

BAB II

LANDASAN TEORI

2.1 Penelitian Sebelumnya

Boy (2012) melakukan penelitian tentang Aplikasi *Monitoring* Kedatangan Kapal dan Penjadwalan Kerja di Koperasi Tenaga Kerja Bongkar Muat Usaha Karya, dimana pada pembuatan aplikasi *monitoring* ini berpusat pada proses penjadwalan kapal. *Fitur* tambahan dari aplikasi ini adalah menggunakan teknologi *sms gateway*. *Fitur* tersebut memberitahukan informasi *monitoring* dan penjadwalan kapal ke para pekerja, hal ini sangat menghemat waktu untuk pemberitahuan informasi *monitoring* dan penjadwalan kapal.

Dari penelitian di atas, perbedaannya adalah informasi *monitoring* yang ditampilkan dalam bentuk visualisasi informasi yang bertujuan untuk memudahkan *user* dalam memahami informasi *monitoring* yang diberikan. Aplikasi yang dihasilkan akan menggunakan visualisasi informasi berupa *dashboard* yang menggunakan *operational strategic*. Proses *monitoring* dilakukan untuk proyek pembuatan kapal yang berbeda dengan penelitian sebelumnya yang mana penelitian sebelumnya berfokus pada penjadwalan bongkar muat kapal.

2.2 Monitoring

Menurut Mercy (2005) *monitoring* didefinisikan sebagai siklus kegiatan yang mencakup pengumpulan, peninjauan ulang, pelaporan dan tindakan atas informasi suatu proses yang sedang diimplementasikan. Menurut Cheryl (2005) *monitoring* adalah proses penilaian kualitas kinerja untuk pengendalian internal dari waktu ke waktu. *Monitoring* sangat penting karena kebanyakan operasional

perusahaan terus berubah untuk memenuhi kebutuhan pasar dan untuk mendapatkan keuntungan.

1.2.1 Efektifitas Sistem *Monitoring*

Sistem *monitoring* akan memberikan dampak yang baik bila dirancang dan dilakukan secara efektif. Menurut Mercy (2005) kriteria sistem *monitoring* yang efektif adalah sebagai berikut ini:

1. Sederhana dan mudah dimengerti (*user friendly*). *Monitoring* harus dirancang dengan sederhana namun tepat sasaran. Konsep yang digunakan adalah singkat yaitu sederhana, jelas yaitu mudah dimengerti, dan padat yaitu berbobot.
2. Focus pada beberapa indikator utama. Indikator diartikan sebagai titik krisis dari suatu *scope* tertentu. Banyaknya indikator membuat pelaku dan obyek *monitoring* tidak fokus. Maka itu, fokus diarahkan pada indikator utama yang benar-benar mewakili bagian yang dipantau.
3. Perencanaan matang terhadap aspek-aspek penting. Tujuan perancangan sistem adalah aplikasi teknis yang terarah dan terstruktur. Maka itu, perencanaan aspek teknis harus dipersiapkan secara matang dan mengandung unsur 5W1H.
4. Prosedur pengumpulan dan penggalian data. Data yang didapatkan dalam pelaksanaan *monitoring* pada *on going process* harus memiliki prosedur yang tepat dan sesuai. Selain itu, hal tersebut bertujuan untuk memudahkan pelaksanaan proses masuk dan keluarnya data.

2.2.1 Tujuan Sistem *Monitoring*

Menurut Amsler (2009) berikut ini merupakan beberapa tujuan dari sistem *monitoring*, antara lain:

1. Memastikan suatu proses dilakukan sesuai dengan prosedur yang berlaku. Sehingga, proses berjalan sesuai jalur yang disediakan (*on the track*).
2. Menyediakan probabilitas tinggi akan keakuratan data bagi pelaku *monitoring*.
3. Mengidentifikasi hasil yang diinginkan pada suatu proses dengan cepat.
4. Menumbuh kembangkan motivasi dan kebiasaan positif pekerja.

2.3 *Web*

Menurut Sutarman (2003) *website* (situs web) adalah merupakan alamat (URL) yang berfungsi sebagai tempat penyimpanan data dan informasi dengan berdasarkan topik tertentu. *Website* disebut juga dengan WWW (*World Wide Web*) adalah jaringan beribu-ribu komputer dikategorikan menjadi dua, yaitu *server* dan *client* dengan menggunakan software khusus membentuk sebuah jaringan yang disebut jaringan *client-server*. URL (*Uniform Resource Locator*) adalah suatu sarana yang digunakan untuk menentukan lokasi informasi pada suatu *web server* dan dapat diibaratkan suatu alamat, dimana alamat tersebut terdiri dari tiga bagian, yaitu protokol yang digunakan oleh suatu browser untuk mengambil informasi, nama komputer (*server*) dimana informasi tersebut berada, dan jalur atau *path* serta nama file dari suatu informasi.

2.4 *Key Performance Indicator (KPI)*

Untuk indikator yang digunakan dalam laporan pengerjaan proyek pembuatan kapal PT Orela Group, sesuai dengan keterangan Staff yang berperan

sebagai bagian yang mengurus pembuatan laporan pengerjaan proyek akan dijelaskan pada Tabel 2.1.

Tabel 2.1 Tabel indikator *dashboard* laporan pengerjaan proyek

No.	Indikator	Unit Pengukuran	Grade	Persentase (%)
1.	Persentase progres proyek per proyek	Persen (%)	Baik	81 - 100
			Kurang	0 - 80
2.	Biaya keseluruhan proyek per proyek	Rupiah (Rp)	Baik	81 - 100
			Kurang	0 - 80
3.	Biaya pemakaian material per proyek	Rupiah (Rp)	Baik	81 - 100
			Kurang	0 - 80
4.	Biaya sumber daya manusia per proyek	Rupiah (Rp)	Baik	81 - 100
			Kurang	0 - 80
5.	Pemakaian jumlah material per proyek	Angka	Baik	81 - 100
			Kurang	0 - 80
6.	Pembagian durasi waktu main task per proyek	Angka	Baik	81 - 100
			Kurang	0 - 80
7.	Pembagian jumlah tenaga kerja main task per proyek	Angka	Baik	81 - 100
			Kurang	0 - 80
8.	Pemakaian durasi waktu main task per proyek	Angka	Baik	81 - 100
			Kurang	0 - 80
9.	Pemakaian jumlah tenaga kerja main task per proyek	Angka	Baik	81 - 100
			Kurang	0 - 80

Sumber: Dokumen standart *monitoring* PT Orela Shipyard.

2.5 Visualisasi

Menurut Frey (2008) dalam bukunya menyebutkan bahwa sebuah visualisasi yang tepat adalah semacam narasi, memberikan jawaban yang jelas atas pertanyaan tanpa rincian asing. Dengan berfokus pada tujuan awal dari pertanyaan, Anda dapat menghilangkan rincian seperti itu karena pertanyaan itu memberikan acuan untuk apa yang diperlukan dan apa yang tidak diperlukan.

2.6 *Dashboard*

Menurut Few (2006) *dashboard* adalah sebuah tampilan visual dari informasi terpenting yang dibutuhkan untuk mencapai satu atau lebih tujuan, digabungkan dan diatur pada sebuah layar, menjadi informasi yang dibutuhkan dan dapat dilihat secara sekilas. *Dashboard* merupakan tampilan pada satu monitor komputer penuh yang berisi informasi yang bersifat kritis, agar kita dapat mengetahui hal-hal yang perlu diketahui, dan biasanya kombinasi teks dan grafik, tetapi lebih ditekankan pada grafik.

2.6.1 Tujuan Penggunaan *Dashboard*

Menurut Eckerson (2006) menjelaskan bahwa ada beberapa tujuan dari penggunaan *dashboard*, sebagai berikut:

3. Mengkomunikasikan strategi.
4. Memonitor dan menyesuaikan pelaksanaan strategi.
5. Menyampaikan wawasan dan informasi ke semua pihak.

2.6.2 Jenis *Dashboard*

Menurut Hariyanti (2008) dalam tulisannya menyebutkan pendapat dari Eckerson dan Few bahwa *dashboard* bisa dikelompokkan sesuai dengan level manajemen yang didukungnya, sebagai berikut:

1. *Strategic dashboard*.
 - a. Mendukung manajemen level strategis.
 - b. Informasi untuk membuat keputusan bisnis, memprediksi peluang, dan memberikan arahan pencapaian strategis.
 - c. Fokus pada pengukuran kinerja *high-level* dan pencapaian tujuan strategis organisasi.

- d. Mengadopsi konsep *balance score card*.
- e. Informasi yang diberikan tidak terlalu detail.
- f. Konten informasi tidak terlalu banyak dan disajikan secara ringkas.
- g. Informasi disajikan dengan mekanisme yang sederhana, melalui tampilan yang *unidirectional*.
- h. Tidak didesain untuk berinteraksi dalam melakukan analisis yang lebih detail.
- i. Tidak membutuhkan data *real time*.

2. *Tactical dashboard*.

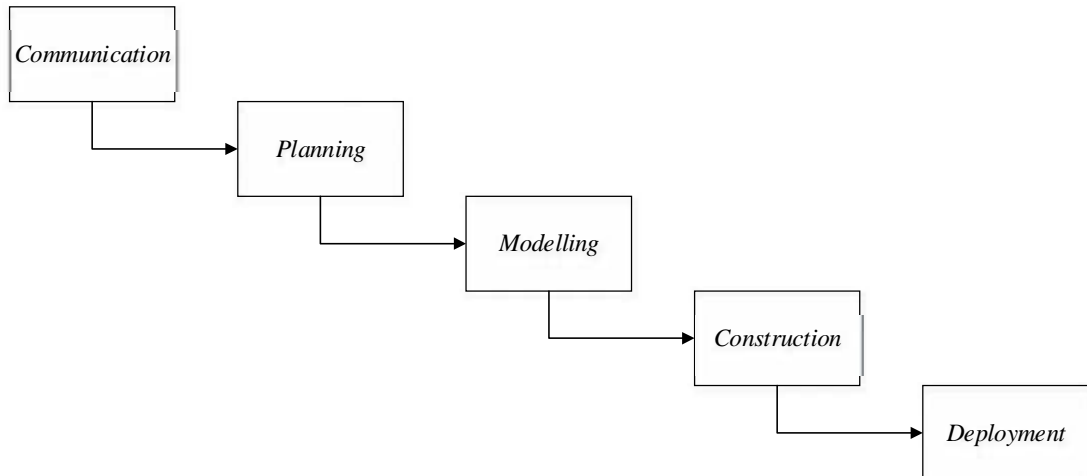
- a. Mendukung manajemen *tactical*.
- b. Memberikan informasi yang diperlukan oleh analisis untuk mengetahui penyebab dan kejadian.
- c. Fokus pada analisis untuk menemukan penyebab dari suatu kondisi atau kejadian tertentu.
- d. Dengan fungsi *drill down* dan navigasi yang baik.
- e. Memiliki konten informasi yang lebih banyak (analisis perbandingan, pola/tren, dan evaluasi kerja).
- f. Menggunakan media penyajian yang “cerdas” yang memungkinkan pengguna melakukan analisis terhadap data yang kompleks.
- g. Didesain untuk berinteraksi dengan data.
- h. Tidak memerlukan data *real time*.

3. *Operational dashboard.*

- a. Mendukung manajemen level operasional.
- b. Memberikan informasi tentang aktivitas yang sedang terjadi, beserta perubahannya secara *real time* untuk memberikan kewaspadaan terhadap hal-hal yang perlu direspon secara cepat.
- c. Fokus pada *monitoring* aktifitas dan kejadian yang berubah secara konstan.
- d. Informasi disajikan spesifik, tingkat kedetailan yang cukup dalam.
- e. Media penyajian yang sederhana.
- f. *Alert* disajikan dengan cara yang mudah dipahami dan mampu menarik perhatian pengguna.
- g. Bersifat dinamis, sehingga memerlukan data *real time*.
- h. Didesain untuk berinteraksi dengan data, untuk mendapatkan informasi yang lebih detail, maupun informasi pada level lebih atas (*high level data*).

3.7 *Software Development Life Cycle (SDLC)*

Menurut Pressman (2010) didalam *software development life cycle* terdapat beberapa model diantaranya adalah model *waterfall*, terkadang disebut sebagai siklus hidup klasik, menunjukkan sistematis, pendekatan sekuensial untuk penyebaran perangkat lunak yang dimulai dengan spesifikasi permintaan pelanggan dan berlangsung melalui perencanaan, pemodelan, *construction* dan *deployment* yang berakhir pada dukungan yang berkelanjutan dari terselesainya *software*.



Gambar 2.1 SDLC dengan metode *waterfall* (Pressman, 2010)

1. *Communication*.

Pada tahap ini merupakan tahap awal dan tahap pengumpulan data. Dalam tahap ini terbagi menjadi tiga bagian yaitu observasi, studi literature, dan wawancara. Observasi yaitu terjun langsung ke lokasi untuk mengamati masalah yang terjadi. Studi literature adalah mencari referensi buku-buku untuk menyelesaikan masalah yang ada. Wawancara merupakan proses tanya jawab untuk mendapatkan jawaban untuk mempermudah penyelesaian masalah.

2. *Planning*.

Tahap lanjutan dari *communication*, pada tahap ini melakukan proses analisis sistem. Analisis sistem dilakukan setelah melakukan tahap *communication*. Keluaran yang dihasilkan oleh analisis sistem adalah *user requirements*, dan *functional requirements*. *user requirements* merupakan kebutuhan apa saja yang diinginkan oleh pengguna atau *user*. Sedangkan, *functional requirements* adalah fungsi-fungsi yang akan digunakan pada aplikasi yang akan dibangun dan tahapan setelah *user requirements* telah disusun.

3. *Modelling.*

Pada tahap ini menjelaskan tentang syarat-syarat yang dibutuhkan untuk merancang sebuah aplikasi yang dapat diperkirakan sebelum masuk pada tahap *construction*. Tahap ini terbagi menjadi tiga bagian, yaitu data *modelling*, proses *modelling*, dan *user interface*. Ketiga bagian tersebut merupakan dasar untuk pembuatan aplikasi yang akan dilakukan pada tahap berikutnya.

4. *Construction.*

Tahap ini terbagi menjadi dua bagian yaitu *coding*, dan *testing*. *Coding* merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. Pada tahap *coding* merupakan tahapan secara nyata dalam mengerjakan suatu aplikasi. Setelah proses *coding* selesai, maka tahap selanjutnya adalah *testing*. *Testing* bertujuan untuk menemukan kesalahan-kesalahan yang terjadi pada aplikasi yang telah dibangun dan kemudian dilakukan perbaikan *coding* agar aplikasi dapat digunakan sesuai dengan fungsinya.

5. *Deployment.*

Tahap ini merupakan tahap terakhir untuk pembuatan aplikasi. Setelah melakukan analisis, desain, *coding*, dan *testing*. Maka aplikasi yang telah dibangun sudah siap untuk digunakan oleh pengguna atau *user*. Selanjutnya dilakukan pemantauan secara berkala terhadap aplikasi yang dibangun untuk menghindari kesalahan-kesalahan yang tidak diharapkan.

4.8 Database

Menurut Marlinda (2004), *database* adalah suatu susunan/kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir/dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya. Penyusunan satu *database* digunakan untuk mengatasi masalah-masalah pada penyusunan data yaitu redundansi dan inkonsistensi data, kesulitan pengaksesan data, isolasi data untuk standarisasi, banyak pemakai (*multiple user*), masalah keamanan (*security*), masalah kesatuan (*integration*), dan masalah kebebasan data (*data independence*).

