

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Sekilas Tentang EPSBED**

EPSBED adalah singkatan dari Evaluasi Program Studi Berbasis Evaluasi Diri, dalam department ini tugasnya untuk melaporkan jumlah dosen, jumlah mahasiswa, bagaimana sarana prasarananya, PMB, Penelitian Dosen, Nilai Mahasiswa, mengajukan NIDON : Nomor Induk Dosen Nasional dan NIMAN : Nomor Induk Mahasiswa Nasional, tujuannya adalah untuk menghindari Dosen tetap di 2 Perguruan Tinggi, atau untuk menghindari mahasiswa mempunyai Ijazah Palsu.

Salah satu indikator suatu program studi pada perguruan tinggi masih aktif dan taat asas adalah dengan melihat laporan EPSBED. Pada pelaksanaannya sangat sulit bagi Kopertis untuk mencapai 100% dalam pengumpulan laporan EPSBED dari perguruan tinggi, hal ini di sebabkan oleh beberapa hal misalnya tidak semua program studi yang dimiliki perguruan tinggi aktif (program studi tersebut tidak memiliki mahasiswa), dan kurangnya perhatian perguruan tinggi pada saat itu tentang penting laporan EPSBED.

Pada akhir tahun 2009 dibuat aturan baru bahwa laporan EPSBED menjadi salah satu syarat utama untuk pengusulan perpanjangan ijin program studi. Hal ini yang membantu untuk memberikan pemahaman tentang pentingnya laporan EPSBED. Dengan adanya aturan tersebut Kopertis tidak perlu lagi bersusah payah mengingatkan perguruan tinggi untuk memberikan laporan. Apabila perguruan

tinggi tidak memberikan laporan EPSBED maka dengan sendirinya perpanjangan ijin program studi tidak bisa diproses.

Pada tahun 2013 perpanjangan ijin program studi dilebur ke dalam proses akreditasi, dengan kata lain bahwa setiap program studi yang status akreditasi masih berlaku berarti status perpanjangan ijin masih berlaku, demikian pula sebaliknya.

Ada beberapa keuntungan dan manfaat yang diperoleh dari data ESPBED tersebut yaitu :

- 1) Bagi Kopertis laporan EPSBED sangat bermanfaat dalam proses pemberian beasiswa PPA, BBM dan Bidik Misi.
- 2) Pada tahun ini banyak permohonan legalisir ijazah yang ijazahnya dikeluarkan pada tahun 2001 kebawah yang ternyata dipergunakan sebagai syarat menjadi calon legislatif (caleg). Ternyata inilah salah satu pentingnya laporan EPSBED sehingga Kopertis tidak tertipu dalam melegalisir ijazah dan juga bagi perguruan tinggi semua lulusan dapat terjamin.
- 3) Bagi perguruan tinggi yang akan mewisuda lulusannya data EPSBED menjadi acuannya, bila semua calon wisudawan terdaftar dalam data EPSBED maka Kopertis berkenan untuk menghadiri wisudanya.

#### **2.1.1 Kaitan EPSBED dengan Lembaga Lain**

- 1) DIKTI (Pendidikan Tinggi)

DIKTI adalah lembaga yang memiliki peranan penting dalam dunia pendidikan dan merupakan lembaga yang dikhususkan untuk berbagai macam pendidikan tingkat tinggi mulai dari sarana prasarana hingga pihak

terkait seperti mahasiswa, dosen, beasiswa, perguruan tinggi negeri, perguruan tinggi swasta, berbagai event, lomba, dan sebagainya.

DIKTI sangat berperan penting dalam mengelola EPSBED. Setelah Perguruan Tinggi menyerahkan laporan EPSBED kepada KOPERTIS, maka KOPERTIS akan menyerahkan laporan EPSBED (sudah diperiksa oleh KOPERTIS) tersebut kepada DIKTI untuk diupload ke PDPT (Pangkalan Data Pendidikan Tinggi).

## 2) BAN-PT (Badan Akreditasi Nasional Perguruan Tinggi)

BAN-PT merupakan satu satunya badan akreditasi yang diakui oleh Pemerintah. BAN-PT memiliki wewenang untuk melaksanakan sistem akreditasi pada pendidikan tinggi, dalam wewenang ini termasuk juga melaksanakan akreditasi bagi semua institusi pendidikan tinggi baik untuk Perguruan Tinggi Negeri (PTN), Perguruan Tinggi Swasta (PTS), Perguruan Tinggi Agama (PTA) dan Perguruan Tinggi Kedinasan (PTK). Fungsi utama BAN-PT menurut peraturan perundangan yang ada (UU No. 20 tahun 2003, PP No. 60/1999, SK Menteri Pendidikan Nasional No. 118/U/2003), pada dasarnya adalah membantu Menteri Pendidikan Nasional dalam pelaksanaan salah satu kewajiban perundangannya, yaitu penilaian mutu Perguruan Tinggi.

Lembaga BAN-PT ini tidak ada kaitannya dengan EPSBED, karena Laporan untuk akreditasi tidak digabungkan ke dalam Laporan EPSBED. Tetapi untuk proses konfirmasi BAN-PT sering melihat PDPT, apakah laporan tersebut sama atau tidak dengan data yang ada pada PDPT. Laporan

untuk akreditasi ini disebut Borang. Borang adalah instrumen akreditasi yang berupa formulir yang berisikan data dan informasi yang digunakan untuk mengevaluasi dan menilai mutu suatu program studi.

### **2.1.2 Metode Pelaporan EPSBED**

Sampai saat ini ada dua cara pelaporan data EPSBED dari Perguruan Tinggi ke Kopertis, yaitu (Sumber presentasi sosialisasi EPSBED, Kopertis 3, 2012 ) :

1. Modus Tatap Muka : mengirim file laporan langsung ke kopertis
2. Berbasis web online : mengupload database laporan melalui website kopertis. Tapi cara ini belum banyak yang memakainya, karena belum efektif.

Jenis file pelaporan EPSBED :

- Pelaporan mdb
- Pelaporan csv
- Pelaporan dbf

## **2.2 Pengertian Sinkronisasi dan Validasi**

### **2.2.1 Sinkronisasi**

Sinkronisasi adalah proses pengaturan jalannya beberapa proses pada saat yang bersamaan. Tujuan utama sinkronisasi adalah menghindari terjadinya inkonsistensi data karena pengaksesan oleh beberapa proses yang berbeda (mutual exclusion) serta untuk mengatur urutan jalannya proses sehingga dapat berjalan dengan lancar dan terhindar dari deadlock atau starvation (Stallings,2001).

### 2.2.1.1 Jenis Sinkronisasi

#### 1). Sinkronisasi dua arah

Sinkronisasi dua arah adalah jenis sinkronisasi normal di mana slave dan master menukar informasi tentang data yang dimodifikasi dalam perangkat. slave perangkat yang pertama mengirimkan modifikasi. Menurut informasi dari slave, master memproses permintaan sinkronisasi dan data dari slave dibandingkan dan disatukan dengan data di master. Setelah itu, master akan mengirimkan data yang dimodifikasi untuk perangkat slave, yang kemudian dapat memperbarui databasenya dengan data dari master.

#### 2). Sinkronisasi Lambat (*Low Sync*)

Sinkronisasi lambat (*Low Sync*) adalah bentuk sinkronisasi dua arah di mana semua item dalam satu atau lebih database yang dibandingkan satu sama lain pada field-field dasar. Dalam prakteknya, sinkronisasi lambat berarti slave mengirimkan semua data dalam database ke master, dan server melakukan analisis sinkronisasi untuk data ini dan data di master. Setelah analisis sinkronisasi, master mengembalikan semua modifikasi yang diperlukan kembali ke slave. Juga, slave mengembalikan item Map untuk semua item data yang ditambahkan oleh master.

#### 3). Sinkronisasi satu arah

##### a. Sinkronisasi satu arah dari slave

Sinkronisasi satu arah dari slave adalah jenis sinkronisasi di mana slave mengirimkan semua modifikasi ke master tetapi master tidak mengirim modifikasi kembali ke slave. Dengan demikian, master mencakup semua data

yang dimodifikasi dari slave tetapi slave tidak tahu tentang modifikasi di master. Operasi ini hanya mengekspor data dari slave ke master.

b. Mengganti semua data dengan data slave (Sinkronisasi Update dari Slave)

"Sinkronisasi update dari slave" adalah jenis sinkronisasi di mana slave mengirimkan semua data dari database ke master. Master diharapkan untuk mengganti semua data dalam database dengan data yang dikirim oleh slave. Ini berarti bahwa data slave akan menimpa semua data di database master. Operasi ini seperti mengekspor data ke master, di samping itu semua data di master akan diganti.

c. Sinkronisasi satu arah dari master

Sinkronisasi satu arah dari master adalah kasus di mana slave mendapatkan semua modifikasi dari master tetapi slave tidak mengirimkan modifikasi ke master. Dengan demikian, slave mencakup semua data yang dimodifikasi dari master tetapi master tidak tahu tentang modifikasi di slave. Operasi ini impor data dari master.

d. Mengganti semua data dengan data master (Sinkronisasi Update dari Master)

"Sinkronisasi update dari master" adalah jenis sinkronisasi di mana master akan mengirimkan semua data ke slave. Slave diharapkan untuk mengganti semua data dalam database dengan data dari master. Ini berarti bahwa data master akan menimpa semua data di database slave. Operasi ini seperti mengimpor dari master, di samping itu semua data di slave akan diganti.

### **2.2.2 Validasi**

Validasi berasal dari kata validity yang mempunyai arti sejauh mana ketepatan dan kecermatan suatu alat ukur dalam melakukan fungsi ukurnya. Suatu alat ukur yang valid tidak hanya mampu menghasilkan data yang tepat, akan tetapi juga harus memberikan gambaran yang cermat mengenai data tersebut. Menggunakan alat ukur yang dimaksudkan untuk mengukur suatu aspek tertentu, akan tetapi bila tidak dapat memberikan hasil ukur yang cermat dan teliti akan menimbulkan kesalahan atau eror. Alat ukur yang valid akan memiliki tingkat kesalahan yang kecil sehingga angka yang dihasilkannya dapat dipercaya sebagai angka yang sebenarnya atau angka yang mendekati keadaan sebenarnya (Azwar, 1986).

## **2.3 Aplikasi Web dan Database**

### **2.3.1 Aplikasi Web**

Perkembangan dunia internet telah melahirkan suatu fasilitas layanan baru, yaitu web, yang merupakan layanan terpenting internet. Fasilitas web mengizinkan pengakses untuk mengakses dan berinteraksi dengan teks, grafik, animasi, foto, suara dan video. Web secara fisik adalah kumpulan komputer pribadi, web browser, koneksi ke ISP, komputer server, router, dan switch yang digunakan untuk mengalirkan informasi dan menjadi wahana pertama sebagai pihak terkait.

Web dikenal dengan sistem client-server. Komputer pengguna disebut komputer client, sedangkan komputer yang di akses disebut server. Web

menggunakan protokol yang disebut HTTP (*HyperText Transfer Protocol*) yang berjalan pada TCP/IP. Perikat antarhalaman web disebut *Hyperlink* dan *Hypertext*. Agar dapat menjelajahi web, kita membutuhkan perangkat lunak yang disebut web browser. Kemudian, membangun sebuah web ditulis pada bahasa komputer yang dikenal *Hypertext Markup Language* (HTML).

Pada prakteknya, bahasa komputer berbasis web tidak hanya HTML, melainkan juga melibatkan bahasa pemrograman lain seperti PHP atau Perl. Tujuannya adalah untuk membentuk halaman yang bersifat dinamis. Aplikasi web itu sendiri dibagi menjadi dua, Web Statis dan Web Dinamis. Web statis dibentuk dengan menggunakan HTML saja. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus-menerus untuk mengikuti setiap perubahan yang terjadi. Kelemahan ini diatasi dengan model aplikasi web dinamis. Dengan memperluas kemampuan HTML, yakni dengan menggunakan perangkat lunak tambahan, perubahan informasi dalam halaman-halaman web dapat ditangani melalui perubahan data, bukan melalui perubahan program. Sebagai implementasinya, aplikasi dapat dikoneksikan ke basis data. Dengan demikian, perubahan informasi dapat dilakukan oleh operator atau yang bertanggung jawab terhadap kemuktakiran data, dan tidak menjadi tanggung jawab pemrograman atau Webmaster.

### **2.3.2 Database**

Menurut Bambang Hariyanto (2008), data adalah rekaman mengenai fenomena/fakta yang ada atau yang terjadi. Data pada pokoknya adalah refleksi



fakta yang ada. Data mengenai fakta-fakta penting organisasi harus 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 agar data yang dimiliki organisasi dapat berdaya guna secara maksimal.

*Database* adalah kumpulan data yang saling berhubungan yang merefleksikan fakta-fakta yang terdapat di organisasi. Saat satu kejadian muncul di dunia nyata mengubah *state* organisasi/perusahaan/sistem maka satu perubahan pun harus dilakukan terhadap data yang disimpan di *database*. *Database* merupakan komponen utama sistem informasi karena semua informasi untuk pengambilan keputusan berasal dari data di *database*. Pengelolaan *database* 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 pengaksesan basisdata. Fungsi sistem manajemen basisdata saat ini yang penting adalah menyediakan basis untuk sistem informasi manajemen.

## 2.4 Software yang Digunakan dalam Pembuatan Aplikasi

### 2.4.1 HTML

HTML kependekan dari *Hyper Text Markup Language*. Dokumen HTML adalah file teks murni yang dapat di buat dengan editor teks sembarang. Dokumen ini dikenal sebagai web page. Dokumen HTML merupakan dokumen yang disajikan dalam browser web surfer. Dokumen ini umumnya berisi informasi atau interface aplikasi di dalam internet. Ada dua cara untuk membuat sebuah web page, dengan HTML editor atau dengan editor teks biasa (misalnya notepad).

Dokumen HTML disusun oleh elemen-elemen. “Elemen” merupakan istilah bagi komponen-komponen dasar pembentuk dokumen HTML. Beberapa contoh elemen adalah head, body, table, paragraf, dan list. Elemen dapat berupa teks murni, atau bukan teks, atau keduanya. (Abdul Kadir, 2008)

### 2.4.2 CSS

CSS merupakan singkatan dari *Cascading Style Sheet*, merupakan fitur baru dari HTML 4.0. hal ini diperlukan setelah melihat perkembangan HTML menjadi kurang praktis karena web pages terlalu banyak dibebani hal-hal yang berkaitan dengan faktor tampilan seperti font dan lain-lain. Untuk itu jika kumpulan isi gaya (*style*) tersebut dikelola secara terpisah maka manajemen pages menjadi lebih mudah dan efisien. (Bambang Hariyanto, 2008)

### 2.4.3 PHP

Menurut Budi Raharjo (2011), PHP merupakan salah satu bahasa pemrograman berbentuk skrip yang ditempatkan dalam server dan diproses di server. Hasilnya yang dikirim ke klien, tempat pemakai menggunakan *browser*. Secara khusus, PHP dirancang untuk membentuk aplikasi web dinamis. Maksudnya, PHP mampu menghasilkan *website* yang secara terus-menerus hasilnya bisa berubah-ubah sesuai dengan pola yang diberikan. Hal tersebut tergantung pada permintaan *client browser*-nya (bisa menggunakan *browser* Opera, Internet Explorer, Mozilla, dan lain-lain). Umumnya, pembuatan web dinamis berhubungan erat dengan *Database* sebagai sumber data.

PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), Cold Fusion, ataupun Perl. Namun, Perlu diketahui bahwa PHP sebenarnya bisa dipakai secara *Command Line*. Artinya, skrip PHP dapat dijalankan tanpa melibatkan *web server* maupun *browser*.

Model kerja HTML diawali dengan permintaan suatu halaman web oleh *browser*. Berdasarkan URL (*Uniform Resource Locator*) atau dikenal dengan sebutan alamat internet, *browser* mendapat alamat dari *web browser*, mengidentifikasi alamat yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *web server*. Selanjutnya, *web server* akan mencari *file* yang diminta dan memberikan isinya ke *web browser* (atau yang biasa disebut *browser* saja). *Browser* yang mendapatkan isinya segera melakukan proses penerjemah kode HTML dan menampilkan ke layar pemakai.

Bagaimana halnya yang diminta adalah sebuah halaman PHP? Prinsipnya serupa dengan kode HTML. Hanya saja, ketika berkas PHP yang diminta didapatkan oleh web server, isinya segera dikirimkan ke mesin PHP dan mesin inilah yang memproses dan memberikan hasilnya (berupa kode HTML) ke *Web Server*. Selanjutnya, *web server* menyampaikan ke klien.

#### **2.4.4 JavaScript**

JavaScript adalah kode untuk menyusun halaman web yang memungkinkan dijalankan di sisi klien (pada browser yang digunakan pemakai). Karena dijalankan di sisi klien, maka JavaScript digunakan untuk membuat tampilan lebih dinamis. (Abdul Kadir, 2011).

#### **2.4.5 JQuery**

Jquery merupakan salah satu pustaka yang dikembangkan dengan mengembangkan JavaScript. Kehadirannya adalah untuk memudahkan penulisan kode JavaScript. Dengan menggunakan JQuery, penulisan kode JavaScript menjadi lebih sederhana (lebih ringkas). Selain itu, yang lebih penting lagi, pembuatan web yang interaktif dan menarik menjadi jauh lebih mudah diimplementasikan daripada jika Anda menggunakan JavaScript sendiri. (Abdul Kadir, 2011).

#### 2.4.6 MYSQL

MySQL adalah salah satu jenis *database server* yang sangat terkenal kepopulerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *databasenya*. Selain itu ia bersifat *open source* (tidak perlu membayar untuk menggunakannya) pada pelbagai platform (kecuali untuk jenis Enterprise, yang bersifat komersial).

MySQL termasuk jenis RDBMS (*Relational Database Management System*). Itulah sebabnya, istilah seperti tabel, baris, dan kolom digunakan pada MySQL. Pada MySQL, sebuah *database* mengandung satu atau beberapa kolom.

#### 2.5 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:

- *use case diagram*
- *class diagram*
- *statechart diagram*
- *activity diagram*
- *sequence diagram*
- *collaboration diagram*
- *component diagram*
- *deployment diagram*

(Munawar, 2005).

### 2.5.1 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.

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

*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.

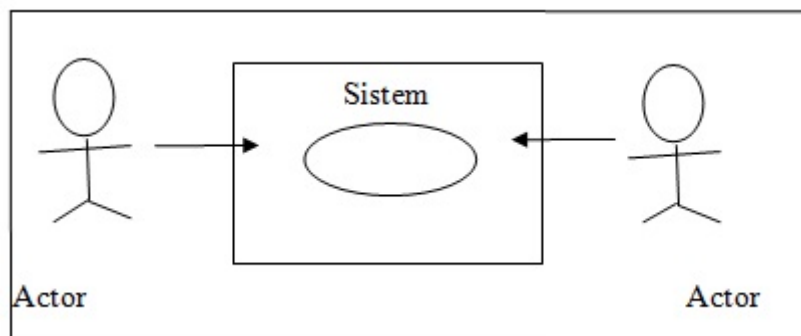
Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang

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-extend use 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 menunjukkan 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 2.1 Use Case Model (Munawar, 2005)






### 2.5.2 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)

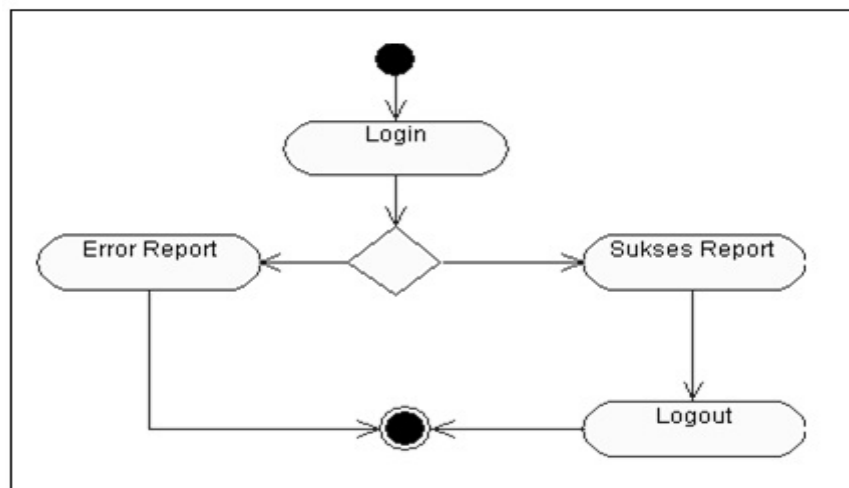
Keterangan	Simbol
Titik Awal atau permulaan.	
Titik Akhir atau akhir dari aktivitas.	
Aktiviti, atau aktivitas yang dilakukan oleh aktor.	
Decision, atau pilihan untuk mengambil keputusan.	
Arah tanda panah alur proses.	

*Activity diagram* menunjukkan apa yang terjadi, tetapi tidak menunjukkan siapa yang melakukan apa. Dalam pemrograman 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 mengelompokkan *activity* berdasarkan *actor* (mengelompokkan *activity* dari sebuah urutan yang sama). *Actor* bisa ditulis nama *actor* ataupun sekaligus dengan lambang *actor* (*stick figure*) pada *use case*

*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 :

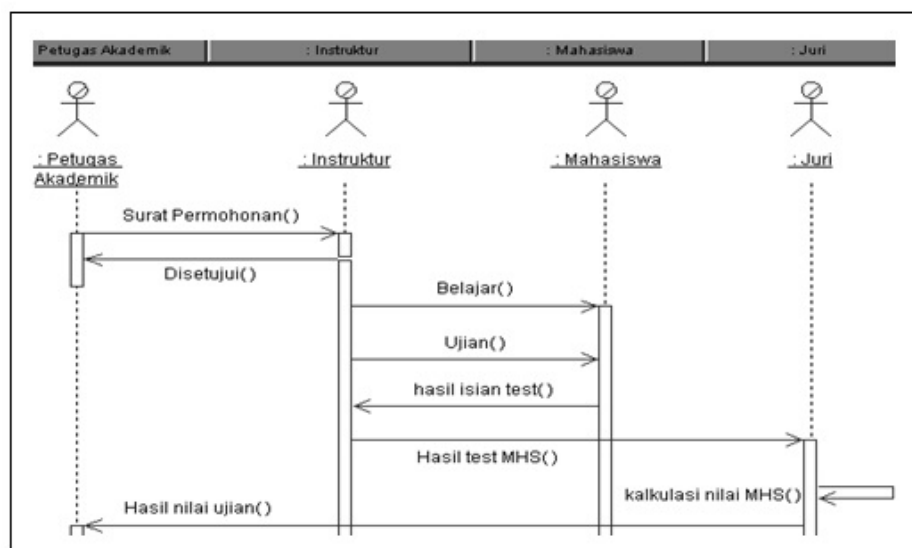


Gambar 3.2 *Activity* Diagram (Romi Satrio Wahono, 2003)

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.

### 2.5.3 Sequence Diagram

*Sequence* diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case* : interaksi yang terjadi antar class, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi Berikut contoh sederhana *Sequence* diagram pada gambar 2.3.



Gambar 2.3 Contoh *Sequence* Diagram (Sri Dharwiyanti, 2003)

