

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Berkaca dari pesatnya laju perkembangan teknologi modern, sistem *Management of Change* (MOC) akan lebih efektif jika menggunakan sistem komputerisasi. Sistem ini akan membuat pengelolaan data yang lebih aman.

(Fransiska, 2012) membuat sebuah penelitian yang berjudul “ANALISIS DAN PERANCANGAN APLIKASI MANAJEMEN PROYEK KHUSUSNYA PERMINTAAN PERUBAHAN (*CHANGE REQUEST*) BERBASIS WEB UNTUK DIVISI IT PADA PT WOM FINANCE, TBK”. Penelitian tersebut bertujuan agar dapat memudahkan *project manager* yang ada di divisi teknologi informasi untuk menghasilkan informasi yang akurat sehingga *project manager* dapat mengambil keputusan yang tepat untuk ke depannya.

Pembuatan aplikasi *Management of Change* juga telah dilakukan sebelumnya dalam penelitian yang berjudul “APLIKASI CUSTOMER CHANGE REQUEST (CCR) BERBASIS WEB PADA PT AERO SYSTEMS INDONESIA” Penelitian tersebut salah satunya bertujuan untuk membantu dalam pengelolaan data *CCR* (studi kasus PT Aero System Indonesia) (Fitri, 2011).

Dari kedua penelitian sebelumnya, aplikasi yang dibuat hanya berupa pengelolaan data *MOC*. Dalam skripsi ini, penulis membutuhkan aplikasi yang tidak hanya berupa aplikasi pengelolaan data *MOC* saja. Berdasarkan pustaka tersebut, maka penulis menambahkan fitur-fitur sebagai berikut:

1. Sistem dapat mencetak *observation agreement*.
2. Sistem dapat mengunggah *observation agreement*.

2.2. Landasan Teori

2.2.1. Definisi *Management of Change*

Management of Change (MOC) adalah serangkaian proses yang digunakan untuk memastikan bahwa perubahan strategis yang signifikan dalam organisasi dilakukan secara terkontrol dan sistematis, untuk mengatasi resistensi terhadap perubahan dalam rangka meningkatkan keterlibatan dan pencapaian tujuan organisasi untuk transformasi efektif. Pencapaian perubahan yang berkelanjutan dimulai dengan pemahaman yang jelas tentang keadaan organisasi saat ini, diikuti dengan pelaksanaan strategi yang tepat dan ditargetkan (Tresnajaya, 2014).

2.2.2. Penilaian Risiko

Penilaian risiko adalah metode sistematis dalam melihat aktivitas kerja, memikirkan apa yang dapat menjadi buruk, dan memutuskan kendali yang cocok untuk mencegah terjadinya kerugian, kerusakan, atau cedera di tempat kerja. Penilaian ini harus juga melibatkan pengendalian yang diperlukan untuk menghilangkan, mengurangi, atau meminimalkan risiko (Antariksa, 2010).

2.2.3. *Observation Agreement*

Observation agreement atau persetujuan observasi merupakan perjanjian yang secara khusus dipergunakan untuk menyebut kontrak antar dua pihak atau lebih mengenai aktivitas terhadap suatu proses atau objek dengan maksud merasakan yang bertujuan untuk mendapatkan informasi-informasi yang dibutuhkan untuk melanjutkan suatu penelitian.

2.2.4. Aplikasi *Desktop*

Aplikasi *desktop* adalah suatu aplikasi yang dapat berjalan sendiri dalam suatu komputer dengan *operating system* atau *platform* tertentu tanpa menggunakan *browser* atau koneksi *Internet*. Salah satu bahasa pemrograman yang digunakan dalam pengembangan aplikasi *desktop* adalah bahasa pemrograman *C#* (Konixbam, 2009).

2.2.5. Definisi Bahasa Pemrograman C#

C# (dibaca *C sharp*) adalah bahasa pemrograman baru yang diciptakan oleh *Microsoft* yang dikembangkan di bawah kepemimpinan Anders Hejlsberg yang telah menciptakan berbagai macam bahasa pemrograman termasuk *Borland Turbo C++* dan *Orland Delphi*. Bahasa *C#* juga telah distandardisasi secara internasional oleh *ECMA*. Seperti halnya bahasa pemrograman yang lain, *C#* bisa digunakan untuk membangun berbagai macam jenis aplikasi, seperti aplikasi berbasis *windows (desktop)* dan aplikasi berbasis *web* serta aplikasi berbasis *web services* (Syakir, 2015).

2.2.6. Windows Form

Windows Forms (WinForms) adalah komponen klien dari *framework*, memungkinkan tugas aplikasi umum. Pada *windows*, bentuknya adalah visual yang menampilkan informasi kepada penggunanya. Untuk membangun aplikasi *WinForms* dengan menempatkan kontrol pada bentuk dan mengembangkan tanggapan terhadap tindakan pengguna (Azim, 2015).

2.2.7. SQL Server

Structured Query Language (SQL) Server adalah sistem manajemen *database* relasional (*RDBMS*) yang dirancang untuk aplikasi dengan arsitektur *client/server*. Istilah *client*, *server*, dan *client/server* dapat digunakan untuk merujuk kepada konsep yang sangat umum atau hal yang spesifik dari perangkat keras atau perangkat lunak. Pada *level* yang sangat umum.

Client adalah setiap komponen dari sebuah sistem yang meminta layanan atau sumber daya (*resource*) dari komponen sistem lainnya. *Server* adalah setiap komponen sistem yang menyediakan layanan atau sumber daya ke komponen sistem lainnya. *RDBMS* adalah dasar untuk *SQL* dan untuk semua sistem *database modern* seperti *MS SQL Server*, **IBM DB2*, *Oracle*, *MySQL*, dan *Microsoft Access* (Mustofa, 2013).


2.2.8. *Unified Markup Language*

Unified Markup Language (UML) merupakan bahasa pemodelan secara grafis yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan seluruh rancangan aplikasi perangkat lunak. Penggunaan model UML bertujuan untuk mengidentifikasi bagian-bagian yang termasuk dalam lingkup sistem di dalam aplikasi. Model UML yang dipakai dalam pengembangan aplikasi penjualan dan pembelian antara lain adalah *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram* (Adiyasa, 2017).






a. *Use Case Diagram*

Use Case Diagram mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem yang dibuat. Dapat dikatakan *Use Case* digunakan untuk mengetahui fungsi yang ada di dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi yang dibutuhkan. Simbol-simbol yang digunakan dalam *Use Case Diagram* dapat dilihat pada Tabel 2.1 dan Tabel 2.2.

Tabel 2.1 Simbol-Simbol Dalam *Use Case Diagram*

NO	Gambar	Nama Gambar	Keterangan
1.		<i>Use Case</i>	Merupakan fungsionalitas yang disediakan sistem sebagai unit yang bertukar pesan dengan <i>actor</i> .





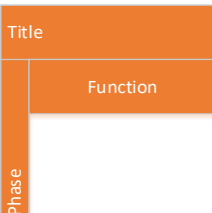
Tabel 2.2 Simbol-Simbol Dalam *Use Case Diagram* (lanjutan)

NO	Gambar	Nama Gambar	Keterangan
2.		<i>Actor</i>	Merupakan <i>abstraction</i> dari orang yang mengaktifkan fungsi dari target sistem dan merupakan orang yang berinteraksi dengan <i>use case</i> .
3.		<i>Association</i>	Digambarkan dengan garis tanpa panah yang mengindikasikan siapa yang berinteraksi secara langsung dengan sistem.
4.		<i>Generalization</i>	Mengindikasikan siapa yang berinteraksi secara pasif dengan sistem.
5.		<i>Include</i>	Mengidentifikasi hubungan antar dua <i>use case</i> dimana satu <i>usecase</i> memanggil <i>usecase</i> yang lain.
6.		<i>Extend</i>	Merupakan perluasan dari <i>use case</i> jika kondisi atau syarat terpenuhi.

b. Activity Diagram

Activity Diagram merupakan diagram yang digunakan untuk menggambarkan *work flow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *Activity Diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol-Simbol Dalam *Activity Diagram*

NO	Gambar	Nama Gambar	Keterangan
1.		<i>Start Point</i>	Merupakan awal dalam aktifitas
2.		<i>End Point</i>	Merupakan akhir dalam aktifitas.
3.		<i>Activities</i>	Menggambarkan suatu proses atau kegiatan bisnis
4.		<i>Decision Point</i>	Menggambarkan pilihan untuk pengambilan keputusan dalam aktifitas.
5.		<i>Swimlane</i>	Digunakan untuk pembagian <i>activity diagram</i> yang menunjukkan siapa yang melakukan aktifitas.

c. *Class Diagram*

Class Diagram merupakan diagram yang digunakan untuk menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class Diagram* secara khas meliputi: Nama Kelas (*Class Name*), Atribut (*Attributes*), Operasi (*Operations*), dan Relasi (*Relationships*).