



BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian sebelumnya yang telah dilakukan oleh Ismail [1]. Penelitian ini menjelaskan bahwa masalah yang terjadi adalah pada pelaporan harian masih memanfaatkan *group chat* untuk melaporkan kegiatan dan mengisi form laporan ketika berada dikantor saja. Aplikasi ini dibuat menggunakan metode wawancara dan metode pengembangan sistem berupa *Data Flow Diagram* (DFD) yang bertujuan untuk memudahkan para pengawas lapangan dalam melakukan pelaporan progress proyek serta memudahkan *owner* CV. Bina Cipta Consultant dalam melakukan pemantauan setiap proyek. Hasil penelitian menyatakan bahwa sistem ini dapat memberikan laporan harian proyek yang telah diinputkan dan juga membantu *owner* dalam melakukan pemantauan progress dari setiap proyek dengan mudah.

Penelitian lain telah dilakukan oleh Dhuha dkk[2]. Penelitian ini menjelaskan bahwa PT. Swadaya Graha (Semen Indonesia Group) adalah perusahaan yang bergerak di bidang manufaktur dan jasa konstruksi. Penelitian ini dibuat menggunakan metode *waterfall* yang bertujuan untuk memberikan data laporan secara *real-time*, dan sebagai sarana informasi bagi para pemangku kepentingan. Hasil penelitian menyatakan bahwa sistem dapat menyimpan data progress proyek kedalam sistem basis data.

Penelitian lain juga dilakukan oleh W. Bimantoro [3]. Penelitian ini menjelaskan masalah yang terjadi adalah berkas proyek terkadang hilang atau rusak karena masih berbentuk *hardcopy*. Sistem ini dibuat untuk memudahkan admin CV. Dwi Tunggal Abadi dalam melakukan pencatatan data proyek, memudahkan pencarian data proyek, dan memudahkan pembuatan laporan proyek. Sistem ini dibuat menggunakan *framework Codeigniter* dan metode pengembangan *Data Flow Diagram* (DFD). Hasil penelitian menyatakan bahwa sistem dapat melakukan pencatatan data proyek agar memudahkan dalam perekapan.

Penelitian lain juga dilakukan oleh Fajri, dkk [4]. Penelitian yang dibuat menggunakan metode *waterfall* ini dilakukan terhadap progres pengerjaan proyek bidang bina marga bertujuan untuk menghasilkan aplikasi untuk melakukan monitoring progres pekerjaan proyek berbasis web dan tersedia aplikasi pelaporan progres pekerjaan proyek yang

terintegrasi sehingga memudahkan instansi dalam memonitoring proyek secara *online*. Hasil penelitian ini menyatakan bahwa aplikasi ini mampu mempermudah kinerja bidang bina marga terbukti dengan hasil wawancara yang menunjukkan bahwa aplikasi yang telah di bangun dapat bekerja sesuai keinginan *user*.

Penelitian lain juga telah dilakukan oleh Akbar, dkk [5]. Penelitian ini menggunakan metode prototype ini diharapkan pihak perusahaan dapat memperoleh informasi lebih cepat sehingga pengerjaan proyek dapat diselesaikan sesuai target pengerjaan proyek, dan diharapkan sistem informasi ini dapat membantu dalam mengambil keputusan dan melakukan pengawasan terhadap proyek yang dijalankan CV. Radityatama Engineering Konsultan. Hasil penelitian ini adalah sebuah sitem informasi monitoring perkembangan proyek berbasis web.

Penulis bermaksud membuat Aplikasi Pemantauan Progress Proyek Berbasis Web, dimana sistem web ini dibuat menggunakan metode *Rapid Application Development* (RAD) dan dapat diakses pada PC atau Laptop serta *smarthphone*. Aplikasi ini akan memberikan kemudahan dalam melakukan pengawasan proyek, pencatatan data, pencarian data, serta memberikan informasi terkait bahan material. Sistem ini diharapkan dapat membantu pimpinan, administrasi dan keuangan, serta pengawas untuk mempermudah dalam menjalankan perusahaan.

2.2 Landasan Teori

2.2.1 Rekayasa Perangkat Lunak

Perangkat lunak (*software*) merupakan sebuah program yang berisi kumpulan perintah untuk melakukan proses mengolah data. *Software* digunakan sebagai penghubung antara manusia dan perangkat keras komputer, berfungsi untuk menerjemahkan bahasa manusia ke dalam bahasa mesin sehingga mesin dapat mengerti keinginan pengguna dan kemudian menjalankan perintah yang diberikan. Selanjutnya *software* memberikan hasil yang diinginkan [6].


A. Sistem Informasi

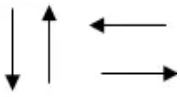





Sistem informasi merupakan suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Sistem informasi di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Sistem informasi merupakan cara untuk mengumpulkan atau menghimpun data – data terkait sistem informasi dan mempublikasikannya secara *electronic* melalui internet [7].


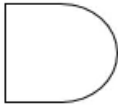
B. Flowchart

Flowchart adalah sebuah bentuk representasi simbolik dari prosedur atau algoritma untuk menyelesaikan suatu masalah. Penggunaan *flowchart* akan mempermudah pengguna dalam melakukan pengecekan bagian yang terlupakan dalam menganalisis masalah. Simbol flowchart serta keterangannya terdapat pada Tabel 2.1 [12].

Tabel 2. 1 Simbol Flowchart

No	Gambar Simbol	Nama	Keterangan
1.		<i>Terminal point</i>	Menunjukkan mulai atau berakhirnya suatu proses.

2.		<i>Flow direction</i>	Simbol yang digunakan untuk menghubungkan antar simbol, berfungsi juga untuk menunjukkan alur sebuah proses.
3.		Proses	Menunjukkan sebuah kegiatan yang dilakukan oleh komputer atau sistem.
4.		<i>Decision</i>	Simbol yang digunakan untuk memilih sebuah keputusan atau proses sesuai dengan kondisi yang ada.
5.		<i>Input output</i>	Menunjukkan sebuah <i>input</i> atau <i>output</i> yang ada tidak bergantung pada jenis peralatannya.
6.		<i>Manual input</i>	Menunjukkan proses <i>input</i> yang dilakukan secara <i>manual</i> menggunakan <i>online keyboard</i> .
7.		<i>Manual operation</i>	Menunjukkan proses/kegiatan yang dilakukan tanpa

			menggunakan komputer.
8.		<i>Document</i>	Menunjukkan <i>input</i> berupa dokumen dalam bentuk kertas atau <i>output</i> yang harus dicetak.
9.		<i>Display</i>	Simbol yang menunjukkan adanya penggunaan peralatan <i>output</i> .

C. Metode Pengujian Sistem

Metode pengujian sistem yang digunakan adalah metode *black box testing*. Black box testing merupakan pengujian kualitas perangkat lunak yang berfokus pada fungsionalitas perangkat lunak. Pengujian black box bertujuan untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan pada struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi [9].

D. Uji Usability

Uji *Usability* merupakan salah satu cara untuk mengetahui apakah user dapat mudah menggunakan aplikasi, seberapa efisien dan efektif sebuah aplikasi dapat membantu user mencapai tujuannya dan apakah user puas dengan aplikasi yang digunakan [14].

2.2.2 Pemrograman Berbasis Objek

Pemrograman berorientasi objek atau *object oriented programming* (OOP) adalah suatu pemrograman berdasarkan konsep “objek”, yang berisi data dalam bentuk kode serta, dalam bentuk field atau biasa disebut dengan atribut yang dibuat dalam bentuk fungsi atau dikenal sebagai method. Semua fungsi dan data yang ada terdapat didalam kelas-




kelas. Ada beberapa keuntungan yang didapat jika membuat Program berorientasi objek, diantaranya ;



1. *Reusability*, kode yang sudah dibuat dapat digunakan kembali.
2. *Extensibility*, pemrogram bisa membuat method baru atau mengubah method yang sudah ada kemudian diubah sesuai keinginannya tanpa harus membuat kode dari awal.
3. *Maintainability*, program yang sudah dibuat akan lebih mudah dikelola jika program yang dibuat dalam skala besar yang memungkinkan terjadinya *error* dalam proses pengembangannya dapat diatasi dengan OOP karena OOP sudah menggunakan konsep modularitas [10].

A. Use Case Diagram

Use case merupakan metode pemodelan untuk menggambarkan *external view* sistem yang akan kita buat. *Use case* digunakan untuk menyatakan fungsionalitas yang disediakan dan digunakan secara intensif untuk menghimpun konteks system. Simbol use case serta keterangannya terdapat pada Tabel 2.2 [11].

Tabel 2. 2 Simbol *Use Case Diagram*



No	Gambar Simbol	Nama	Keterangan
1.		<i>Use Case</i>	Deskripsi urutan dari aksi-aksi yang ditampilkan sistem bagi suatu <i>actor</i> .
2.		<i>Actor</i>	<i>Actor</i> merupakan orang, sistem atau perangkat keras yang akan berhubungan dengan <i>use case</i> .
3.		<i>Extend</i>	Menunjukkan bahwa <i>use case</i> target merupakan tambahan fungsionalitas dari <i>use case</i> lainnya.

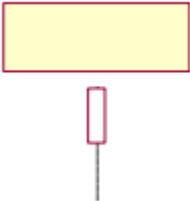
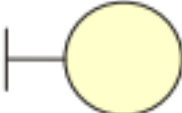
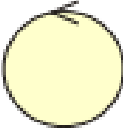
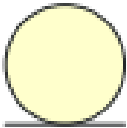



4.		<i>Association</i>	Penghubung antara <i>actor</i> dan <i>use case</i> .
5.		<i>Include</i>	Menunjukkan sebuah <i>use case</i> memerlukan <i>use case</i> yang lain untuk menjalankan fungsinya.

B. *Sequence Diagram*

Sequence diagram merupakan gambaran dari tahap-tahap atau urutan yang harus dilakukan agar dapat menghasilkan sesuatu sesuai dengan *use case diagram*. Sebuah *use case diagram* akan menjadi sebuah *sequence diagram*. *Sequence diagram* menjelaskan interaksi antar objek dan menjelaskan komunikasi yang terjadi antar objek. Simbol *Sequence diagram* beserta keterangannya terdapat pada Tabel 2.3 [11].

Tabel 2. 3 Simbol *Sequence Diagram*

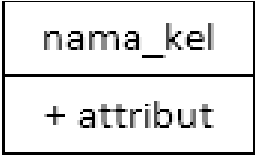




No	Gambar Simbol	Nama	Keterangan
1.		<i>Actor</i>	<i>Actor</i> menggambarkan entitas yang berada diluar sistem dan berinteraksi dengan sistem.
2.		<i>Lifeline</i>	<i>Lifeline</i> digunakan untuk mengeksekusi sebuah objek didalam <i>sequence</i> .

3.		<i>General</i>	<i>General</i> digunakan untuk menggambarkan sebuah entitas tunggal.
4.		<i>Boundary</i>	<i>Boundary</i> digunakan untuk berinteraksi dengan sistem lain yang dapat berupa.
5.		<i>Control</i>	<i>Control</i> digunakan untuk mengatur aliran dari informasi yang menuju scenario.
6.		<i>Entity</i>	<i>Entity</i> digunakan untuk menunjukkan sebuah penyimpanan data atau informasi.
7.		<i>Message</i>	<i>Message</i> digunakan saat dua objek yang berbeda bertemu.
8.		<i>Message to self</i>	<i>Message to self</i> digunakan untuk menggambarkan hubungan antar objek itu sendiri serta diurutkan dengan kejadian yang terjadi.
9.		<i>Message return</i>	<i>Message return</i> digunakan untuk menggambarkan hasil dari pengiriman <i>message</i> .

C. Class Diagram

Class diagram adalah penghubung antar objek, yang juga dapat memodelkan kelas, atribut, dan operasi. *Class diagram* dapat memberikan gambaran mengenai sebuah sistem yang lebih luas dengan cara memperlihatkan kelas – kelas dan hubungannya. Simbol Class diagram beserta keterangan terdapat pada Tabel 2.4 [12].

Tabel 2. 4 Simbol Class Diagram

No	Gambar Simbol	Keterangan
1.		Kelas , kelas pada struktur sistem.
2.		Interface , sama dengan konsep pemrograman berorientasi objek (OOP).
3.		Association , relasi antar <i>class</i> disertai dengan <i>multiplicity</i> .
4.		Directed Association , relasi antar <i>class</i> dengan arti umum.
5.		Dependency , menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain.

2.2.3 Basis Data




Basis data adalah kelompok himpunan dari data yang saling terorganisir dan terhubung sedemikian rupa sehingga dapat dimanfaatkan kembali secara cepat dan mudah. Pengertian lainnya adalah berbagai data dalam bentuk arsip/file/tabel yang terhubung dan tersimpan pada penyimpanan elektronik, guna memudahkan pengaturan, pemilahan, pengorganisiran, dan pengelompokan. Basis data memiliki beberapa komponen bahasa, menurut fungsinya, bahasa basis data meliputi DDL


atau *Data Definition Language* yang merupakan kumpulan definisi yang tersimpan dalam data *dictionary*. Contoh perintah DDL yaitu *CREATE*, *ALTER*, *RENAME*, dan *DROP*. Selanjutnya ada DML atau *Data Manipulation Language* yang berisi akumulasi pengoperasian manipulasi basis data atau biasa disebut dengan bahasa *query*. Menyediakan fungsi untuk menambah, memodifikasi, menghapus dan mengambil data dari basis data. Contoh dari perintah DML yaitu *INSERT*, *UPDATE*, *DELETE*, dan *SELECT* [10].

A. ERD (*Entity Relationship Diagram*)

Entity relationship diagram merupakan sebuah teknik pendekatan yang menggambarkan hubungan antara model. Dalam hubungan tersebut ERD menunjukkan objek yang berupa data (*Entity*) dan hubungan (*Relationship*) yang ada pada *Entity* berikutnya. Menurut Simarmata, *Entity Relationship Diagram* adalah suatu pemodelan data yang membantu untuk mengatur data dalam suatu proyek ke dalam entitas-entitas dan menentukan hubungan entitas tersebut. Simbol ERD serta keterangannya terdapat pada Tabel 2.5 [12].

Tabel 2. 5 Simbol Entity Relationship Diagram

No	Gambar Simbol	Nama	Keterangan
1.		<i>Entity</i>	<i>Entity</i> merupakan kumpulan dari objek yang dapat diidentifikasi secara unik.
2.		<i>Relationship</i>	<i>Relationship</i> digunakan untuk menghubungkan entitas yang ada.
3.		Atribut	Atribut merupakan karakteristik unik

			yang dimiliki oleh entitas.
4.		<i>Association</i>	<i>Association</i> merupakan penghubung antara entitas dan relasi atau entitas dengan entitas.

2.2.4 Rekayasa Web

Rekayasa Web yaitu model rekayasa suatu perangkat lunak yang berguna dalam pengembangan aplikasi berbasis web. Sedangkan menurut Pressman, Rekayasa web merupakan proses yang berguna untuk membuat aplikasi web dengan kualitas tinggi. Rekayasa web dalam prosesnya menekankan pada aktivitas teknis dan manajemen [10].

2.2.5 Manajemen Proyek

Manajemen Proyek adalah Suatu proses manajemen pada suatu proyek dari awal hingga akhir proyek agar tujuan proyek tercapai dengan baik, tepat waktu, sesuai mutu yang disyaratkan dan sesuai biaya yang disediakan. Manajemen Proyek merupakan penerapan pengetahuan, keterampilan, alat, dan teknik untuk kegiatan proyek agar memenuhi persyaratan proyek. Dan semua harus dikelola secara ahli untuk memberikan hasil yang tepat waktu, sesuai anggaran, pembelajaran dan integrasi yang dibutuhkan organisasi [13]

A. Proyek Konstruksi

Konstruksi berasal dari kata "membangun", membangun istana pasir, benteng dari bantal, atau rumah kartu adalah contoh dari membangun sesuatu. Dalam istilah teknik, konstruksi biasanya dikaitkan dengan struktur besar seperti rumah, rel kereta api, dan pembangkit listrik.

Arti konstruksi dalam istilah teknik, adalah kegiatan menyatukan elemen yang berbeda, menggunakan desain dan rencana terperinci, untuk membuat struktur untuk lokasi tertentu. Saat kamu membangun struktur besar, kamu perlu memiliki rencana yang jelas tentang bagaimana kamu akan melakukannya. kamu juga perlu mengetahui lokasi spesifiknya. Arsitek dan insinyur merancang dan membangun struktur dengan mempertimbangkan lokasi tersebut [13].

2.2.6 Metode Pengembangan Sistem

Metodologi adalah ilmu pemahaman atau cara yang digunakan untuk memperoleh suatu kebenaran atau memecahkan masalah menggunakan penelusuran dengan urutan atau tatacara tertentu sesuai dengan apa yang akan dikaji atau diteliti [15].

A. Tahap pengumpulan data

Tahap pengumpulan data dapat diperoleh secara langsung dari objek penelitian. Tahapan tersebut diantaranya :

a) Studi Pustaka

Studi ini dilakukan dengan cara mempelajari, meneliti, dan menelaah berbagai catatan serta berbagai laporan yang berkaitan dengan masalah yang akan diselesaikan.

b) Studi lapangan

Dilakukan dengan cara pengumpulan data secara langsung ke lapangan dengan menggunakan teknik pengumpulan data sebagai berikut :

1. Wawancara

Yaitu teknik pengumpulan data dengan cara tanya jawab dengan direktur CV. Kuku Putra Mandiri.

2. Observasi

Yaitu teknik pengumpulan data dengan melakukan pengamatan secara langsung di CV. Kuku Putra Mandiri.

B. Tahap Pengembangan Sistem

Dalam mengembangkan Aplikasi Pemantauan Progress Proyek Berbasis Web, penulis menggunakan metode *Rapid Application Development* (RAD). Tahapan pengembangan sistem berdasarkan sebagai berikut yang terdapat pada Gambar 2.1.



Gambar 2. 1 Metode RAD

Tahap – tahap pada metode ini sebagai berikut :

a. Rencana Kebutuhan (*Requirement Planning*)

User dan *analyst* mengadakan pertemuan untuk mengetahui tujuan dari sistem dan kebutuhan informasi untuk mencapai tujuan. Tahap ini merupakan hal terpenting karena adanya keterlibatan dari kedua pihak.

b. Proses Desain Sistem (*Design System*)

Tahapan ini adalah dimana keaktifan *user* yang terlibat menentukan apakah sistem ini mencapai tujuan atau tidak karena di dalam proses ini terdapat proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara *user* dan *analyst*. *User* dapat langsung memberikan saran atau masukan apabila terdapat ketidaksesuaian pada desain. *Analyst* merancang sistem dengan mengacu pada dokumentasi kebutuhan *user* yang dibuat pada tahap sebelumnya.

c. Implementasi (*Implementation*)

Tahap saat programmer yang mengembangkan desain telah disetujui oleh *user* dan *analyst*. Programmer akan melakukan pengujian sebelum diaplikasikan pada suatu organisasi untuk mengetahui apakah ada kesalahan atau tidak. *User* dapat memberikan tanggapan atas sistem yang sudah dibuat [16].

~Halaman Ini Sengaja Dikosongkan~