BAB II

LANDASAN TEORI

2.1 Monitoring

Monitoring adalah pengawasan yang berarti proses pengamatan, pemeriksaan, pengendalian dan pengoreksian dari seluruh kegiatan organisasi. George R. Tery (2006) mengartikan pengawasan adalah mendeterminasi apa yang telah dilaksanakan, maksudnya mengevaluasi prestasi kerja dan apabila perlu, menerapkan tidankan-tindakan korektif sehingga hasil pekerjaan sesuai dengan rencana yang telah ditetapkan.

2.1.1 Contoh Aplikasi Monitoring

1. Aplikasi Monitoring Jaringan

Gambar 2.1 Aplikasi Monitoring Jaringan

Berdasarkan gambar diatas menjelaskan tentang aplikasi monitoring jaringan. Semakin meningkatnya ukuran dan jumlah perangkat jaringan maka akan semakin kompleks masalah yang ada pada jaringan tersebut. Hal tersebut

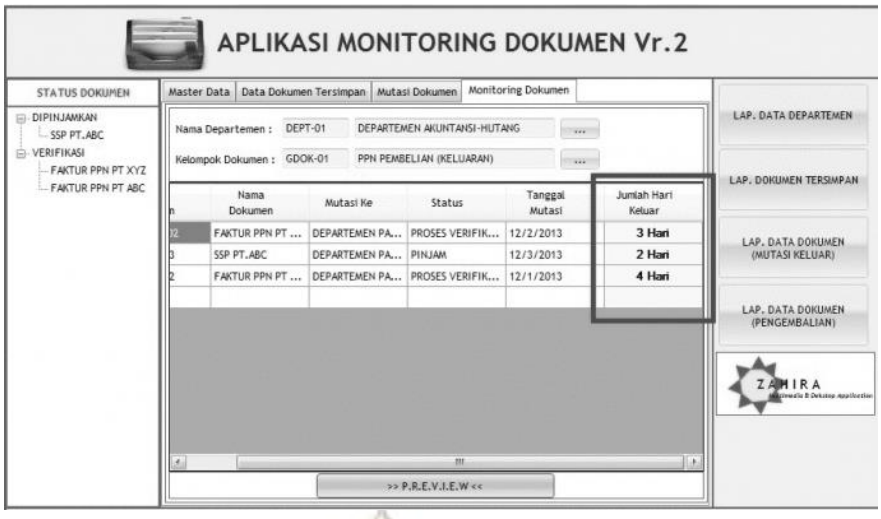
tentunya membutuhkan pengawasan secara terus menerus terhadap seluruh perangkat jaringan untuk menjamin ketersediaan atau availability layanan.

2. Aplikasi Monitoring Curah Hujan

Gambar 2.2 Aplikasi Monitoring Curah Hujan

Berdasarkan gambar 2.2 Aplikasi Monitoring Curah Hujan merupakan aplikasi perkebunan sawit yang berfungsi untuk melakukan monitoring curah hujan baik dalam bentuk data hari hujan, lama waktu hujan dan jumlah curah hujan. Lama waktu hujan pada Aplikasi Monitoring Curah Hujan dinyatakan dalam satuan menit, sedangkan jumlah curah hujan dalam satuan Milimeter.

2.Aplikasi Monitoring Dokumen



Gambar 2.3 Aplikasi Monitoring Dokumen

Berdasarkan gambar 2.3 menjelaskan dalam kegiatan perusahaan untuk memperoleh pendapatan dibutuhkan biaya yang digunakan untuk mendukung kegiatan operasional perusahaan. Penyusunan biaya operasional terwujud dalam *dokumen* anggaran yang disusun sebelum tahun operasional berikutnya berjalan dan disesuaikan dengan target serta potensi yang dimiliki.

2.2 Tujuan Monitoring

Monitoring dilakukan untuk mendapatkan hasil yang baik, dengan tujuan sebagai berikut :

a. Mengkaji apakah kegiatan-kegiatan yang dilaksanakan telah sesuai dengan rencana
b. Mengidentifikasi masalah yang timbul agar langsung dapat diatasi
c. Melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan kegiatan.
d. Mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan,

e. Menyesuaikan kegiatan dengan lingkungan yang berubah, tanpa menyimpang dari tujuan. (Sujamto, 1986 : 81-82)

2.3 Jenis Monitoring (Pengawasan)

Terdapat jenis-jenis monitoring diantaranya :

1. Pengawasan Ekstern dan Intern

a. Pengawasan Ekstern

Pengawasan ektern atau pengawasan dari luar, yakni pengawasan yang menjadi subyek pengawas adalah pihak luar dari organisasi obyek yang diawasi.

b. Pengawasan Intern

Pengawasan intern merupakan pengawasan yang dilakukan dari dalam organisasi yang bersangkutan.

2. Pengawasan Preventif, Represif dan Umum

a. Pengawasan Preventif

Pengawasan Preventif adalah pengawasan yang dilakukan sebelum pelaksanaan, yakni pengawasan yang dilakukan terhadap sesuatu yang bersifat rencana.

b. Pengawasan Represif

Pengawasan Represif merupakan pengawasan yang dilakukan setelah pekerjaan atau kegiatan dilaksanakan.

c. Pengawasan Umum

Pengawasan umum adalah pengawasan terhadap seluruh aspek pelaksanaan tugas pokok organisasi.

3. Pengawasan Langsung dan Pengawasan Tidak Langsung

a. Pengawasan Langsung

Pengawasan Langsung adalah pengawasan yang dilakukan dengan cara mendatangi dan melakukan pemeriksaan di tempat (*on the spot*) terhadap obyek yang diawasi.

b. Pengawasan tidak langsung

Pengawasan Tidak Langsung merupakan pengawasan yang dilakukan tanpa mendatangi tempat pelaksanaan pekerjaan atau obyek yang diawasi atau pengawasan yang dilakukan dari jarak jauh yaitu dari belakang meja.

4. Pengawasan Formal dan Informal

a. Pengawasan Formal

Pengawasan Formal adalah pengawasan yang dilakukan oleh instansi/pejabat yang berwenang (resmi) baik yang berifat intern dan ekstern.

b. Pengawasan Informal

Pengawasan Informal yakni pengawasan yang dilakukan oleh masyarakat atau *social control,* misalnya surat pengaduan masyarakat melalui media massa atau melalui badan perwakilan rakyat. (Sujamto, 1986 : 81-82)

2.4 Tipe Monitoring

Tipe-tipe monitoring diantaranya yaitu:

A. Monitoring Rutin

Kegiatan mengkompilasi informasi secara reguler berdasarkan sejumlah indikator kunci. Monitoring rutin dapat dipergunakan untuk mengidentifikasi penerapan program dengan atau tanpa perencanaan.

B. Monitoring Jangka Pendek

Dilakukan untuk jangka waktu tertentu dan biasanya diperuntukkan bagi aktifitas yang spesifik. Seringkali bila aktifitas atau proses-proses baru diterapkan, manajer ingin mengetahui, apakah sudah diterapkan sesuai rencana dan apakah sesuai dengan keluaran yang diinginkan.

2.5 Langkah – Langkah dalam Monitoring

Menurut Sujamto, (1986) Terdapat beberapa langkah-langkah dalam melakukan monitoring diantaranya yaitu :

1. Perencanaan
    a) Merancang sistem monitoring yang spesifik: apa yang akan dimonitor, mengapa dan untuk siapa (user).
    b) Menentukan scope monitoring: luasnya area (RS, puskesmas non TT)? apakah bersifat klinis atau service? Siapa yang terlibat? Berapa lama monitoring akan dilakukan?

c) Memilih dan menentukan indikator menentukan batasan sasaran kelompok misalnya kelompok anak dibawah 2th, 5th atau antara 12-60 bln ?

d) Menentukan sumber-sumber informasi, memilih metoda pengumpulan data, seperti metoda observasi, interview petugas, rapid survey untuk cakupan atau pengobatan di rumah (home treatment).

2. Implementasi

- Memilih menentukan proses supervisi dan prosessingnya (kemana akan dikirim).
- Tabulasi data dan analisa data : membandingkan temuan atau pencapaian aktual dengan perencanaan.
- Temuan dalam monitoring: apakah ada penyimpangan, bila ada perlu diidentifikasi masalah penyebabnya.
- Menggali penyebab dan mengambil tindakan perbaikan: menggali penyebab terjadinya masalah. Rencana monitoring perlu disusun jangka pendek untuk menjamin bahwa tindakan/prosedure dilaksanakan sesuai standar (rencana) serta memberi efek sesuai dengan harapan

3. Menentukan kelanjutan monitoring

Kegiatan monitoring dirancang untuk memperoleh hasil kinerja sekarang atau jangka pendek bagi manajer atau user lainnya. Ketika program atau kegiatan rutin telah memberikan perubahan signifikan,

maka kelangsungan program kinerja memerlukan perhatian. Review secara periodik tetap diperlukan. Sistem informasi manajemen akan membantu manajer untuk mempertimbangkan kapan indikator dan frekuensi monitoring dikurangi dan pada bagian mana perlu direncanakan lagi dan dilanjutkan.

2.6 Kesehatan

Pengertian kesehatan atau sehat menurut Paune (1983) Sehat adalah fungsi efektif dari sumber-sumber perawatan diri(self care resources) yang menjamin tindakan untuk perawatan diri (self care action) merupakan pengetahuan ketrampilan dan sikap. Self care action merupakan perilaku yang sesuai dengan tujuan diperlukan untuk memperoleh , mempertahankan, dan meningkatkan fungsi psikososial dan spiritual.

Sakit Menurut Pemons adalah keadaan dimana fisik, emosional, intelektual, sosial, perkembangan atau seseorang berkurang atau terganggu, bukan hanya keadaan dimana terjadinya proses penyakit. Sakit berarti suatu keadaan yang memperlihatkan adanya keluhan dan gejala sakit secara subyektif dan obyektif, sehingga penderita butuh pengobatan untuk menjadi sehat.

Sakit adalah ganguan dalam fungsi normal individu sebagai totalitas termasuk keadaan organisme sebagai sistem biologis dan penyesuaian sosialnya.

2.7 Tunjangan

Menurut Hariandja (2005:244) mengemukakan bahwa : ”Tunjangan adalah keseluruhan balas jasa yang diterima oleh pegawai sebagai akibat dari pelaksanaan pekerjaan di organisasi dalam bentuk uang atau lainnya, yang dapat berupa gaji upah, bonus, insentif, dan tunjangan hari raya, uang makan, uang cuti, dan lain-lain”.

2.8 HTML

Menurut Fathul Wahid (2005:135) Hypertext Markup Language adalah sekumpulan perintah yang terformat yang digunakan untuk membuat halaman dokumen web. Ketika membuka sebuah halaman web, maka browser akan menginterpretasikan perintah HTML pada halaman tersebut dan ke dalam teks dan grafik.

2.9 CSS

CSS merupakan kependekan *Cascading Style Sheet* yang berfungsi untuk mengatur tampilan dengan kemampuan jauh lebih baik dari *tag* maupun atribut standar HTML.

CSS sebenarnya adalah suatu kumpulan atribut untuk fungsi format tampilan dan mdapat digunakan untuk mengontrol tampilan banyak dokumen secara bersamaan.

Keuntungan menggunakan CSS yaitu jika ingin mengubah dokumen, maka tidak perlu mengubah satu persatu.

Penggunaan CSS ada dua cara yaitu dengan menyispkan kode CSS langsung dalam kode HTML atau simpan menjadi file tersendiri berekstensi *.css. Dengan meyimpan sebagai file tersendiri akan

memudahkan untuk mengontrol tampilan dalam banyak dokumen secara langsung. CSS mendapat dukungan penuh pada browser versi 4 dan pada versi sebelumnya, hanya *Internet Explorer* yang masih mampu mengenal CSS. Tampilan CSS dapat berbeda jika ditampilkan pada menu browser yang berbeda.(Diar Puji Oktavian, 2010)

2.10 PHP

Menurut Peranginangin (2006 : 2) PHP singkatan dari PHP *Hypertext Preprocessor* yang di gunakan sebagai script *server-side* dalam pengembangan *web* yang disisipkan pada dokumen HTML.

PHP dikatakan sebagai sebuah *server-side embedded script language* artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh server tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di server.

Ketika menggunakan PHP sebagai *server-side embedded script language* maka server akan melakukan hal-hal sebagai berikut :

a. Membaca permintaan dari *client/browser*
b. Mencari halaman/page di server.
c. Melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman/page.

Mengirim kembali halaman tersebut kepada *client* melalui *internet* atau *intranet*.

2.11 Basisdata

Menutut Bambang Hariyanto (2008), data adalah rekaman mengenai fenomena/fakta yang ada atau yang terjadi. Data pada pokonya adalah refleksi fakta yang ada. Data mengenai fakta-fakta penting organisasi harus direkam dan dikelola secara baik sehingga dapat dipakai/diakses secara efisien sehingga efektif mendukung operasi dan pengendalian organisasi. Data merupakan sumber daya penting pada manajemen modern. Untuk itu, organisasi perlu melakukan penataan dan manajemen data yang baik agara data yang dimiliki organisasi dapat berdaya guna secara maksimal.

Basisdata adalah kumpulan data (elementer) yang secara logic berkaitan dalam merepresentasikan fenomena/fakta secara terstruktur dalam domain tertentu untuk mendukung aplikasi pada sistem tertentu. Basisdata adalah kumpulan data yang saling berhubungan yang merefleksikan fakta-fakta yang terdapat di organisasi. Basisdata mendeskripsikan *state* organisasi/perusahaan/sistem. Saat satu kejadian muncul si dunia nyata mengubah *state* organisasi/perusahaan/sistem maka satu perubahan pun harus dilakukan terhdadap data yang disimpan di basisdata. Basisdata merupakan komponen utama sistem informasi karena semua informasi untuk pengambilan keputusan berasal dari data di basisdata. Pengelolaaan basisdata yang buruk dapat mengakibatkan ketidaktersediaan data penting yang digunakan untuk menghasilkan informasi yang diperlukan dalam pengambilan keputusan.

Sistem manajemen basisdata atau DBMS (*Database Management System*) adalah perangkat lunak untuk mendefinisikan, menciptakan, mengelola, dan mengendalikan pengsaksesan basisdata. Fungsi sistem manajemen basisdata saat ini yang penting adalah menyediakan basis untuk sistem informasi manjemen.

2.12 MySQL

Pendapat Sukarno (2006, hal:3) mengenai pengertian MySQL adalah merupakan perangkat lunak untuk system manajemen database (database management system). Karena sifatnya yang open source dan memiliki kamampuan menampung kapasitas yang sangat besar, maka MySQL menjadi database yang sangat popular dikalangan programmer web. Pada bulan Mei 1996, MySQL versi 1.0 berhasil dirilis namun penggunanya terbatas 4 orang saja. namun di bulan Oktober di tahun yang sama versi 3.11.0 dilepaskan ke public tapi belum bersifat open source. Bulan Juni 2000, MySQL AB mengumumkan bahwa sejak versi 3.23.19, MySQL adalah merupakan software database yang bebas berlisensi GPL atau General Public License yang open source. Mulanya MySQL hanya berjalan di system operasi linux namun pada saat MySQL versi 3.22 tahun 1998-1999 sudah tersedia diberbagai platform termasuk windows. Ini terjadi karena MySQL menjadi semakin populer dan dilirik banyak orang karena kestabilan dan kecepatan yang meningkat. Beberapa keuanggulan dari MySQL adalah:

a. Mampu menangani jutaan user dalam waktu yang bersamaan.
b. Mampu menampung lebih dari 50.000.000 record.

c. Sangat cepat mengekskusi perintah.

d. Memiliki user privilege yang mudah dan efisien.

2.13 UML

UML itu singkatan dari Unified Modelling Language. Sesuai dengan kata terakhir dari kepanjangannya, UML merupakan salah satu bentuk language atau bahasa. Menurut pencetusnya, UML didefinisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem.

Karena tergolong bahasa visual, UML lebih mengedepankan penggunaan diagram untuk menggambarkan aspek dari sistem yang sedang dimodelkan. Memahami UML itu sebagai bahasa visual itu penting, karena penekanan tersebut membedakannya dengan bahasa pemrograman yang lebih dekat ke mesin. Bahasa visual lebih dekat ke mental model pikiran kita, sehingga pemodelan menggunakan bahasa visual bisa lebih mudah dan lebih cepat dipahami dibandingkan apabila dituliskan dalam sebuah bahasa pemrograman.

UML adalah salah satu bentuk notasi atau bahasa yang sama yang digunakan oleh professional dibidang software untuk menggambarkan atau memodelkan sebuah system software. Sebelumnya ada banyak notasi atau bahasa lain untuk mencapai keperluan yang sama misalnya DFD (Data Flow Diagram). Tetapi sejak matang dan populernya teknologi pemrograman, perancangan, dan analisis berorientasi objek, UML telah menjadi de facto standard language.

Ada tiga cara dalam memakai UML dalam melakukan pemodelan system:

1. UML sebagai ketsa

   UML digambarkan dalam sketsa coretan-coretan dalam kertas atau whitboard secara tidak formal. Biasanya digunakan dalam sesi diskusi tim untuk membahas aspek tertentu dalam tahap analisis dan perancangan.

2. UML sebagai blueprint system

   Seperti diagram kelistrikan adalah blueprint dari komponen atau produk yang akan dihasilkan, UML juga bisa menggambarkan blueprint yang identik untuk sebuah system software.

3. UML sebagai bahasa pemrograman

   UML berfungsi sebagai bahasa pemrograman mencoba melakukan semuanya dengan UML sampai kepada produk jadinya. Analisis dan perancangan dilakukan dengan diagram-diagram yang ada dalam UML, sementara sebuah tool atau generator bisa menghasilkan produk akhir dari diagram-diagram ini. Diagram-diagram yang terdapat dalam UML antara lain:

   a. *use case diagram*
   b. *class diagram*
   c. *statechart diagram*
   d. *activity diagram*
   e. *sequence diagram*

f. *collaboration diagram*

g. *component diagram*

h. *deployment diagram* (Munawar, 2005).

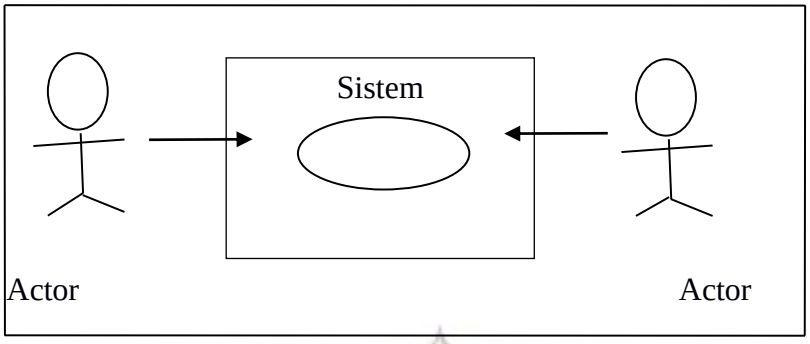
2.13.1 Use Case

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include*akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal.

Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.Sebuah *use case* juga dapat meng-*extenduse case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Diagram use case menunjukan 3 aspek dari system yaitu Actor, use case dan system/sub system boundary. Actor mewakili peran orang, system yang lain atau alat ketika berkomunikasi dengan use case.



Gambar 2.4 *Use Case Model (Munawar, 2005)*

2.13.2. Activity Diagram

Menurut (Romi Satria Wahono, 2003)Pada dasarnya Diagram aktivitas adalah Diagram *flowchart* yang diperluas yang menunjukkan aliran kendali satu aktivitas ke aktivitas lain. Kegunaan diagram ini adalah untuk memodelkan workflow atau jalur kerja, memodelkan operasi, bagaimana objek-objek bekerja, aksi-aksi dan pengaruh terhadap objek. Simbol-simbol yang terdapat dalam *Activity* Diagram, diantaranya sebagai berikut :

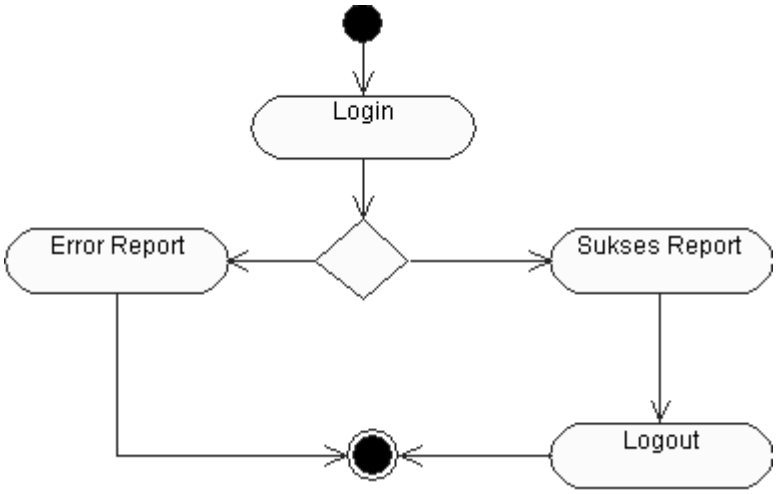
Tabel 2.1 Simbol Activity Diagram (Romi Satrio Wahono, 2003)

Keterangan	**Simbol**
Titik Awal atau permulaan.	
Titik Akhir atau akhir dari aktivitas.	
Aktiviti, atau aktivitas yang dilakukan oleh aktor.	Login

Decision, atau pilihan untuk mengambil keputusan.	
Arah tanda panah alur proses.	

Activity diagram menunjukkan apa yang terjadi, tetapi tidak menunjukkan siapa yang melakukan apa. Dalam pemprograman hal tersebut tidak menunjukkan *class* mana yang bertanggungjawab atas setiap *action*. Pada pemodelan bisnis, hal tersebut tidak bisa menunjukkan organisasi mana yang menjalankan sebuah *action*. *Swimlane* adalah sebuah cara untuk mengelompokan *activity* berdasarkan *actor* (mengelompokkan *activity* dari sebuah urutan yang sama). *Actor* bisa ditulis nama *actor* ataupun sekaligus dengan lambang *actor* (*stick figure*) pada *usecase diagram*. *Swimlane* digambarkan secara vertikal, walaupun terkadang digambarkan secara horizontal.

Activity diagram merupakan salah satu diagram yang umum digunakan dalam UML untuk menjabarkan proses atau aktivitas dari aktor. Sebagai contoh, pelanggan melakukan *login* (masuk) pada halaman *website* untuk bergabung, jika pelanggan belum terdaftar, maka akan ditolak oleh sistem dan dikembalikan. Proses penjabarannya adalah sebagai berikut :



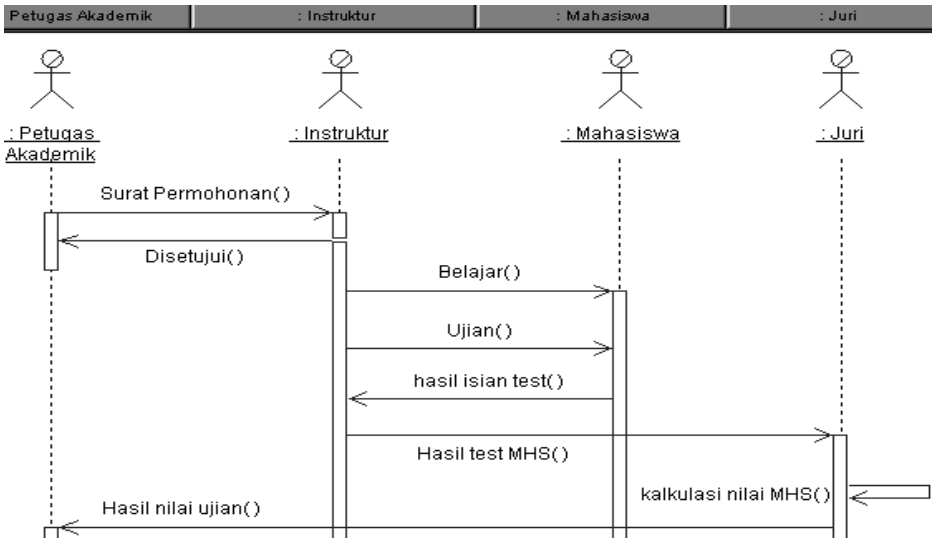
Gambar 2.5 Activity Diagram (Romi Satrio Wahono, 2003)

Di dalam *Activity* diagram tersebut dijelaskan bahwa *user* melakukan proses *login* untuk dapat memasuki area sistem, jika proses *login* dan/atau *user* belum teregistrasi, maka *user* akan ditolak oleh sistem tersebut dan diberi pesan *error*. Selain itu, bila *user* telah teregistrasi dan memasukkan kode *login* dengan benar maka akan diberi akses untuk masuk ke sistem, dan diberikan pesan sukses. *User* dapat *logout* (keluar) untuk mengakhiri sesi.

2.13.3 Sequence Diagram

Menurut Sri Dharwiyanti (2003) *Sequence Diagram* adalah suatu diagram yang digunakan untuk memodelkan skenario penggunaan. Sekenario penggunaan adalah barisan kejadian yang terjadi selama satu kesekusi sistem. *Sequence* diagram digunakan untuk :

1. *Overview* perilaku sistem.
2. Menunjukkan objek-objek yang diperlukan.
3. Mendokumentasikan skenario dari suatu *use-case*
4. Memeriksa jalur-jalur pengaksesan



Gambar 2.6 contoh Sequence Diagram (Sri Dharwiyanti, 2003)