakukan oleh computer(manual).
3.		<i>Decision (Logika)</i>	Menyatakan suatu kondisi yang akan menghasilkan dua kemungkinan jawaban.
4.		<i>Predefine Proses</i>	Menyediakan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
5.		<i>Terminal</i>	Menyatakan awal atau akhir suatu program
6.		<i>Keying Operation</i>	Menyatakan segala jenis operasi yang diproses menggunakan suatu mesin yang mempunyai keyboard.
7.		<i>Off Line Storage</i>	Menyatakan data dalam simbol akan disimpan ke suatu media tertentu.
8.		<i>Manual Input</i>	Memasukan data secara manual dengan menggunakan online keyboard

Tabel 2. 5 Flowchart Input-Output Symbols

No	Simbol	Nama	Keterangan
1.		<i>Input-output</i>	Menyatakan proses input dan output tanpa tergantung

			dengan jenis peralatanya.
2.		<i>Punched Card</i>	Menyatakan input berasal dari kartu atau output ditulis ke kartu.
3.		<i>Magnetic-Tape Unit</i>	Menyatakan input berasal dari pita magnetic atau output disimpan ke pita magnetic
4.		<i>Disk Storage</i>	Menyatakan input berasal dari disk atau output disimpan ke disk.
5.		<i>Document</i>	Mencetak laporan ke printer
6.		<i>Display</i>	Menyatakan peralatan <i>output</i> yang digunakan berupa layar (video, computer)

C. Metode Pengujian Sistem

Pengujian *Black-box Testing* adalah pengujian yang dilakukan dan hanya fokus pada fungsionalitas dari spesifikasi *software*. Pendefinisian kondisi *input* dan pengetesan nantinya akan dilakukan oleh *tester*[14]. Ujicoba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya:

- 1) Fungsi-fungsi yang salah atau hilang
- 2) Kesalahan *interface*
- 3) Kesalahan dalam struktur data atau akses *database* eksternal
- 4) Kesalahan performa
kesalahan inialisasi dan terminasi

2.2.6 Pemrograman Berorientasi Objek (PBO)

Pemrograman berorientasi objek adalah suatu metode pembangunan perangkat lunak yang menyusun perangkat lunak sebagai kumpulan objek yang berisi data dan operasi secara sistematis. Pemrograman berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas seperti rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek dan pengujian berorientasi objek[12]. Dalam Pemrograman Berorientasi Objek terdapat beberapa konsep dasar, yaitu[15]:

1. *Attribute*

Atribut dari sebuah kelas adalah variable global yang dimiliki sebuah kelas, atribut dapat memiliki hak akses private, public maupun protected.

2. *Method*

Method adalah fungsi atau prosedur yang dibuat oleh seseorang programmer didalam suatu *Class*. Sebuah *class* boleh memiliki lebih dari satu *method* dengan nama yang sama asalkan memiliki parameter masukan yang berbeda sehingga kompilier atau interpreter dapat mengenali *method* mana yang dipanggil.

3. *Class*

Didefinisikan sebagai sebuah denah yang mendefinisikan variable – variable dan metode – metode yang umum untuk semua objek dari jumlah jenis tertentu. Sebuah kelas menyerupai struktur yang merupakan tipe data sendiri, misalkan tipe data titik yang terdiri dari koordinat x dan y.

4. *Object*

Merupakan perwujudan dari *class*, setiap *object* akan mempunyai attribute dan *method* yang dimiliki oleh *class* nya. *Object* merupakan sesuatu yang memiliki identitas (nama), pada umumnya juga memiliki data tentang dirinya maupun *object* lain dengan mempunyai kemampuan untuk melakukan sesuatu dan bisa bekerja sama dengan objek lainnya.

5. *Encapsulation*

Proses pemaketan data objek bersama metode – metode lainnya. Manfaatnya adalah menyembunyikan rincian – rincian implementasi dari objek lain. Bagian internal objek mempunyai tampilan terbatas disbanding bagian eksternal objek. Bagian

eksternal objek sering disebut antarmuka objek karena berlaku sebagai antarmuka terhadap objek lain.

6. *Inheritance*

Merupakan proses penciptaan kelas baru dengan mewarisi karakteristik kelas yang telah ada, ditambah karakteristik unit kelas baru. Dalam terminologi java, kelas yang diwarisi disebut superklas, sedang kelas yang mewarisi disebut subklas. Java hanya memungkinkan pewarisan tunggal 22 (*single inheritance*). Pewarisan di Java dapat dilakukan terhadap bentukan *class* dan *interface*.

2.2.7 Basis Data

Database atau basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa file teks ataupun *Database Management System* (DBMS). Kebutuhan basis data dalam sistem informasi meliputi memasukan, menyimpan dan mengambil data serta membuat laporan berdasarkan data yang telah disimpan[12].

A. *Database Management System (DBMS)*





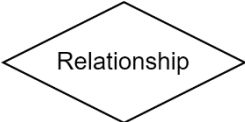
Database Management System (DBMS) atau dalam bahasa Indonesia sering disebut sebagai sistem manajemen basis data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola dan menampilkan data. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal seperti menyediakan fasilitas untuk mengelola akses data, mampu menangani integritas data, mampu menangani akses data dan mampu menangani *backup* data[12].

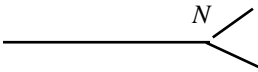
B. *Entity Relationship Diagram (ERD)*

Permodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk permodelan basis data relasional. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot, dan beberapa notasi lain. Namun yang banyak digunakan

adalah notasi dari Chen. Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen[12]:

Tabel 2. 6 Simbol-Simbol ERD

No.	Simbol	Nama	Keterangan
1.		Entitas /entity	Data inti yang akan disimpan atau benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi computer, penamaan entitas biasanya lebih ke kata benda.
2.		Atribut	Karakteristik dari tiap entitas maupun tiap relationship
3.		Atribut kunci primer	Digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id
4.		Atribut multinilai / <i>multivalue</i>	Kolom data yang disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
5.		Relasi	Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.

6.		Asosiasi / <i>association</i>	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian
----	---	----------------------------------	---

2.2.8 Telegram

Telegram merupakan salah satu aplikasi yang mengirimkan pesan secara instan yang memiliki kelebihan dari fitur-fitur yang dimilikinya. Sebagai aplikasi pesan singkat yang *realtime*, *Telegram* memberikan kemudahan akses bagi pengguna karena tersedia pada platform *mobile* maupun desktop. Pada platform *mobile Telegram* dapat digunakan di platform *iphone*, *android* dan *windows phone*, sedangkan pada platform desktop *Telegram* dapat digunakan di *Windows*, *Linux*, *Mac OS* dan juga *Web browser*. *Telegram* mengklaim sebagai aplikasi pesan massal tercepat dan teraman yang berada di pasar[16].

2.2.9 Framework Codeigniter

Codeigniter adalah sebuah framework PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Framework ini dilengkapi banyak *library* dan *helper* yang berguna di dalamnya dan tentunya mempermudah proses *development*. *Codeigniter* merupakan aplikasi *open source* berupa *framework* PHP dengan model MVC (*Model*, *View*, *Controller*) untuk membangun aplikasi web dinamis dengan cepat dan mudah. *Codeigniter* ini memungkinkan para pengembang untuk menggunakan *framework* secara parsial atau secara keseluruhan. Artinya bahwa *Codeigniter* masih memberi kebebasan kepada para pengembang untuk menulis bagian-bagian kode tertentu di dalam aplikasi menggunakan cara konvensional atau dengan *syntax* umum didalam PHP, tidak harus menggunakan aturan penulisan kode di *Codeigniter*[17].