

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Kajian penelitian sebelumnya dilakukan oleh Muhammad Maulana Sabrin, Muhammad Amin, Muhammad Hasbi dengan judul “Sistem Informasi Penjualan Aspal Curah Pada PT. Bumi Sarana Utama Kalla Aspal Berbasis Web” Dalam bahasa pemrograman yang digunakan adalah php, database yang di gunakan yaitu MySQL dan metode pengembangan perangkat lunak menggunakan metode *waterfall*, dengan adanya aplikasi ini di harapkan mempermudah kinerja dalam pendataan dari arsip penjualan tersebut [3].

Penelitian lainnya telah dilakukan oleh Yeni Angraini, Donaya Pasha, Damayanti, Aan Setiawan dengan judul “Sistem Informasi Penjualan Sepeda Berbasis Web Menggunakan Framework Codeigniter (Studi Kasus: Orbit Station)” Dalam sistem tersebut, konsumen dapat melihat sepeda dan stok sepeda yang ingin dibeli, admin dapat menginput data barang dan stok barang, konsumen dapat memesan produk. Dalam membangun sistem ini menggunakan *framework Codeigniter* [1].

Penelitian lain juga dilakukan oleh Andi Saputra, Ashari Imamuddin, Pria Sukamto dengan judul “Rancang Bangun Aplikasi Sistem Penjualan Case Study: PT. X Application of Design and Building Sales System” Dalam Sistem ini terdapat surat penawaran harga yang dapat diajukan oleh *costumer* serta dalam sistem ini terdapat pencatatan pesanan dan juga terdapat surat penagihan. Sistem ini berbasis *desktop* [4].

Sistem yang akan dikembangkan oleh penulis tentu saja akan berbeda dengan yang sudah ada sebelumnya. Dibandingkan dengan penelitian sebelumnya, sistem yang akan dikembangkan dalam penelitian ini akan menggunakan *framework laravel* dengan menggunakan bahasa pemrograman php serta menggunakan *database* MySQL. Sistem ini akan berbasis *website*, agar dapat diakses kapan saja.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategis dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan [5].

2.2.2 Website

Website atau situs juga dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing - masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*) [6].

2.2.3 MySQL

Menurut Rulianto Kurniawan, MySQL merupakan suatu jenis database server yang sangat terkenal. MySQL termasuk jenis RDBMS (*Relational Database Manajement System*). MySQL mendukung bahasa pemrograman PHP, bahasa permintaan yang terstruktur, karena pada penggunaannya SQL memiliki beberapa aturan yang telah distandarkan oleh asosiasi yang bernama ANSL. MySQL merupakan RDBMS (*Relational Database Management System*) server. RDBMS adalah program yang memungkinkan pengguna *database* untuk membuat, mengelola, dan menggunakan data pada suatu model relational. Dengan demikian, tabel-tabel yang ada pada *database* memiliki relasi antara satu tabel dengan tabel lainnya [8].

2.2.4 Penjualan

Penjualan (*sales*) adalah aktivitas atau bisnis dalam menjual produk atau jasa. Aktivitas penjualan adalah hal yang sangat penting bagi perusahaan, terutama untuk meraih keuntungan. Pengertian penjualan secara umum adalah kegiatan jual beli dijalankan oleh dua belah pihak atau lebih dengan alat pembayaran yang sah. Penjualan ini dapat dilakukan dengan berbagai metode, seperti penjualan langsung serta agen penjualan. Tujuan utama penjualan adalah mendatangkan keuntungan dari produk atau barang yang dijual. Dalam pelaksanaannya, penjualan tidak dapat dilakukan tanpa adanya kontribusi dari pelaku yang bekerja, seperti pedagang, agen, dan tenaga pemasaran [9].

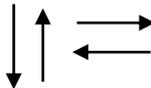
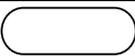
2.2.5 Flowchart

Flowchart adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Ada beberapa jenis-jenis *flowchart* diantaranya:

- A. Bagan alir sistem (*system flowchart*)
- B. Bagan alir dokumen (*document flowchart*)
- C. Bagan alir skematik (*schematic flowchart*)
- D. Bagan alir program (*program flowchart*)
- E. Bagan alir proses (*process flowchart*)

Simbol-simbol dalam *flowchart* dapat dilihat pada Table 2.1.

Table 2. 1 Simbol *Flowchart*

No	Simbol	Keterangan
1	 Flow Direction Symbol	Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga <i>connecting line</i> .
2	 Terminator Symbol	Yaitu simbol untuk permulaan (<i>start</i>) atau akhir (<i>end</i>) dari suatu kegiatan.
3	 Processing Symbol	Yaitu simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.
4	 Manual Operation Symbol	Yaitu simbol yang menunjukkan pengolahan data yang tidak dilakukan oleh komputer.

5	 Decision Symbol	Yaitu simbol untuk pemilihan proses berdasarkan kondisi yang ada.
5	 Input-Output Symbol	Yaitu simbol yang menyatakan proses input dan output tanpa tergantung dari jenis peralatannya.
6	 Document Symbol	Yaitu simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

2.2.6 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah hasil kerja dari konsorium berbagai organisasi yang berhasil dijadikan sebagai standar baku dalam *Object Oriented Analysis dan Design (OOAD)*. Macam-macam dari *Unified Modeling Language (UML)* antara lain: *use case diagram*, *sequence diagram* dan *class diagram*.

a. Use Case

Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem tersebut *scenario* sedangkan pengguna disebut *actor*. *Actor* adalah sebuah peran yang biasa dimainkan oleh pengguna dalam interaksinya dengan sistem. Model *use case* adalah bagian dari model *requirement*. Definisi lain *use case* adalah abstraksi dari interaksi antara sistem dan *actor*. *Use case* dibuat berdasarkan keperluan *actor*.

Berdasarkan definisi diatas maka dapat disimpulkan bahwa *Use Case* adalah kontruks untuk mendeskripsikan bagaimana sistem akan terlihat dimata pengguna potensial yang terdiri dari sekumpulan *scenario* dan *actor*. Sedangkan *use case diagram* memfasilitasi komunikasi diantara analis dan pengguna serta analis dan klien. Simbol *Use Case* dapat dilihat pada Table 2.2.

Table 2. 2 Simbol *Use Case Diagram*

No.	Simbol	Nama Simbol dan Keterangan
1.		<i>Actor</i> , Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Dependency</i> , Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent).
3.		<i>Generalization</i> , Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
4.		<i>Include</i> , Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5.		<i>Extend</i> , Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>Association</i> , yang menghubungkan antara objek satu dengan objek lainnya
7.		<i>System</i> , Menspesifikasikan paket yang menampilkan sistem secara terbatas
8.		<i>Use Case</i> , Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang

		menghasilkan suatu hasil yang terukur bagi suatu actor.
9.		<i>Collaboration</i> , Interaksi aturan-aturan dan elemen lain yang bekerjasama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10.		<i>Note</i> , Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

b. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah obyek dan *message* yang diletakan antara obyek-obyek didalam *use case*.

Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*. *Sequence Diagram* menambahkan dimensi waktu pada interaksi diantara obyek. Simbol-simbol yang dipakai dalam pembuatan *sequence diagram* dapat dilihat pada Table 2.3.

Table 2. 3 Simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
1	 Actor	<i>Actor</i>	Menggambarkan <i>user</i> atau pengguna.
2	 Entity Object	<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan.

3		<i>Boundary Class</i>	Menggambarkan sebuah form
4		<i>Control Class</i>	Menghubungkan antara <i>boundary</i> dengan tabel.
5		<i>Lifeline</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.
6		Activation bar	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya. Menggambar lamanya suatu pesan di proses
7		<i>Message</i>	Menggambarkan pengiriman pesan.

c. *Class Diagram*

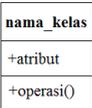
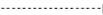
Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas. Sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas [10]. Diagram kelas dibuat agar *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

Class diagram membantu dalam visualisasi struktur kelas – kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak. *Class*

diagram memperlihatkan hubungan antar kelas dan penjelasan detail tiap – tiap kelas di dalam model desain (dalam *logical view*) dari suatu sistem. Selama proses analisis, *class diagram* memperlihatkan aturan – aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama proses analisis, *class diagram* memperlihatkan aturan – aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap *decian*, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat

Berikut beberapa simbol dari *class diagram*:

Table 2. 4 Simbol *Class Diagram*

No.	Simbol	Nama	Keterangan
1.		Kelas	Kelas pada struktur sistem.
2.		Asosiasi/ <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
3.		Asosiasi berarah/ <i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
4.		Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
5.		Kebergantungan / <i>Dependency</i>	Relasi antarkelas dengan makna

6.		Agregasi / <i>Aggregation</i>	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).
			kebergantungan antarkelas.