

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Pengembangan Aplikasi**

Pengembangan aplikasi kerap disalahartikan sebagai kegiatan seorang *programmer* melakukan sebuah aktifitas yaitu *coding*, proses menulis kode padahal sebenarnya pengembangan aplikasi lebih dari hanya melakukan proses menulis kode. Pengembangan Aplikasi adalah serangkaian proses yang dilakukan seorang atau sekelompok *programmer* untuk membuat rangkaian proses dari rancangan awal aplikasi, membuat prototype aplikasi, implementasi, pengujian akhir aplikasi hingga aplikasi selesai dan siap untuk digunakan.

Dalam pengembangan aplikasi ada beberapa tahap yang telah dirancang sebagai syarat agar sebuah aplikasi dapat dibuat secara terstruktur dan terancang dengan baik. Tahap tersebut adalah pengumpulan informasi, perencanaan, pengembangan, dan perawatan. Tahap ini merupakan tahap yang penting untuk *programmer* dalam membangun aplikasi. Dalam pelaksanaannya ada beberapa metode pengembangan aplikasi yang kerap digunakan oleh para *programmer* yaitu seperti metode waterfall, metode spiral, metode XP (*Extreme Programming*), metode Kanban, *agile development*, metode *prototype* dan metode scrum. Dalam hal ini penulis akan membahas metode yang akan digunakan pada penelitian ini yaitu metode *prototype*.

#### **2.2 Metode Prototype**

##### **2.2.1 Pengertian Metode Prototype**

Metode *prototype* adalah proses pengembangan aplikasi yang menggunakan pendekatan model kerja. Menurut (Mulyanto, 2009), metode ini sangat baik digunakan untuk menyelesaikan masalah kesalahpahaman antara user dan analis yang timbul akibat user tidak mampu mendefinisikan secara jelas kebutuhannya. Metode inipun menjadi salah satu metode pengembangan aplikasi yang banyak digunakan. Menurut (O'Brien, & James, 2005) metode *prototype* ini disebut juga sebagai desain aplikasi cepat (*rapid application design/RAD*) karena metode ini menyederhanakan dan mempercepat pembuatan desain sistem.

##### **2.2.2 Proses Metode Prototype**

Pada dasarnya metode *prototype* membantu menghubungkan antara *klien* dan *developer* dimana kadang *klien* hanya memberikan kebutuhan umum tanpa adanya detail *input* maupun

*output* dan *developer* pun tidak yakin akan efisiensi algoritma yang digunakan maka proses yang terdapat pada metode *prototype* yang sangat membantu proses pengembangan aplikasi.

Yaitu sebagai berikut:

- Pengumpulan kebutuhan sistem: *klien* dan *developer* melakukan pertemuan untuk menentukan tujuan dari pengembangan aplikasi, kebutuhan dan gambaran bagian yang diperlukan. Pengumpulan kebutuhan meliputi kebutuhan *hardware*, *software* dan juga sumber daya manusia. Pengumpulan kebutuhan sistem harus mendefinisikan kebutuhan sistem secara detail dan spesifik yaitu:
  - a. Masukan yang diperlukan (input).
  - b. Keluaran yang dihasilkan (output).
  - c. Operasi yang dilakukan (proses).
  - d. Sumber data yang ditangani.
  - e. Pengendalian (maintenance).
- Perancangan: Perancangan terdiri dari aktivitas desain yang akan menghasilkan spesifikasi fungsional. Menurut Loanna. Et (2007), desain terbagi antara desain interface, data dan proses yang bertujuan untuk menghasilkan spesifikasi produk dan interface, struktur database, pemrosesan dan pengendalian. Desain sistem yang baik mencakup tujuh bagian:
  - a. Fitur menu yang cepat dan mudah.
  - b. Tampilan input dan output.
  - c. Laporan yang mudah dicetak.
  - d. Data dictionary yang menyimpan informasi disetiap *field*.
  - e. Database dengan format yang baik.
  - f. Menampilkan *query* yang tersimpan pada *database*.
  - g. Struktur sederhana dan Bahasa pemrograman yang mengoptimalkan pemrosesan, waktu dan prosedur.
- Pengujian sistem: Menurut Sommerville. (2001), pengujian terdiri dari :
  - a. Pengujian unit yang menguji komponen individual secara independent.
  - b. Pengujian modul.
  - c. Pengujian sub sistem yang terintegrasi.
  - d. Pengujian sistem yang bertujuan menemukan kesalahan dari interaksi antara subsistem dan *interface*.
  - e. Pengujian penerimaan data.

- f. Dokumentasi pencatatan dari awal sampai akhir program.



Gambar 2.1 Model *Prototype*

### 2.2.3 Kelebihan dan Kekurangan Metode *Prototype*

Kelebihan metode *Prototype*

1. Komunikasi yang terjalin baik antara *developer* dan *klien*.
2. *Developer* bekerja lebih baik dalam menentukan kebutuhan *klien*.
3. Menghemat waktu dalam pengembangan aplikasi.
4. Penerapan lebih mudah karena *klien* mengetahui apa yang diinginkan.

Kekurangan metode *Prototype*

1. Beresiko tinggi untuk masalah yang tidak terstruktur, perubahan yang sangat besar dan persyaratan data yang tidak menentu.
2. Interaksi antara *developer* dan *klien* menjadi sangat penting.
3. Hubungan antara *klien* dengan computer yang disediakan tidak mencerminkan teknik perancangan yang baik.

## 2.3 *Electronic Learning*

### 2.3.1 Definisi *Electronic Learning*

*Electronic learning* (e-learning) adalah pembelajaran yang menggunakan media teknologi dan perangkat mobile. Perangkat tersebut dapat berupa komputer, gawai, laptop, tablet PC, PDA dan lain sebagainya. Dengan e-learning, user dapat mengakses segala konten pembelajaran di mana saja dan kapan saja tanpa harus ada disuatu tempat khusus dan waktu tertentu. Menurut Haryono dan Alatas (2000) e-learning merupakan sebuah konsep belajar

jarak jauh dengan memanfaatkan teknologi telekomunikasi dan informasi. Dari definisi diatas dapat diambil kesimpulan bahwa *electronic learning* adalah model pembelajaran yang membawa ketersediaan materi dan dapat diakses setiap saat dan memberikan visualisasi yang menarik. Ada beberapa fitur penting yang harus dimiliki oleh perangkat e-learning yaitu fitur untuk dapat terkoneksi dengan perangkat teknologi lain dan fitur kepada pengajar ataupun pelajar untuk dapat berkomunikasi satu sama lain.

Istilah *electronic learning* lebih merujuk pada penggunaan perangkat elektronik seperti komputer, laptop, PDA dan perangkat teknologi lain yang digunakan dalam proses belajar mengajar. Tujuan dari pengembangan *electronic learning* adalah proses pembelajaran sepanjang waktu, dimana siswa dapat lebih aktif dan menghemat waktu dalam proses pembelajaran karena tidak perlu hadir di suatu kelas untuk mengerjakan dan mengumpulkan tugas, cukup menggunakan aplikasi *electronic learning* maka akan ada wadah untuk melakukan kegiatan seperti disebutkan diatas. Secara tidak langsung hal tersebut akan meningkatkan kualitas belajar terhadap siswa itu sendiri.

### **2.3.2 Fungsi dan Manfaat *Electronic learning***

#### **Fungsi *Electronic Learning***

Terdapat tiga fungsi utama e-learning dalam proses belajar mengajar yaitu sebagai tambahan, pelengkap dan juga pengganti.

##### **1. Tambahan**

*Electronic learning* berfungsi sebagai tambahan, dimana pelajar dapat kebebasan memilih untuk memanfaatkan materi yang terdapat di aplikasi *electronic learning* atau tidak. Karena sifatnya yang fungsional, pelajar dapat memanfaatkan untuk menambah pengetahuan.

##### **2. Pelengkap**

Pada dasarnya *electronic learning* dirancang sebagai tambahan untuk melengkapi materi pembelajaran di dalam kelas. *Electronic learning* juga difokuskan untuk memperkuat materi yang didapat dan sebagai remedial bagi pelajar setelah mendapat pembelajaran konvensional.

##### **3. Pengganti**

Di beberapa perguruan tinggi di luar maupun dalam negeri memberikan alternatif pembelajaran kepada pelajar melalui *electronic learning*. Biasanya alternatif ini dilakukan jikalau pengajar sedang tidak bisa menghadiri kelas. Tapi ada juga yang memang ingin menggunakan teknologi untuk melakukan pembelajaran.

### **Manfaat *Electronic learning***

Manfaat electronic learning terdiri dari dua sudut, yaitu dari sudut pelajar dan pengajar:

#### 1. Pelajar

Kegiatan electronic learning memungkinkan berkembangnya fleksibilitas belajar yang tinggi sehingga pelajar dapat mengakses segala materi pembelajaran setiap saat dan dapat dilakukan berulang kali. Pelajar juga dapat berkomunikasi setiap saat dengan pengajar. Pelajar dapat memastikan penguasaannya terhadap materi pembelajaran.

#### 2. Pengajar

Dengan menggunakan aplikasi e-learning pengajar dapat lebih mudah melakukan pemutakhiran bahan belajar sesuai dengan kurikulum yang berlaku. Selain itu pengajar juga dapat mengontrol kegiatan belajar dan mengetahui kapan pelajar belajar, topik apa yang sedang dipelajari, berapa lama pembelajaran tersebut diulang, mengecek soal soal latihan yang diberikan kepada pelajar dan tentunya memeriksa hasil jawaban soal-soal yang diberikan.

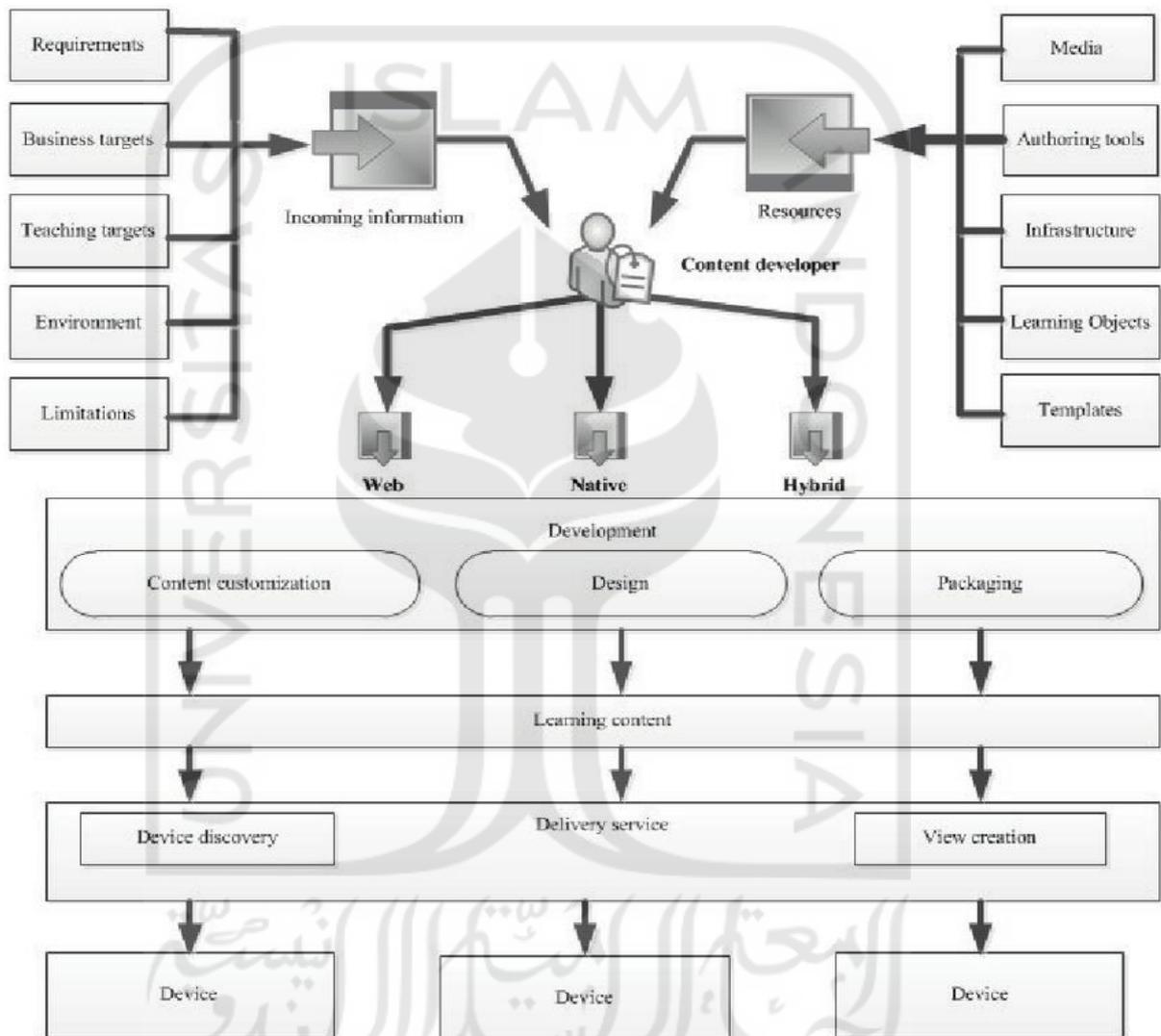
### **2.3.3 Model Desain *Electronic learning***

Pengembangan e-learning melibatkan tiga aspek yaitu: infrastruktur teknologi, sumber daya dan lingkungan. Setiap aspek memiliki peran yang berbeda tetapi tetap menciptakan suatu system yang padu. Teknologi e-learning mencakup seluruh bagian dari aplikasi dan proses, termasuk *Computer Based Learning*, *Web Based Learning*, *Virtual Classroom* dan *Digital Collaboration*. E-learning cenderung menggunakan Personal Computer (PC) dan internet sebagai media utama, sedangkan e-learning menggunakan perangkat mobile seperti gawai, PDA, laptop dan sebagainya. Menurut Andy. Yonatan (2007) menjelaskan bahwa e-learning memiliki beberapa kelebihan disbanding dengan system e-learning lainnya, yaitu:

- a. Portabilitas: perangkat mobile lebih mudah dibawa-bawa dan lebih mudah dipakai untuk membuat catatan atau memasukkan data dimanapun.
- b. Mendukung pembelajaran: generasi yang ada saat ini lebih menyukai perangkat mobile seperti PDA, telepon seluler, laptop dan perangkat handheld games.
- c. Meningkatkan motivasi: kepemilikan terhadap perangkat mobile cenderung meningkatkan komitmen untuk memakai dan mempelajarinya.
- d. Jangkauan lebih luas: perangkat mobile cenderung lebih murah sehingga dapat terjangkau oleh masyarakat secara lebih luas.
- e. Pembelajaran tepat waktu: meningkatkan *performance* kerja/pembelajaran sesuai dengan kebutuhan pelajar.

## 2.4 Learning Content Development (LCD)

Learning Content Development adalah sebuah kerangka kerja yang fungsinya sebagai fondasi awal untuk merancang sebuah system e-learning. LCD bisa disebut juga sebagai konsep pengembangan aplikasi e-learning. Berikut adalah sebuah tampilan kerangka kerja Learning Content Development seperti pada Gambar 2.2.:



Gambar 2.2 Kerangka kerja *Learning Content Development*

Dari Gambar 2.2 diatas bisa dilihat aliran kerja dari LCD ada beberapa komponen yang dibutuhkan oleh *developer* dalam pembuatan sistem e-learning:

### Informasi yang dibutuhkan

- **Kebutuhan:** Hal pertama yang dibutuhkan oleh *developer* adalah mencari apa saja yang dibutuhkan untuk pembuatan aplikasi e-learning. Kebutuhan disini bisa berupa data-data awal yang akan berkaitan dengan pembuatan sistem.

- Target bisnis: Menentukan target dari pembuatan aplikasi e-learning dari segi bisnis menjadi penting karena menentukan kedepannya pembuatan aplikasi ini dapat digunakan secara terus menerus dan tidak menjadi sia sia.
- Target Pengajar: dengan memiliki target, Pengajar akan mengetahui apa saja yang akan membuat sistem ini berhasil dalam sisi pembelajaran. Apakah sistem ini sudah sesuai dengan kebutuhan pelajar dan sebagainya.
- Lingkungan: informasi ini dibutuhkan *developer* agar dalam pembuatan sistem, segala bentuk rancangan dan desain tidak melenceng dengan lingkungan tempat sistem akan digunakan.
- Pembatasan: Pembatasan dilakukan agar saat pembuatan sistem tetap terarah dan terstruktur dan menghindari dalam pembuatan fitur yang tidak berguna.

### **Resources**

- Media: Media adalah alat penghubung. Dalam hal ini media berfungsi sebagai penghubung antara informasi yang telah didapat dan *developer* untuk menghasilkan sebuah sistem e-learning.
- *Authoring Tools*: Alat penyusun berupa paket perangkat lunak yang digunakan yang dapat digunakan sesuai kebutuhan sistem e-learning maupun alat multimedia seperti teks, gambar, video, audio dan animasi.
- Infrastruktur: Kebutuhan perangkat keras yang digunakan untuk pembuatan sistem e-learning. Infrastruktur dapat berupa computer, printer atau apapun yang mendukung pembuatan sistem.
- Objek pembelajaran: Dalam Pendidikan pengajar harus menentukan objek pelajaran apa yang akan diberikan. Sama halnya dengan e-learning juga menentukan objek apa saja yang akan dibuat menjadi konten dalam sistem.
- *Templates*: Format awal yang telah ditentukan. Digunakan sebagai titik awal dalam pembuatan aplikasi sehingga tidak perlu membuat format kembali saat menggunakannya.

Setelah mendapatkan informasi dan sumber yang sesuai. Developer bebas memilih bentuk dari aplikasi kedalam bentuk *Web Based Application*, *Native Based Application* dan *Hybrid Based Application*. Pengembangan dilanjutkan dengan pembuatan desain, kostumisasi konten dan kemasan. Pembuatan konten pembelajaran seperti penambahan materi, tugas-tugas dan kuis ditambahkan setelah pengembangan diatas selesai dilakukan. Selanjutnya adalah tahapan terakhir dengan menguji kedalam perangkat yang telah ditentukan.

## 2.5 *Unified Modelling Language (UML)*

Berikut ini definisi *Unified Modelling Language (UML)* menurut beberapa ahli :

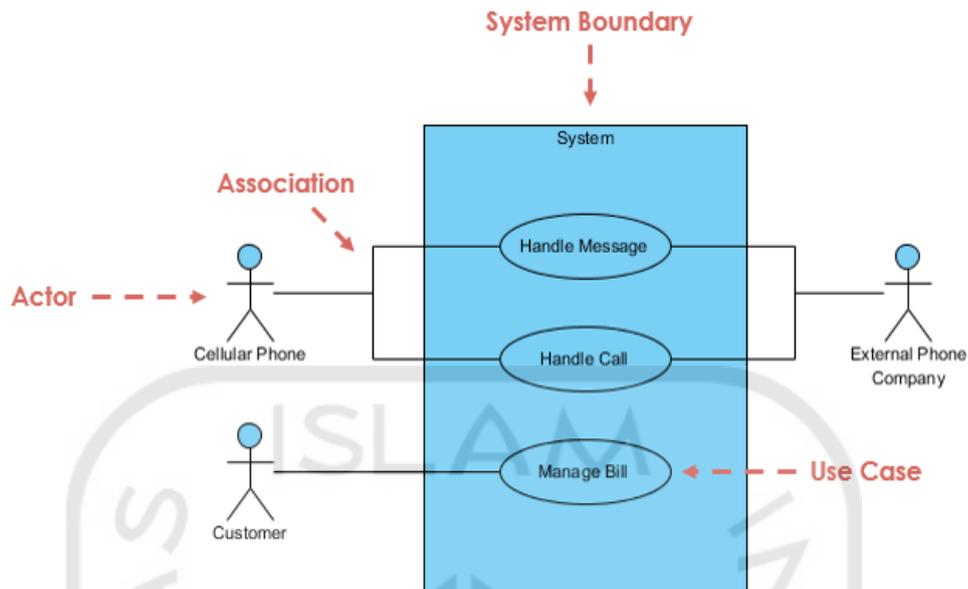
- *Unified Modeling Language (UML)* adalah sekumpulan kaidah pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak dari segi obyek (Whitten & Bentley, 2007).
- *Unified Modeling Language (UML)* adalah bahasa yang telah menjadi standard untuk visualisasi, menetapkan, membangun dan mendokumentasikan artefak suatu system perangkat lunak (Hend, 2006).
- *Unified Modeling Language (UML)* adalah alat bantu analisis serta perancangan perangkat lunak berbasis objek (Nugroho.Adi, 2005)

Berdasarkan beberapa pendapat para ahli dapat diambil kesimpulan bahwa “*Unified Modelling Language (UML)* adalah Bahasa grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan perekaman dari sistem perangkat lunak berorientasi objek”.

Perkembangan UML dimulai pada tahun 1994, Grady Booch dan James Rumbaugh menggabungkan metode pengembangan berorientasi objek yang bertujuan menciptakan proses standar untuk sistem pengembangan berorientasi objek. Kemudian Ivar Jacobson bergabung dan Bersama membuat sebuah standar Bahasa pemodelan objek UML versi 1.0.

### 2.5.1 *Use Case Diagram*

*Use Case Diagram* adalah sebuah diagram yang menggambarkan antara sistem dan *user*, bagaimana *user* akan berinteraksi dengan sistem dan siapa yang akan menggunakan sistem. Pembuatan *use case diagram* lebih terfokus pada fungsi sebuah sistem dan tidak berdasarkan alur atau urutan kejadian. *Use Case* mewakili sebuah tujuan dari sistem serta menjelaskan urutan kegiatan dan interaksi pengguna untuk mencapai tujuan pembuatan sistem. (Whitten & Bentley, 2007). Berikut adalah sebuah contoh *Use Case Diagram* pada Gambar 2.3.



Gambar 2.3 Use Case Diagram

Terdapat komponen yang terlibat dalam sebuah *use case*, diantaranya adalah :

a. *Use Case*

*Use case* digambarkan dalam bentuk elips yang merupakan aktivitas sistem yang dilakukan oleh aktor seperti pada Gambar 2.4.



Gambar 2.4 Simbol Use Case

b. *Actor*

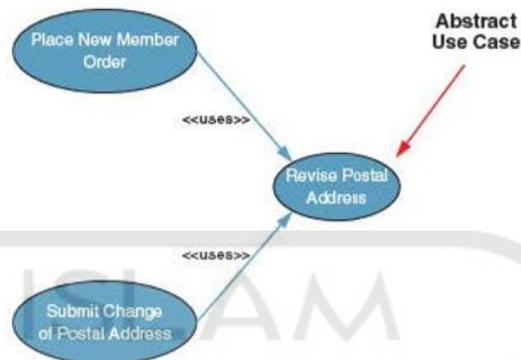
*Actor* merupakan pelaku yang melakukan interaksi dengan sistem. Actor disimbolkan seperti pada Gambar 2.5.



Gambar 2.5 Simbol Actor

c. *Relationship*

*Relationship* digambarkan sebagai anak panah yang menghubungkan antara actor dan use case. Relationship disimbolkan seperti pada contoh Gambar 2.6.



Gambar 2.6 *Relationship*

Terdapat *relationship* yang terdapat pada *use case diagram*:

1. *Association*, menghubungkan link antar elemen.
2. *Generalization*, biasa disebut pewarisan, sebuah elemen dapat merupakan spesialisasi dari elemen lainnya.
3. *Dependency*, sebuah elemen bergantung dalam beberapa cara ke elemen lain.
4. *Aggregation*, bentuk *association* dimana elemen berisi elemen lainnya.

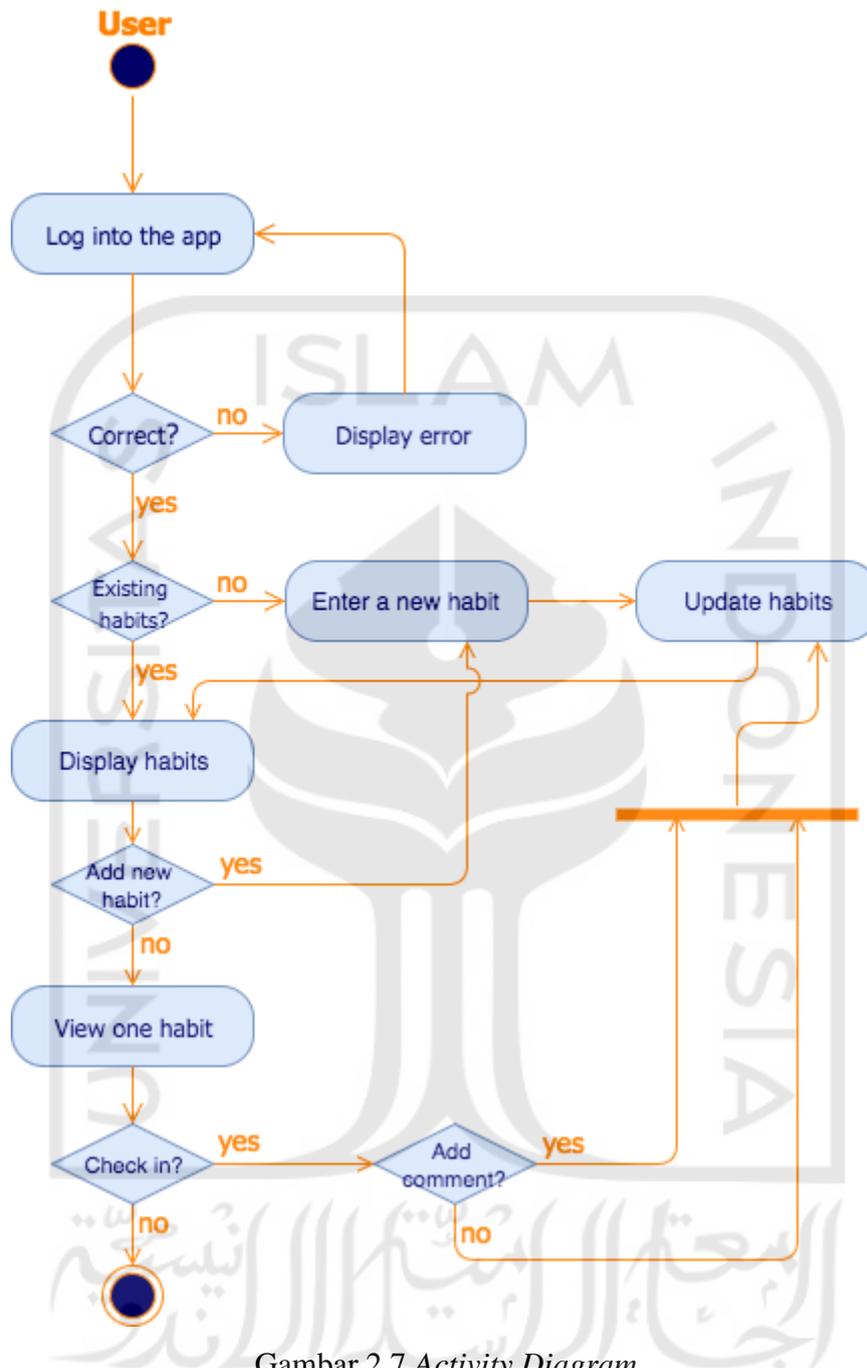
Tipe *relationship* yang mungkin terjadi pada *use case diagram*:

1. *Include*, memiliki syarat yang harus dipenuhi agar event dapat terjadi, yaitu disaat sebuah *use case* adalah bagian dari *use case* lainnya.
2. *Extends*, hanya akan terpenuhi di bawah kondisi tertentu seperti menggerakkan peringatan.
3. *Communicates*, merupakan pilihan selama asosiasi hanya tipe *relationship* antara aktor dan *use case*.

### 2.5.2 *Activity Diagram*

*Activity diagram* merupakan rangkaian aliran dari suatu aktivitas yang berurutan dan dibentuk dalam suatu operasi sehingga dapat digunakan untuk aktivitas lainnya. Diagram ini sangat mirip dengan *flowchart* karean memiliki model yang mengalir dari satu status ke status lainnya. Namun mekanisme pada *activity diagram* dibuat secara paralel. Pembuatan *activity diagram* pada awal pemodelan membantu memahami keseluruhan proses.

Gambar 2.7 merupakan salah satu contoh Activity Diagram yang notasi sebagai berikut:



Gambar 2.7 Activity Diagram

a. *Initial Node*

Berbentuk lingkaran penuh, menggambarkan dimulainya sebuah proses.

b. *Action*

Berbentuk panjang dengan sisi melengkung, menggambarkan setiap tahapan. Urutan tindakannya dibuat sebanyak total aktivitas yang ditampilkan oleh diagram.

c. *Flow*

Tanda panah pada diagram menunjukkan perkembangan dengan tindakan. Kebanyakan tidak disertai kata-kata untuk mengidentifikasinya.

d. *Decision*

Berbentuk belah ketupat dengan 1 aliran masuk dan 2 atau lebih aliran keluar. Aliran yang dibuat untuk mengidentifikasi kondisi.

e. *Merge*

Berbentuk belah ketupat dengan 2 atau lebih aliran masuk dan 1 aliran keluar, digunakan untuk menyatukan yang dipisahkan oleh decision.

f. *Fork*

Berbentuk balok hitam panjang yang memiliki 1 aliran masuk dan 2 atau lebih aliran keluar, yang berperan untuk memecah alur paralel dalam sebuah aktivitas.

g. *Join*

Berbentuk balok hitam panjang yang memiliki 2 atau lebih aliran masuk dan 1 aliran keluar. Seluruh tindakan yang ada harus selesai sebelum melanjutkan proses selanjutnya.

h. *Activity Final*

Berbentuk lingkaran penuh didalam sebuah lingkaran lain yang menandakan berakhirnya suatu Activity Diagram.

## 2.6 Skala Likert

Skala likert merupakan salah satu metode penghitungan terhadap penilaian, sikap, pendapat dan persepsi seseorang dalam sebuah variabel penelitian. Variabel yang diukur menjadi sebuah indikator yang kemudian dijadikan sebagai titik tolak untuk menyusun item instrument berupa pernyataan atau pertanyaan (Sugiyono, 2010).

Jawabannya mempunyai gradasi dari sangat positif sampai sangat negatif kemudian dapat diberi skor, misalnya Sangat Setuju dengan skor 5, Setuju dengan skor 4, Ragu-ragu dengan Skor 3, Tidak Setuju dengan skor 2 dan Sangat Tidak Setuju dengan skor 1 (Sugiyono, 2010).

Tabel 2.1 merupakan contoh perhitungan kuesioner dengan skala likert dalam bentuk *checklist*.

Tabel 2.1 Skala Likert bentuk *checklist*

No	Pertanyaan	Jawaban				
		SS	S	R	TS	STS
1	Sekolah ini akan menggunakan teknologi informasi dalam pelayanan administrasi dan akademik		√			
2	.....					

Kemudian dari kuesioner tersebut diberikan kepada 100 orang karyawan dengan hasil 25 orang menjawab SS, 40 orang menjawab S, 5 orang menjawab R, 20 orang menjawab TS dan 10 orang menjawab STS. Selanjutnya dianalisis dengan menghitung rata-rata pada setiap jawaban responden, dengan perhitungan sebagai berikut:

Jumlah skor SS	= 25 x 5 = 125	
Jumlah skor S	= 40 x 4 = 160	
Jumlah skor R	= 5 x 3 = 15	
Jumlah skor TS	= 20 x 2 = 20	
Jumlah skor STS	= 10 x 1 = 10	
<hr/>		
Total	= 350	(2.1)

Idealnya jika seluruh responden menjawab SS maka skor bernilai  $5 \times 100 = 500$ , sedangkan total skor yang diperoleh pada penelitian adalah 350. Berdasarkan data tersebut maka tingkat persetujuan responden terhadap pertanyaan dari kuesioner diatas adalah  $(350 : 500) \times 100\% = 70\%$  dari yang diharapkan.