

## **BAB II**

### **LANDASAN TEORI**

#### **II.1. Perancangan**

Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh dan berfungsi. Perancangan sistem dapat dirancang dalam bentuk bagal alir sistem (*System flowchart*), yang merupakan alat bentuk grafik yang dapat digunakan untuk menunjukkan urutan-urutan proses dari sistem (*Hanik Mujiati, 2014:24*).

#### **II.2. Aplikasi**

Aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan. Aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi software yang dirancang untuk penggunaan khusus, klarifikasi luas dapat dibagi menjadi 2 (dua) yaitu: (*Yudi dan Anggit, 2015:63*)

1. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
2. Aplikasi paket, suatu program dengan dikumentasi tergabung dirancang untuk jenis masalah tertentu.

#### **II.3. Interaktif**

Interaktif berasal dari kata interaksi yaitu hal saling melakukan aksi, hubungan, mempengaruhi, antar hubungan. Interaksi terjadi karena adanya sebab

akibat, yaitu adanya aksi dan reaksi. Interaktif adalah hal yang terkait dengan komunikasi dua arah / suatu hal bersifat saling melakukan aksi, saling aktif dan saling berhubungan serta mempunyai timbal balik antara satu dengan lainnya.(Mochamad Miswar Hadibin at al; 2013:3)

#### **II.4. Sejarah *Game***

*Game* atau permainan dapat diklarifikasikan menjadi dua bagian besar *game* fisik dan *game* elektronik. *Game* fisik mungkin sudah sering kita lakukan dalam kehidupan sehari-hari sewaktu masih anak-anak. Seperti lompat tali, petak umpet dan sebagainya. Dan *game* elektronik merupakan fenomena yang menarik saat ini. Bahkan dapat dikatakan bahwa hampir semua kalangan menyukai *game* elektronik. *Electronic game* atau selanjutnya dapat disebut *video game* pertama kali ditemukan oleh *Thomas t. Goldsmith Jr* dan *Estle Ray Mann*. Penemuan ini dipatenkan pada Januari 1947. Yang mendasari perkembangannya saat ini adalah ketika mereka menemukan *Cathode Ray* sebuah tabung *vacuum* yang digunakan sebagai media untuk membuat simulasi kecepatan tembakan dan arah tembakan sebuah roket. Pada Februari 1947, *Christopher Strachey* memulai pengembangannya ke arah pemrograman yang mulai menggunakan memori dimana aplikasinya diterapkan untuk kebutuhan para pilot. Dan penemuan baru terus berkembang hingga tahun 1959. Namun era perkembangan konsol *game* dimulai setelah masa ini (*Rio Caesar, 2015:114*).

#### **II.4.1 Genre Game**

*Genre game* adalah klasifikasi *game* yang didasari interaksi pemainnya. Visualisasi juga menjadi ukuran klasifikasi *game* ini. Namun untuk beberapa kasus pengembangan *game* membuat kompilasi antar berbagai *genre* ini. Tentu saja variasi format *game* lebih banyak (*Rio Caesar, 2015:114*).

#### **II.5 Android**

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* yang terbuka bagi para penganmbang untuk menciptakan aplikasi. Antarmuka pengguna android didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan didunia nyata, seperti menggesek, mengetuk, mencubit dan membalikkan cubitan untuk memanipulasi obyek pada layar. Android memiliki sejumlah besar komunitas pengembang aplikasi (*apps*) yang memperluas fungsionalisasi perangkat, umumnya ditulis dalam versi bahasa pemrograman Java yang telah di-*Custom* (*Yudi dan Anggit, 2015:64*).

Android adalah istilah dalam bahasa komputer yang berarti “Robot yang menyerupai manusia”. Logo Android sendiri dicerminkan seperti sebuah robot berwarna hijau yang mengacu kepada arti kata android.



**Gambar II.1 Logo Android**

*Sumber: (Florina Yunafania Dkk, 2015:22)*

Android adalah sistem operasi yang bersifat *open source* (Sumber terbuka). Disebut *open source* karena *source code* dari sistem operasi android dapat dilihat, di-*download*, dan dimodifikasi secara bebas. Paradigma *open source* ini memudahkan pengembangan teknologi android, karena semua pihak yang tertarik dapat memberikan kontribusi, baik pada pengembangan sistem operasi maupun aplikasi (*Florina Yunafania, Dkk; 2015:22*).

### **II.5.1 Versi OS Android**

Berikut ini urutan atau tingkatan versi nama OS Android sesuai dengan perkembangannya.

**Tabel II.1 Versi OS Android**

Sumber: ( Defri dan Indri Rahmayun; 2014:65)

<b>Versi Android</b>	<b>Nama Android</b>
Android v1.0	Astro (Alpha)
Android v1.1	<b>Bender (Beta)</b>
Android v1.5	<b>Cupcake</b>

Android v1.6	<b>Donut</b>
Android v2.1	<b>Eclair</b>
Android v2.2	<b>Frozen Yoghurt (Froyo)</b>
Android v2.3	<b>Gingerbread</b>
Android v3.0 – 3.2	<b>Honeycomb</b>
Android v4.0	<b>Ice Cream Sandwich</b>
Android v4.1 – 4.3	<b>Jelly Bean</b>
Android v4.4	<b>Kitkat</b>
Android v5.0 – 5.1	<b>Lollipop</b>

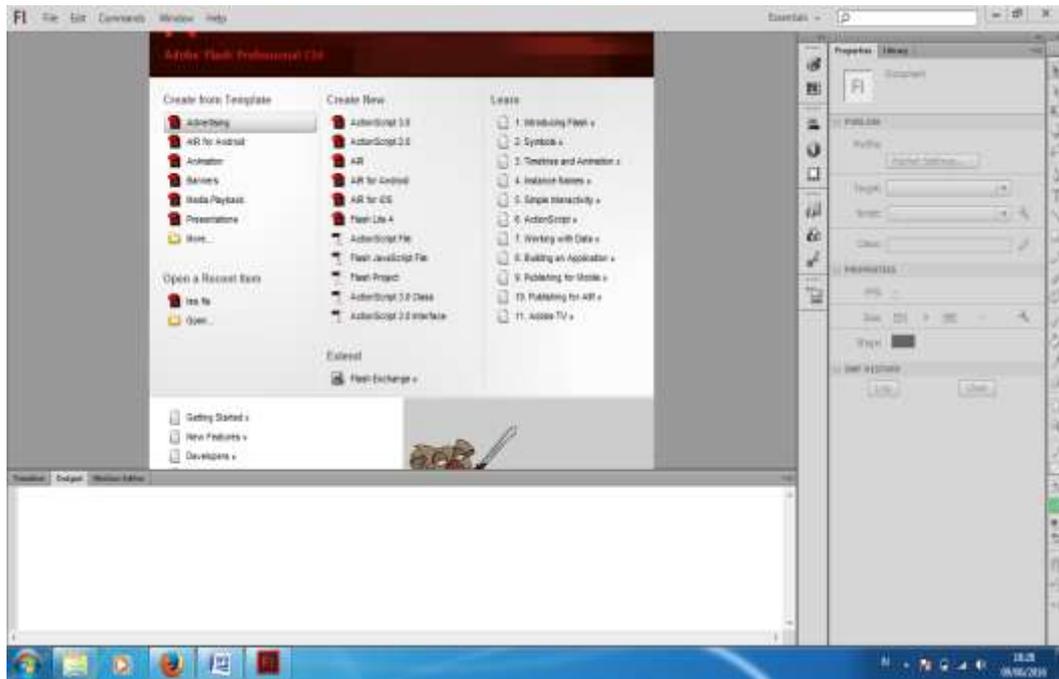
## II.6 Adobe Flash Professional CS6

Adobe Flash ( dahulu bernama Macromedia Flash) adalah salah satu perangkat lunak komputer yang merupakan produk unggulan Adobe *System*. Adobe Flash digunakan untuk membuat gambar vektor maupun animasi gambar tersebut. Berkas yang dihasilkan dari perangkat lunak ini mempunyai file *ekstension* .swf dan dapat diputar di penjelajah web yang telah dipasang *adobe flash player*. Flash menggunakan bahasa pemrograman bernama *Action Script* yang muncul pertama kalinya pada flash 5.

Adapun langkah-langkah untuk menjalankan Adobe Flash Professional CS6 adalah sebagai berikut:

1. Klik tombol *Start > Programs > Adobe Collection > Adobe Flash Professional CS6*.

Tampilan start Adobe flash Professional CS6 dapat dilihat pada gambar berikut ini:



**Gambar II.3 Tampilan Start Adobe Flash Professional CS6**

*Sumber: ( Dr. Suparji Dkk; 2015:84)*

Sebelum tahun 2005, flash dirilis oleh adobe. Flash 1.0 diluncurkan pada tahun 1996 setelah adobe membeli program animasi vektor bernama *Future Splash*. Versi terakhir yang diluncurkan dipasaran dengan menggunakan nama Adobe adalah Adobe Flash 8. Pada tanggal 3 desember 2005 Adobe Systems mengakuisisi Macromedia dan seluruh produknya, sehingga nama Macromedia Flash berubah menjadi Adobe Flash. Adobe Flash merupakan sebuah program yang didesain khusus oleh adobe dan program aplikasi standar authoring tool professional yang digunakan untuk membuat animasi dan bitmap yang sangat menarik untuk keperluan pembangunan situs web yang interaktif dan dinamis. Flash didesain dengan kemampuan untuk membu animasi 2 dimensi yang handal dan ringan sehingga flash banyak digunakan untuk membangun dan memberikan efek animasi pada website, CD interaktif dan yang lainnya. Selain itu aplikasi ini

juga dapat digunakan untuk membuat animasi logo, movie, game, pembuatan navigasi pada situs web, tombol animasi, banner, menu interaktif, interaktif form isian, e-card, *screen saver* dan pembuatan aplikasi-aplikasi web lainnya. Dalam flash terdapat teknik-teknik membuat animasi, fasilitas *action script*, filter custom easing dan dapat memasukkan video lengkap dengan fasilitas *playback* FLV. Keunggulan yang dimiliki oleh adobe Flash ini adalah ia mampu diberikan sedikit *code* pemrograman baik yang berjalan sendiri untuk mengatur animasi yang ada didalamnya atau digunakan untuk berkomunikasi dengan program lain seperti *HTML*, *PHP* dan *database* dengan pendekatan XML, dapat dikolaborasikan dengan *web*.

### **II.6.1 Timeline dan Stage**

Animasi yang dibuat di Flash di organisasikan dengan timeline (representasi grafik yang terdiri dari kumpulan frame). Animasi dapat dibuat pada single frame pada suatu waktu, dengan menambahkan key frames pada timeline secara sekuensial. Layer dapat dipergunakan untuk mengorganisasikan elemen frame. Flash interface berisi vektor drawing tool, host of palletes (color mixing, alignment, applying transformations, setting typography options dan lainnya) (Andi, 2004:7).

### **II.6.2 Symbol dan Tweening**

Objek dapat disimpan pada library dalam bentuk khusus yang dinamakan symbol, sehingga dapat dipergunakan ulang beberapa instance symbol dapat

ditempatkan pada stage. Symbol dapat ditransformasi (ukuran, orientasi). Tween motion dapat dibuat dengan beberapa cara. Hasil tweening dapat dilihat pada timeline berupa tanda panah pada awal dan akhir key frame yang pilih. Tiga macam symbol di dalam flash: ( *Efi Rochmatika, 2015:19*)

1. *Graphics Symbol. Simply reusable vektor objects.* Dipergunakan untuk *motion tweening*.
2. *Button Symbol.* Dipergunakan untuk membuat bagian interaktif.
3. *Movie clip symbol.* Animasi yang dapat ditambahkan ke dalam *movie* utama.

## **II.7. Action Script**

Action script adalah bahasa pemrograman yang di pakai oleh software Flash untuk mengendalikan object-object ataupun movie yang terdapat dalam Flash. Sebenarnya Flash juga tidak bisa menggunakan Action Script dalam pemakaiannya, tapi kalau menginginkan adanya interaktifitas yan lebih kompleks maka Action Script ini dibutuhkan (*Andi,2004:97*).

## **II.8. Story Board**

Story board adalah penggambaran jalan cerita sesuai dengan isi cerita dan berisi pengambilan sudut gambar, pengisian suara, serta efek-efek khusus (*Prilliana Indah Paramitha,2014:11*).

## **II.9. UML (Unified Modelling Language)**

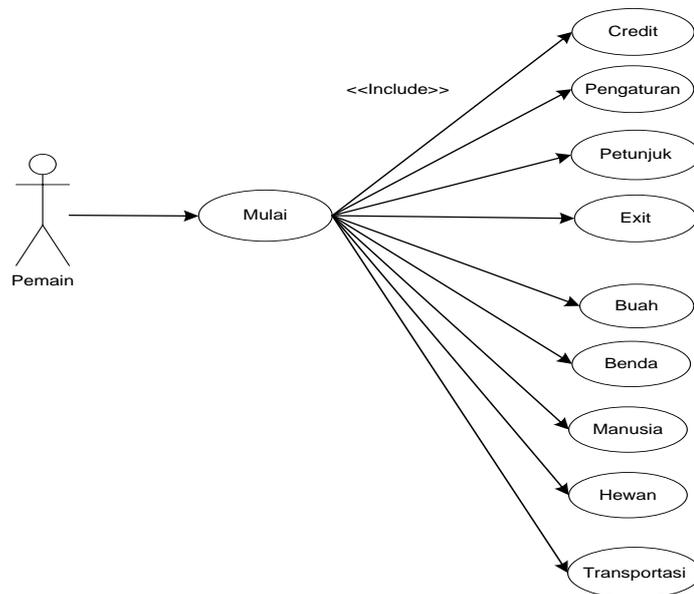
Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan Class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi object seperti C++, Java, C# atau VB. NET. Walaupun demikian, UML tetap dapat digunakan untuk modelling aplikasi prosedural dalam VB atau C. Seperti bahasa-bahasa lainnya, UML mendefenisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak.

Dalam pembuatan tugas akhir ini penulis menggunakan diagram use case yang terdapat di dalam UML. Adapun maksud dari use case diagram diterangkan dibawah ini.

### **1. Use Case Diagram**

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 dan sebagainya. Use case diagram sangat

membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua fitur yang ada pada sistem. Sebuah sistem dapat meng-include fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. sebuah use case dapat di-include 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 behaviornya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain (Prastuti Sulistyorini, 2009:26).

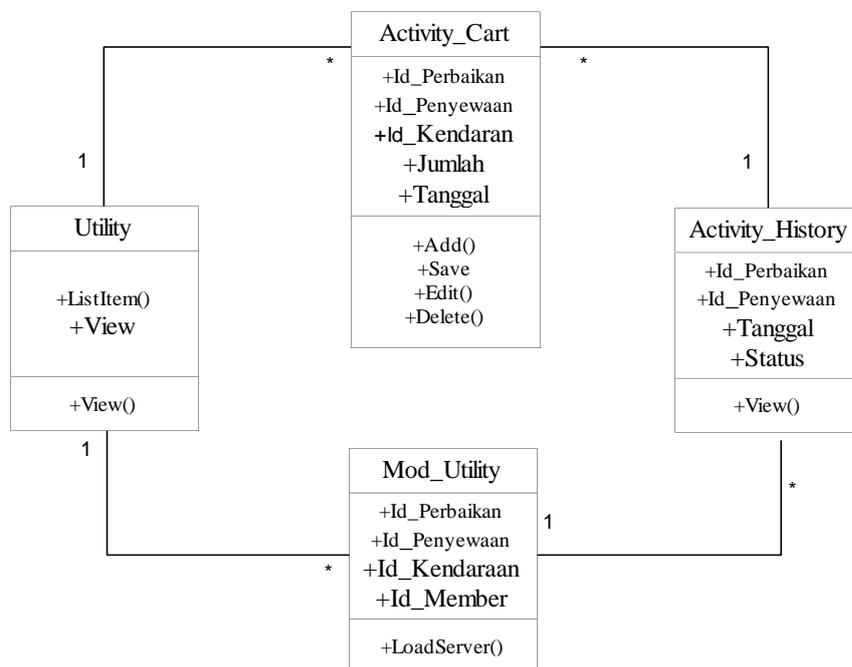


**Gambar II.4 Use Case Diagram**

Sumber : (Prastuti Sulistyorini, 2009: 26)

## 2. Class Diagram

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Berikut adalah simbol-simbol pada diagram kelas:



**Gambar II.6 Class Diagram**

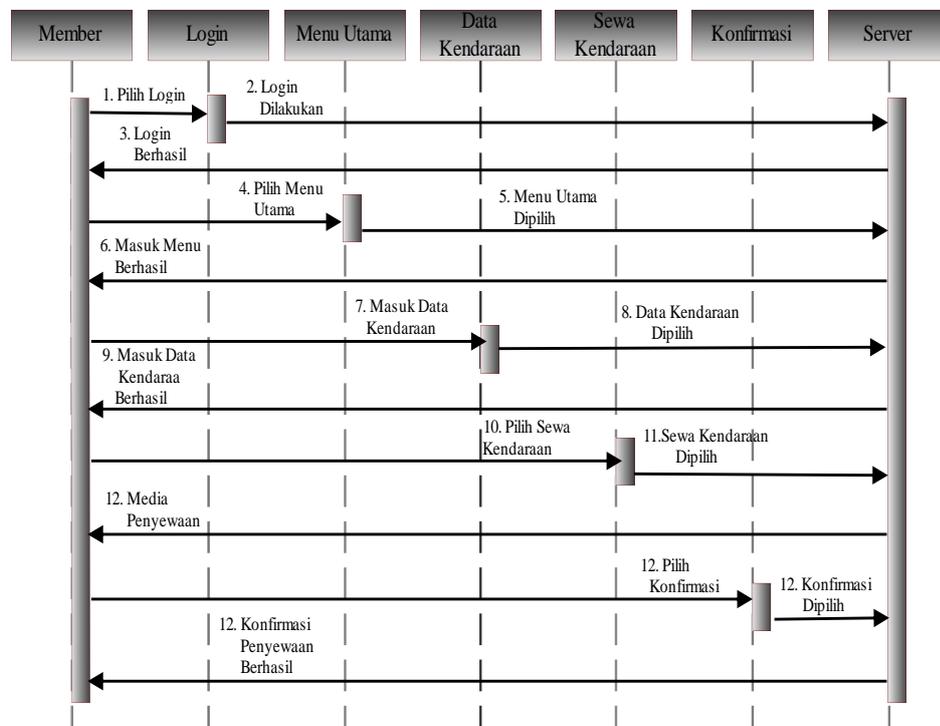
*Sumber: (Prastuti Sulistyorini, 2009: 27)*

## 3. Sequence Diagram

Diagram sequence menggambarkan pekerjaan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Oleh karena itu menggambarkan diagram sequence maka harus diketahui

objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyak diagram sequence yang harus digambar adalah sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sequence sehingga semakin banyak use case yang didefinisikan maka diagram sequence yang harus dibuat juga semakin banyak.



**Gambar II.8 Contoh Diagram Sequence**

*Sumber: (Prastuti Sulistyorini, 2009:29)*

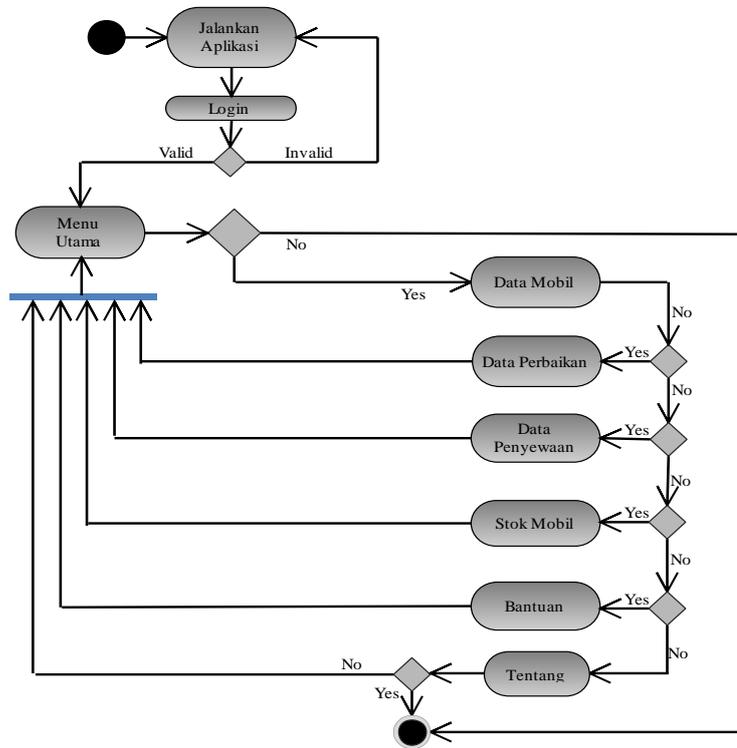
#### 4. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin

terjadi dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan state diagram khusus, dimana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti state, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas.



**Gambar II.9 Activity Diagram**

*Sumber: (Prastuti Sulistyorini, 2009:28)*