



BAB II

DASAR TEORI

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian sebelumnya telah dilakukan pada tahun 2018 dengan judul Monitoring Kemajuan Pengerjaan Proyek Belt Conveyor Plant 14 Hambalang Berbasis Web. Sistem ini bertujuan untuk menghasilkan sistem informasi untuk melakukan monitoring kemajuan proyek berbasis web dan tersedia sistem pelaporan kemajuan proyek yang terintegrasi sehingga memudahkan perusahaan dalam memonitoring proyek secara *online*. Sistem berbasis web ini dibangun dengan menggunakan PHP dan My SQL. Dari hasil penelitian ini, didapatkan sistem informasi monitoring yang dibuat memudahkan pelaporan proyek di lapangan yang dilakukan secara online sehingga perusahaan dapat memonitor kemajuan proyek secara kontinyu dan dapat digunakan untuk menilai perkembangan pekerjaan proyek yang dibandingkan dengan perencanaan proyek sehingga dapat mengambil keputusan untuk melakukan perbaikan[2].

Penelitian lainnya yang sejenis telah dilakukan pada tahun 2020 dengan judul Dinas PUPR Kabupaten Probolinggo dengan mengembangkan sebuah Aplikasi Monitoring Progres Pekerjaan Proyek Di Bidang Bina Marga. Penelitian ini bertujuan menghasilkan aplikasi untuk melakukan monitoring progres pekerjaan proyek berbasis web. Model yang digunakan pada pengembangan sistem ini yaitu *Waterfall*, dengan memanfaatkan PHP dan MySQL. Hasil penelitian menyatakan bahwa aplikasi monitoring progres pekerjaan proyek yang memudahkan konsultan pengawas, staf dan pimpinan dalam memantau progres pekerjaan yang akan dibandingkan dengan progres rencana sehingga dapat mengambil keputusan dalam melakukan perbaikan kedepannya[3].

Penelitian lain yang sejenis mengenai Sistem Informasi Monitoring Kegiatan Proyek Pemancar Sinyal BTS dilakukan pada PT. Swatama Mega Teknik yang berbasis Web dilakukan pada ta. Penelitian ini bertujuan untuk membantu manager proyek dalam memonitor pelaksanaan aktivitas pada tiap harinya dengan efektif, baik pada pembuatan data untuk proyek, pada pelaporan kegiatan harian proyek, maupun saat memantau suatu proyek. Penulisan penelitian ini menggunakan pengembangan metode *Web Development Life Cycle* (WDLC). Hasil dari penelitian ini menyatakan bahwa dengan menggunakan sistem informasi monitoring yang dibuat, maka data tidak lagi disimpan dalam bentuk fisik, melainkan

menggunakan media penyimpanan data digital, sehingga diharapkan dapat mengurangi hilang dan rusaknya data[4].

Adapun perbedaan sistem dari penelitian sebelumnya yang telah dibuat dengan penelitian yang akan penulis kembangkan yaitu pada penelitian sebelumnya sistem lebih mengarah pada monitoring proyek untuk mendapatkan data laporan saja. Sedangkan penelitian yang dilakukan oleh penulis digunakan juga sebagai landasan untuk mengeluarkan surat penagihan pembayaran proyek.

2.2 Landasan Teori

Dalam menunjang penelitian ini, maka diperlukannya teori-teori mendasar. Teori-teori yang digunakan dalam penelitian ini adalah:

2.2.1 Sistem Informasi

Sistem informasi berisi gabungan dari perangkat keras dan perangkat lunak komputer, dan juga perangkat manusia, yang memanfaatkan perangkat keras dan perangkat lunak untuk memproses data[5]. Sedangkan pendapat lain juga menyatakan bahwa sistem informasi merupakan suatu sistem yang berada pada suatu kumpulan yang menggabungkan antara kebutuhan manajemen transaksi rutin, membantu kegiatan operasional, administratif, dan strategis organisasi serta menyediakan keluaran yaitu laporan-laporan yang diperlukan[6].

Terdapat tiga kegiatan mendasar pada sistem informasi, yaitu kegiatan masukan atau *input*, pemrosesan atau *processing*, dan keluaran atau *output*. Tiga kegiatan ini yang akan menghasilkan informasi yang diperlukan oleh organisasi dalam pengambilan keputusan, pengendalian operasi, analisis permasalahan, dan menciptakan produk atau jasa baru.

2.2.2 Rekayasa Perangkat Lunak

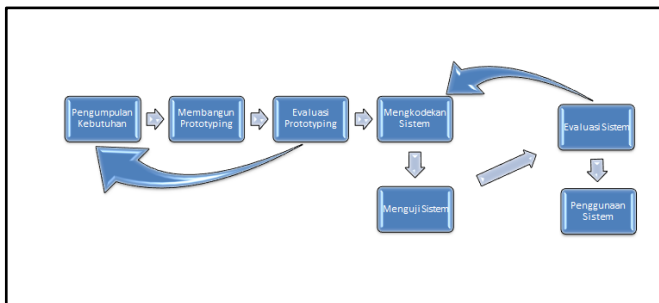
Rekayasa perangkat lunak berperan penting pada pembangunan perangkat lunak di mana fungsinya adalah untuk melakukan pengembangan, pemeliharaan, dan pembangunan kembali dengan menerapkan prinsip-prinsip rekayasa untuk memperoleh *software* yang sesuai dengan kebutuhan dan bisa bekerja dengan efektif dan efisien. Rekayasa perangkat lunak merupakan disiplin ilmu yang berhubungan dengan semua bagian dari produk *software* dari tingkatan awal sampai pada pemeliharaan *software* setelah produksi atau prosedur yang terpadu dan menyeluruh dari semua bagian, sejak sebelum *software* dibangun hingga selesai dibangun dan bahkan hingga pada tingkat semua bagian[7].

1. Metode Pengembangan Sistem

Pada rekayasa perangkat lunak, terdapat beberapa metode yang digunakan dalam pengembangan sistem, salah satu metode yang digunakan dalam pengembangan sistem adalah metode *prototype*. Metode *prototype* ini diawali dengan dikumpulkannya kebutuhan pengguna dari sistem yang akan dikembangkan, setelah itu dibuat sebuah rancangan kilat, yang kemudian dilakukan evaluasi oleh klien.

Metode *prototype* merupakan metode pengembangan *software* yang memungkinkan terdapat interaksi antara *software developer* dengan pemakai sistem, sehingga bisa menangani ketidaksesuaian antara *developer* dengan pemakai[8]. *Prototype* bisa dipakai sebagai penyambung ketidakpahaman klien tentang hal teknis dan membantu menjelaskan rincian kebutuhan yang dibutuhkan klien kepada *developer*[9].

Metode *prototype* adalah suatu metode yang memungkinkan pengembang membuat suatu model perangkat lunak, metode *prototype* ini bisa diterapkan apabila pihak klien tidak dapat menjelaskan informasi secara maksimal perihal kebutuhan yang diinginkannya[10]. Dengan berdasar pada pendapat para ahli, dapat disimpulkan bahwa dengan menggunakan metode *prototype* ini, memungkinkan klien untuk lebih mengetahui mengenai hal teknis sistem dan bagi pengembang untuk lebih memahami kebutuhan pengguna secara rinci. Sehingga dengan menggunakan metode *prototype* ini dapat menangani ketidaksesuaian yang terjadi antara pengembang dengan klien.



Gambar 2. 1 Metode Prototype Menurut Pressman

Berikut adalah tahap-tahap dalam metode *prototyping*[11]:

1. Pengumpulan kebutuhan

Pada tahap ini pelanggan dan pengembang bersama-sama mendefinisikan format dan kebutuhan perangkat lunak secara keseluruhan, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.

2. Membangun *prototyping*

Yang dilakukan pada tahap ini adalah membuat desain sementara yang berpusat pada penyajian kepada pelanggan, misalnya; dengan membuat contoh *input* dan *output*.

3. Evaluasi *prototyping*

Jika *prototype* yang sudah dibangun sesuai dengan keinginan pelanggan maka dilakukan evaluasi. Jika sudah sesuai akan dilanjutkan ketahap pengkodean (*coding*), namun jika belum sesuai maka akan diperbaiki dengan mengulang kembali tahap 1, tahap 2, dan tahap 3.

4. Mengodekan sistem

Tahap ini lakukan jika *prototype* yang dibuat sudah diterima dan disepakati yang dilanjutkan dengan pengkodean dengan bahasa pemrograman yang sesuai

5. Menguji sistem

Sistem yang telah selesai dibuat kemudian akan dilakukan pengujian dengan menggunakan metode *black-box*. Tujuannya adalah untuk mengetahui apakah sistem yang dibuat sudah dapat berfungsi dengan baik dan benar sesuai yang diharapkan.

6. Evaluasi sistem

Tahap ini dilakukan untuk mengetahui apakah sistem yang dibangun telah sesuai dengan yang diharapkan. Jika sudah memenuhi dapat dilakukan tahap selanjutnya, namun jika tidak maka perlu dilakukan pengulangan pada tahap 4 dan tahap 5.

7. Penggunaan sistem

Sistem yang dibangun siap digunakan oleh pelanggan setelah tahap pengujian berhasil dan diterima oleh pelanggan.

2. Pengujian

Pengujian atau *testing* merupakan elemen kritis dari jaminan kualitas perangkat lunak dan merupakan bagian yang tidak terpisahkan dari siklus hidup pengembangan *software* seperti halnya analisis, desain, dan pengkodean[12]. Dalam rekayasa perangkat lunak terdapat beberapa metode untuk pengujian sistem, salah satu metode yang digunakan dalam

pengujian sistem yaitu metode *black-box*. Metode *black-box* ini memiliki tujuan untuk menguji fungsi dari sebuah program yang dikembangkan, apakah sudah sesuai kebutuhan dan spesifikasi.

Black-box testing adalah teknik pengujian *software* yang berfokus pada spesifikasi fungsional dari perangkat lunak. *Black-box* bekerja dengan memfokuskan perhatiannya pada informasi domain, sehingga mengabaikan struktur kontrol sistem. Metode pengujian ini memungkinkan bagi pengembang perangkat lunak untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program[13].

Keuntungan penggunaan metode pengujian *black-box* adalah[13]:

- (1) Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu;
- (2) Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan;
- (3) *Programmer* dan *tester* keduanya saling bergantung satu sama lain.


3. Alat Bantu

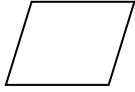
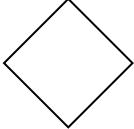
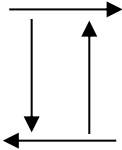


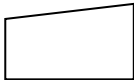
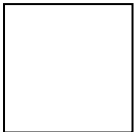
a. *Flowchart* Sistem

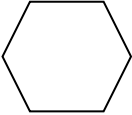
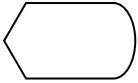
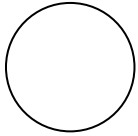
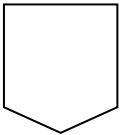
Flowchart adalah sebuah skema dengan simbol-simbol tertentu yang menjabarkan rangkaian dari proses dengan detail serta jalinan antara suatu proses (instruksi) dengan proses yang lainnya dalam sebuah program. Diagram alur bisa menggambarkan dengan jelas, jalannya sebuah algoritma yakni bagaimana menjalankan sebuah rangkaian aktivitas dengan logis dan sistematis[14].

Seperti yang telah disebutkan, *flowchart* memiliki simbol-simbol tertentu untuk menjabarkan rangkaian. Berikut adalah simbol-simbol pada *flowchart* seperti pada Tabel 2.1.

Tabel 2. 1 Simbol-Simbol *Flowchart* Sistem

No.	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Memulai dan mengakhiri suatu program.

2.		<i>Input – Output</i>	Memasukkan data maupun menunjukkan hasil dari suatu <i>process</i> tanpa tergantung dengan jenis peralatannya.
3.		<i>Decision</i>	Memilih proses berdasarkan kondisi yang ada.
4.		<i>Flow</i>	Menghubungkan antara simbol satu dengan simbol yang lain atau menyatakan jalannya arus dalam suatu proses. simbol arus ini sering disebut dengan <i>connecting line</i> .
5.		<i>Document</i>	Merupakan simbol untuk data yang terbentuk informasi.
6.		<i>Manual Operation</i>	Menunjukkan pengolahan yang tidak dilakukan oleh komputer/pc.
7.		<i>Manual Input</i>	Memasukkan data secara manual <i>on-line keyboard</i> .
8.		<i>Process</i>	Simbol yang menyatakan suatu proses yang dilakukan komputer.

9.		<i>Preparation</i>	Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.
10.		<i>Display</i>	Simbol yang menyatakan peralatan <i>output</i> yang digunakan.
11.		<i>On-Page Refenece</i>	Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang sama.
12.		<i>Off-Page Reference</i>	Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang berbeda.


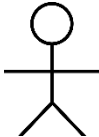

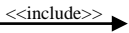
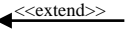

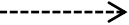
b. *Unified Modeling Language*

Unified Modeling Language (UML) sebuah bahasa yang berdasarkan gambar untuk memvisualisasikan, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis objek [15].

1) *Use Case Diagram*

Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem ini berinteraksi dengan dunia luar, misalnya menyusun sebuah daftar layanan kesehatan. *Use Case Diagram* dapat digunakan untuk memperoleh kebutuhan sistem dan memahami bagaimana sistem seharusnya bekerja. Simbol-simbol *Use Case Diagram* bisa dilihat pada Tabel 2.2. [15]

Tabel 2. 2 Simbol-Simbol *Use Case Diagram*

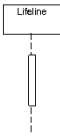
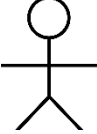
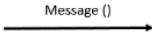
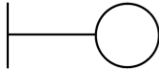

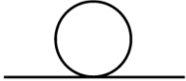
No.	Simbol	Nama	Keterangan
1.		<i>Use case</i>	Deskripsi dari urutan aksi – aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
2.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
3.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>usecase</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas
7.		<i>Generalisasi</i>	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .

2) *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek berupa pesan (*message*) yang digambarkan terhadap waktu. *Sequence diagram* terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Message* digambarkan sebagai garis berpanah dari satu objek

ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metode dari *class*. *Activation* bar menunjukkan lamanya eksekusi sebuah proses. Berikut adalah simbol-simbol yang ada pada *sequence diagram*, sesuai pada Tabel 2.3.[15]

Tabel 2. 3 Simbol-Simbol *Sequence Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2.		<i>Actor</i>	Menggambarkan <i>user</i> atau pengguna.
3.		<i>Message</i>	Spesifikasi dari komunikasi antar <i>objek</i> yang memuat informasi tentang aktivitas yang terjadi.
4.		<i>Boundary</i>	Menggambarkan sebuah <i>form</i> .
5.		<i>Control Class</i>	Menghubungkan <i>boundary</i> dengan tabel.
6.		<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan

2.2.3 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan

terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Pemrograman berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek dan pengujian berorientasi objek[16].

2.2.4 Basis Data

Database merupakan salah satu komponen utama yang sangat penting dalam pembuatan maupun pengembangan sistem informasi. *Database* perlu dibuat sebaik mungkin agar pembuatan sistem berjalan dengan baik. *Database* ini bukan hanya sebatas tempat penyimpanan dari informasi dan data pada sistem, *database* juga berfungsi untuk mengelompokkan data sehingga indentifikasi data dapat dilakukan dengan lebih mudah.

Sistem basis data ialah sebuah sistem di mana sistem ini mengorganisasikan dan mengoperasikan *record-record* dengan komputer untuk menyimpan atau merekam dan juga menjaga data. Sistem basis data bermanfaat secara lengkap dari sebuah organisasi atau perusahaan sehingga dapat memberikan informasi yang optimal yang dibutuhkan oleh pengguna untuk proses pengambilan keputusan[17].

Terdapat penjelasan lain mengenai pengertian *database*, yaitu merupakan sistem yang digunakan sebagai tempat penyimpanan informasi yang sudah melalui proses pengolahan. Terstruktur dan saling berkorelasi satu dengan yang lainnya, dan memiliki kata kunci yang digunakan sebagai label masing-masing file yang disimpan[18].

Dalam memasukkan dan mengambil suatu data dari dan ke dalam media penyimpanan data membutuhkan bantuan perangkat lunak yang disebut dengan Sistem Manajemen Basis Data (DBMS). DBMS merupakan perangkat lunak yang dapat digunakan oleh suatu organisasi untuk memusatkan data, mengelola secara efisien, dan memberikan akses ke data yang disimpan oleh program aplikasi[19]. DBMS ini menyediakan fasilitas sebagai berikut:


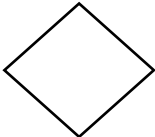
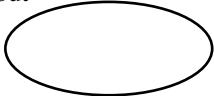
- a. *Data Definition Language* (DDL) dapat digunakan oleh pengguna untuk menguraikan *database*, contohnya seperti menentukan detail tipe dan batasan data yang akan disimpan dalam *database*.




- b. *Data Manipulation Language* (DML) dapat digunakan oleh pengguna untuk memanipulasi data, seperti contohnya menambahkan data, menghapus data, dan mengambil data dari *database*.
- c. Menyediakan akses terkendali pada *database*, seperti sistem keamanan (*security system*), sistem integritas (*integrity system*), sistem pengaturan (*concurrency system*), sistem pengendali pemulihan (*recovery control system*), dan *user accessible catalog*.

1. Entity Relationship Diagram (ERD)

Dalam rekayasa perangkat lunak diperlukan adanya *Entity Relationship Diagram* (ERD). ERD merupakan model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model. Di dalam hubungan ini tersebut dinyatakan yang utama dari ERD adalah menunjuk objek data (*Entity*) dan hubungan (*Relationship*), yang ada pada *Entity* berikutnya. Simbol-simbol ERD dapat di lihat pada Tabel 2.4[20].

Tabel 2. 4 Simbol-Simbol ERD

No	Simbol	Fungsi
1.	Entitas 	Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer.
2.	Relasi 	Relasi yang menghubungkan antar entitas. Biasanya diawali dengan kata kerja.
3.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.

4.	Atribut Kunci Primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan.
5.	Atribut Multinilai 	<i>Field</i> yang butuh disimpan dalam entitas yang dapat memiliki nilai lebih dari satu.
6.	Alur 	Alur memiliki fungsi untuk menghubungkan atribut dengan entitas dan entitas dengan relasi.

2. MYSQL

MySQL merupakan sebuah perangkat lunak sistem manajemen basis data SQL atau yang biasa dikenal dengan *Database Management System* (DBMS), basis data ini *multithread* atau *multi-user*. SQL pertama kali didefinisikan oleh American National Standard Institute (ANSI) pada tahun 1986. Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai Bahasa dasar untuk mengakses *database*-nya sehingga mudah untuk digunakan, kinerja *query* cepat, dan mencukupi untuk kebutuhan *database* perusahaan-perusahaan skala menengah-kecil. MySQL juga bersifat *open source General Public License* (GPL) mulai versi 3.23, pada bulan Juni 2000[21].

MySQL merupakan sistem manajemen basis data yang bersifat *relational*, artinya data-data yang dikelola dalam basis data akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan menjadi jauh lebih cepat. MySQL dapat digunakan untuk mengelola basis data mulai dari yang kecil sampai dengan yang sangat besar. MySQL juga dapat menjalankan perintah-perintah *Structured Query Language* (SQL) untuk mengelola basis data-basis data yang ada di dalamnya.

2.2.5 Manajemen Proyek

Proyek merupakan kegiatan yang berlangsung sementara, proyek dilaksanakan untuk memenuhi tujuan yang sudah dirancang sebelumnya,

dan menghasilkan sesuatu sesuai rencana. Proyek ini kegiatan yang rumit dan tidak rutin, terbatas oleh waktu, biaya, sumber daya, dan spesifikasi kinerja untuk memenuhi kebutuhan konsumen[22].

Tujuan dilaksanakannya proyek yaitu untuk melaksanakan ide yang ada dari seseorang, badan, atau organisasi. Jadi sebuah proyek memiliki masing-masing sifat dan ciri khasnya, di mana sifat dan ciri khasnya ini tidak sama antara satu dengan yang lainnya. Sebuah proyek umumnya bersifat: ciri khasnya kuat, siklus aktivitasnya khas, peranan manajer proyek dominan, dan terdapat usaha pendekatan sistematis yang menguntungkan atau positif[23].

Dalam kegiatan manajemen proyek, pasti dilaksanakan kegiatan penagihan pembayaran proyek. Penagihan adalah suatu tindakan melakukan tagihan, memberi informasi, dan meningkatkan kepada pihak-pihak tertagih bahwa pihak tersebut memiliki kewajiban untuk membayar tagihannya kepada pihak penagih. Dalam akuntansi penagihan biasanya digunakan untuk menunjukkan klaim yang akan dilunasi dengan uang. Dalam Kamus Besar Bahasa Indonesia (KBBI) penagihan merupakan proses, cara, perbuatan menagih, permintaan agar membayar hutang[24].

Pembayaran adalah bergantinya hak pemilik atas sejumlah uang atau dan dari pembayar kepada penerimanya, baik secara langsung maupun melalui media jasa-jasa perbankan[25]. Dalam kegiatan ekonomi, pembayaran melibatkan seperangkat peraturan, organisasi, dan tata cara yang digunakan untuk melakukan pengalihan dana, dengan tujuan untuk melengkapi suatu kewajiban yang timbul.

2.2.6 Sistem Monitoring

Monitoring diartikan sebagai cara memeriksa apakah dalam suatu kegiatan berjalan sesuai konsep awal, apakah timbul masalah, atau berjalan sesuai pola tertentu. Jika adanya masalah yang terjadi dapat segera dipecahkan, tujuannya adalah untuk membuat kemajuan[26].

Sistem monitoring dapat mendukung proses kontrol aktivitas bisnis, merekap aktivitas bisnis, dan menghasilkan laporan. Pemantauan dapat dilakukan pada masing-masing aktivitas, seperti dari rencana atau setelah menyelesaikan bagian tertentu dari pekerjaan[27].

Terdapat pendapat bahwa dalam kegiatan monitoring ini terdapat empat fungsi, di antaranya yaitu[28]:

1. Ketaatan. Peranan dari *stakeholder*, seperti admin, staf, dan keseluruhan yang terlibat harus sudah mengaplikasikan standarisasi prosedur dalam berlangsungnya kegiatan.

2. Pemeriksaan. Aktivitas monitoring yang menentukan target untuk menjadi indikator dalam perolehan kinerja.
3. Laporan. Jalannya monitoring akan membuahkan hasil suatu informasi untuk menerapkan keputusan yang telah ditetapkan pada kurun waktu ataupun periode tertentu.
4. Penjelasan. Urutan monitoring akan membuahkan hasil informasi dan kemudian akan merinci mengenai ketetapan atas semua kemungkinan yang terjadi antara yang direncanakan dan hasil yang dilaksanakan (sinkronisasi).

~Halaman Ini Sengaja Dikosongkan~