

## BAB II

### TINJAUAN PUSTAKA

#### A. Penelusuran Referensi

Penelitian yang dilakukan oleh Ayu Pramyta pada tahun 2008 mengembangkan “Aplikasi Untuk Mengatur Pola Hidup Sehat Berbasis Android”. Aplikasi ini merupakan aplikasi berbasis multimedia yang diperuntukan bagi umum. Dengan aplikasi ini, diharapkan Pengguna aplikasi dapat diberikan wawasan dan informasi mengenai cara pola hidup sehat dan dapat menjaga kesehatan tubuh. Aplikasi ini dibangun dengan menggunakan Eclipse dengan bahasa pemrograman Java. (Ayu Pramyta, 2008)

Panduan gerakan ju-jitsu pada buku besar Institute Ju-jitsu Indonesia pegangan asisten pelatih tahun 2010. (Buku Kurikulum GBPM IJI, 2010)

#### B. Android

Android adalah sebuah sistem informasi untuk sebuah perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middle ware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya *Google Inc.* membeli *Android Inc.* yang merupakan pendatang baru yang menciptakan piranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak dan telekomunikasi, termasuk *Google*, *HTC*, *intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia*.

Pada perilisan perdana android, 5 November 2007, android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, Google merilis kode-kode Android di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler. (Safaat N, 2012)

### **C. Pengertian Aplikasi Android**

Android merupakan OS (Operating System) Mobile yang tumbuh ditengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti Windows Mobile, i-Phone OS, Symbian, dan masih banyak lagi. Akan tetapi, OS yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka. (Hermawan, 2011).

### **D. Konsep Dasar Aplikasi Android**

Android adalah sistem operasi pada gadget dan handphone yang kemampuannya hampir sama dengan pc, dapat mengolah data dan dapat menggunakan internet serta berkomunikasi menggunakan jaringan cellular seperti handphone pada umumnya.

### **E. Arsitektur Aplikasi Android**

Arsitektur yang tersedia pada Android adalah:

1. *Applications* dan *widgets*

*Applications* dan *widgets* ini adalah *layer* di mana kita berhubungan dengan aplikasi saja, di mana biasanya kita download aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di *layer* terdapat aplikasi inti termasuk klien *email*, program SMS, kalender, peta, *browser*, kontak dan lain-lain. Semua aplikasi ditulis menggunakan pemrograman Java.

2. *Application Framework*

Android adalah “*Open Development Platform*” yaitu android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur alarm, dan menambahkan status *notifications*, dan sebagainya. Pengembang memiliki akses penuh menuju *API framework* seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (*reuse*).

Sehingga bisa kita simpulkan *applications framework* ini adalah *layer* di mana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi android, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content providers* yang berupa sms dan panggilan telepon.

Komponen-komponen yang termasuk di dalam *Applications frameworks* adalah sebagai berikut:

- a. *Views*
- b. *Content Provider*
- c. *Resource Manager*
- d. *Notification Manager*
- e. *Activity Manager*

### 3. *Libraries*

*Libraries* ini adalah *layer* di mana fitur-fitur android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan diatas *kernel*, *layer* ini meliputi berbagai *library* C/C++ inti seperti *Libc* dan *SSL*, serta:

- a. *Libraries* media untuk pemutaran media *audio* dan *video*
- b. *Libraries* untuk manajemen tampilan
- c. *Libraries Graphics* mencakup *SGL* dan *OpenGL* untuk grafis 2D dan 3D
- d. *Libraries SQLite* untuk mendukung *database*
- e. *Libraries SSL* dan *WebKit* terintegrasi dengan *web browser* dan *security*
- f. *Libraries LiveWebcore* mencakup modern *web browser* dengan *engine embeded web view*
- g. *Libraries 3D* yang mencakup implementasi *OpenGL ES 1,0 API's*

#### 4. *Android Runtime*

*Layer* yang membuat aplikasi android dapat dijalankan di mana dalam prosesnya menggunakan implementasi *linux*. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi android. Di dalam *Android Runtime* dibagi menjadi dua bagian yaitu:

- a. *Core Libraries*: Aplikasi android dibangun dalam bahasa java, sementara *Dalvik* sebagai *virtualmesinnya* bukan *Virtual Machine* Java, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa java/c yang ditangani oleh *Core Libraries*.
- b. *Dalvik Virtual Machine*: Virtual mesin berbasis resister yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien , di mana merupakan pengembangan yang mampu membuat *linux kernel* untuk melakukan *threading* dan manajemen tingkat rendah.

#### 5. *Linux Kernel*

*Linux kernel* adalah *layer* di mana inti dari *operatingsistem* dari android itu berada. Berisi file-file sistem yang mengatur *system processing, memory, resource, drivers*, dan sistem-sistem operasi android lainnya. *Linux Kernel* yang digunakan android adalah *linux kernel release 2.6*.

## **F. Karakteristik Aplikasi Android**

Menurut Safaat (2012), Android adalah sebuah kumpulan perangkat lunak untuk perangkat mobile yang mencakup sistem operasi, middleware dan aplikasi utama mobile. Android memiliki 4 (empat) karakteristik sebagai berikut:

### **1. Terbuka**

Android dibangun untuk benar-benar terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamera, dan lain-lain. Android menggunakan sebuah mesin virtual yang dirancang khusus untuk mengoptimalkan sumber daya memori dan perangkat keras yang terdapat di dalam perangkat. Android merupakan open source, dapat secara bebas diperluas untuk memasukkan teknologi baru yang lebih maju pada saat teknologi tersebut muncul. Platform ini akan terus berkembang untuk membangun aplikasi mobile yang inovatif.

### **2. Semua Aplikasi dibuat sama**

Android tidak memberikan perbedaan terhadap aplikasi utama dari telepon dan aplikasi pihak ketiga (third-party application). Semua aplikasi dapat dibangun untuk memiliki akses yang sama terhadap kemampuan sebuah telepon dalam menyediakan layanan dan aplikasi yang luas terhadap para pengguna.

3. Memecahkan hambatan pada aplikasi

Android memecah hambatan untuk membangun aplikasi yang baru dan inovatif. Misalnya, pengembang dapat menggabungkan informasi yang diperoleh dari web dengan data pada ponsel seseorang seperti kontak pengguna, kalender, atau lokasi geografis.

4. Pengembangan aplikasi yang cepat dan mudah

Android menyediakan akses yang sangat luas kepada pengguna untuk menggunakan library yang dipergunakan tools yang dapat digunakan untuk membangun aplikasi yang semakin baik. Android memiliki sekumpulan tools yang dapat digunakan sehingga membantu para pengembang dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat.

Google Inc. Sepenuhnya membangun Android dan menjadikan bersifat terbuka (open source) sehingga para pengembang dapat menggunakan Android tanpa mengeluarkan biaya untuk lisensi dari Google dan dapat membangun Android tanpa adanya batasan-batasan. Android Software Development Kit (SDK) menyediakan alat dan Application Programming Interface (API) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman java.

## G. Versi Android

Adapun Versi-versi android yang pernah dirilis adalah sebagai berikut:

### 1. Android Versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam, alarm, voice search (pencarian suara), pengiriman pesan dengan gmail dan pemberitahuan email.

### 2. Android Versi 1.5 (Cupcake)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan android dan sdk (software development kit) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, meng-upload video ke youtube dan gambar ke picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

### 3. Android Versi 1.6 (Donut)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, camcorder dan galeri yang diintegrasikan, CDMA/EVDO, VPN, Gestures dan Text to speech engine.

4. Android Versi 2.0 / 2.1 (Eclair)

Pada 3 Desember 2009 kembali diluncurkan ponsel android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dengan dukungan HTML5.

5. Android Versi 2.2 (Froyo: Frozen Yoghurt)

Pada bulan Mei 2010 Android versi 2.2 diluncurkan. Android inilah yang sekarang sangat banyak beredar di pasaran.

6. Android Versi 2.3 (Gingerbread)

Android versi 2.3 diluncurkan pada Desember 2010, hal-hal yang direvisi dari versi sebelumnya.

7. Android Versi 3.0 (Honeycomb)

Dirilis Februari 2011 sebagai android 3.0 revisi 1 serta android versi 3.0 revisi 2 telah dirilis pada juli 2011.

8. Android Versi 3.1 (Ice Cream Sandwich)

Diumumkan secara resmi pada 10 Mei 2011 diajang Google I/O Developer Conference (San Francisco), pihak Google mengklaim “Android Ice Cream Sandwich” akan dapat digunakan baik di smartphone maupun tablet

9. Android Versi 4.0 (Jelly Bean)

Android versi 4.1 (Jelly Bean), Android Jelly Bean yang diluncurkan pada acara Google I/O lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan input keyboard,

desain baru fitur pencarian, UI yang baru dan pencarian melalui Voice Search yang lebih cepat.

10. Android Versi 4.4 (Kitkat)

Sampai dengan v4.4.4 Dirilis pertama pada tanggal 31 bulan Oktober tahun 2013 di namakan dengan Android kitkat. OS android kitkat memiliki tampilan 100% lebih dinamis dan berbeda total dengan android jelly bean, android kitkat di optimasi pada sisi konsumsi baterai dan kinerja OS lebih cepat ketika dijalankan pada perangkat memiliki spesifikasi lebih rendah, seperti kita tahu jika android jelly bean memiliki kelebihan pada sisi konsumsi baterai yang lebih tinggi dan ketika di jalankan di perangkat yang memiliki versi rendah os ini tidak maksimal.

11. Android Versi 5.0 (Lollipop)

Google akhirnya secara resmi merilis sistem operasi Android versi 5.0 Lollipop. Secara teknis, Android Lollipop diklaim sebagai update sistem operasi terbesar yang pernah dilakukan Google. Konon, perangkat yang menggunakan OS Android Lollipop ini akan mampu berintegrasi antar perangkat seperti smartphone, tablet dan smartwatch berbasis Android. Perangkat smartphone yang beruntung mencicipi sistem operasi ini untuk pertama kali adalah dari keluarga baru Nexus yang akan segera diluncurkan dalam waktu dekat ini, antara lain Nexus 6 dan Nexus 9. (Rosadi, 2015)

## **H. Aplikasi**

Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan tertentu (khusus).

Aplikasi secara umum adalah suatu proses dari cara yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal. Aplikasi (*application*) juga bisa disebut sebagai perangkat lunak (*software*) yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word*, dan *Microsoft Excel*. (Rosadi, 2015)

## **I. Gerak**

Gerak didefinisikan sebagai perubahan tempat atau kedudukan baik hanya sekali maupun berkali-kali. Di dunia sains, gerak memiliki nilai besaran scalar dan vector. Kombinasi dari kedua besaran tersebut dapat menjadi besaran baru yang disebut kecepatan dan percepatan. Gerakan pada sebuah benda umumnya dipengaruhi oleh dua jenis energy, yakni energi potensial dan energi kinetic. Berdasarkan perubahannya gerak dapat dibagi menjadi dua jenis, gerak osilasi dan tidak berosilasi. (Anonim, 2010)

## **J. Ju-jitsu**

Beladiri Ju-jitsu, khususnya aliran Kyushin-Ryu masuk ke Indonesia pada saat sekitar pergolakan Perang Dunia II, yaitu pada tahun 1942 yang dibawa oleh tentara Jepang bernama Ishikawa. Selanjutnya perkembangan di Indonesia terkenal dengan aliran I-Kyushin Ryu.

Ishikawa mewariskan ilmunya kepada R. Soetopo asal daerah Ponorogo yang kemudian menurunkan kepada Drs. Firman Sitompul, Irjen Polisi (Purn) Prof. Drs. DPM. Sitompul, S.H., M.H., M.Min, Drs. Heru Nurcahyo, MM, AKBP. Drs. Heru Winoto. Selanjutnya pengembangan di Indonesia dilanjutkan lagi oleh tokoh-tokoh Ju-jitsan muda lainnya.

Pada tahun 1980 diadakan seminar beladiri yang diadakan oleh Perguruan Tinggi Ilmu Kepolisian (PTIK) Jakarta yang diikuti oleh seluruh perwakilan beladiri yang ada. Hasil yang diperoleh beladiri Ju-jitsu terpilih sebagai salah satu beladiri yang layak untuk dapat dijadikan kurikulum wajib bagi mahasiswa PTIK. Lebih dari itu ketangkasan beladiri Ju-jitsu itu juga mendapat pengakuan dari Kedutaan Besar Jepang, yang pada saat itu hadir sebagai undangan pada seminar beladiri tersebut. Selanjutnya Ju-jitsu menjadi kurikulum wajib bagi mahasiswa PTIK hingga sekarang.

Sebagai informasi, bahwa dalam hal pengembangan kepelatihan di lingkungan Polri, Institut Ju-jitsu Indonesia telah melakukan pendidikan dan latihan di seluruh Polda pada Tahun 1999 yang lalu. Adapun dasar pelatihan dimaksud mengacu kepada :

1. Surat Perintah Kapolri No. Pol. : Sprin/29778/VIII/1999 tanggal 13 Agustus 1999 tentang Penunjukan Kalemdiklat Polri sebagai penanggung jawab dalam menyusun program latihan beladiri Ju-jitsu di lingkungan Polri baik pusat dan wilayah secara bertahap dan di lingkungan pendidikan secara selektif
2. Surat Perintah Kapolri No. Pol. : Sprin/3381/IX/1999 tanggal 22 September 1999 tentang Penunjukan Kalemdiklat Polri untuk

menyelenggarakan pelatihan tenaga pelatih beladiri Ju-jitsu Indonesia, tentang materi metode dan waktu latihan

3. Surat Kapolri kepada para Kapolda No. Pol. B/3534/IX/1999 tanggal 27 September 1999 perihal Pemberitahuan penggantian beladiri Polri dengan beladiri Ju-jitsu.

Disamping itu beladiri Ju-jitsu sangat dirasakan kemanfaatannya oleh instansi lainnya, mengingat banyaknya teknik yang dipelajari dengan segala macam bentuk tantangan yang dapat menyelesaikan perkelahian dengan cepat dan tepat. Sehingga pengembangannya juga dilakukan di beberapa instansi pemerintah, TNI, Polri lainnya seperti : Kantor Sekretariat Negara (Tahun 1987), Kantor Menteri Negara Lingkungan Hidup (1994), Batalion Linud 328 Kostrad (1987), Kopasus (1990), Paspampres (1997), Pasukan Gegana Polri (1987), Dinas Penyelamatan Bawah Air ( Dislambair ) TNI AL dan instansi lainnya.

## **K. Eclipse**

*Eclipse* yang diluncurkan oleh IBM pada tanggal 5 November 2001 merupakan sebuah IDE yang gratis dan *open source* atau yang dapat dikembangkan dan digunakan untuk membangun sebuah program komputer dan dapat dijalankan di semua *platform*. (Satyaputra & Aritonang : 2012)

## L. JAVA

Java merupakan bahasa pemrograman yang dikembangkan dengan menggunakan bahasa C, sehingga pengembang (*programmer*) C tidak mengalami kesulitan beralih ke Java. (Supardi : 2014)

Java diciptakan oleh James Gosling dan Patrick Naughton dalam suatu proyek dari *Sun Microsystems* sekitar tahun 1991. Pada mulanya ingin diberi nama OAK (berasal dari nama pohon yang terdapat pada kantor James Gosling), tetapi karena kata OAK telah ada pada *Sun Microsystems*, maka diberi nama Java (dari inspirasi minum kopi). (Supardi : 2014)

## M. XML (Extensible Markup Language)

XML adalah bahasa *markup* yang digunakan untuk menyimpan data (tidak ada program) dan tidak tergantung dengan *tools* tertentu (seperti *editor*, *dbms*, *compiler*, dsb). XML merupakan suatu bahasa *Markup*. *Markup* yaitu bahasa yang berisikan kode-kode berupa tanda-tanda tertentu dengan aturan tertentu untuk memformat dokumen teks dengan tag sendiri agar dapat dimengerti. Pada android XML digunakan untuk merancang *interface* pada sebuah program yang akan dibuat.

## N. Use Case Diagram

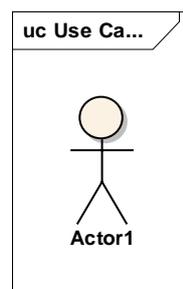
*Use case* diagram merupakan sekelompok simbol dan gambar yang menunjukkan keadaan bagaimana sistem digunakan, apa yang dapat pengguna lakukan terhadap sistem. *Use case* diagram tidak membahas bagaimana fungsi dalam pemrograman berkerja, pada intinya *use case*

diagram menggambarkan interaksi antara pengguna dengan sistem, *use case* diagram mendeskripsikan suatu skenario penggunaan yang bersifat spesifik dalam bahasa yang jelas dan dapat dimengerti dari sudut pandang pengguna (Pressman, 2010).

Dalam beberapa kasus *use case diagram* tidak selalu diperlukan namun penjelasan melalui bentuknya dapat menghasilkan pemahaman yang lebih baik terhadap sistem. Beberapa simbol yang digunakan pada *use case diagram* yaitu :

1. *Actor* (Pengguna)

*Actor* merupakan simbol dari pengguna yang disertakan ke dalam use case diagram yang disesuaikan dengan kebutuhan sistem. Pada Gambar 2.1 menunjukkan simbol actor dalam use case diagram.

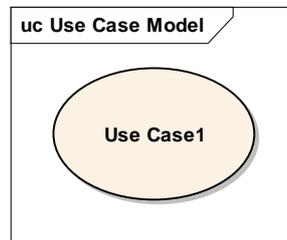


Gambar 2.1 Simbol *actor* dalam *use case*

2. *Use Case*

*Use case* adalah inti simbol dalam *use case* diagram, yang menggambarkan interaksi pengguna dengan sistem yang digambarkan

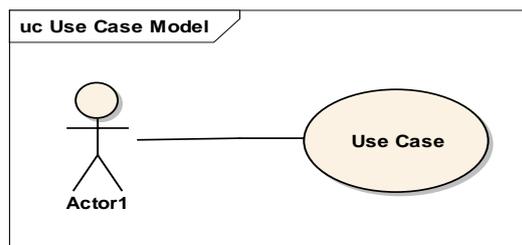
dalam *use case* diagram. Pada Gambar 2.2 menunjukkan simbol *use case* dalam *use case* diagram.



Gambar 2.2 Simbol dari *use case*

3. *Association*

*Association* adalah garis yang menghubungkan antara *actor* dengan *use case*, garis *association* menjelaskan siapa (*actor*) yang berinteraksi terhadap fungsi pada sistem. Pada Gambar 2.3 menunjukkan simbol *association* yang menghubungkan antara *actor* dengan *use case*.

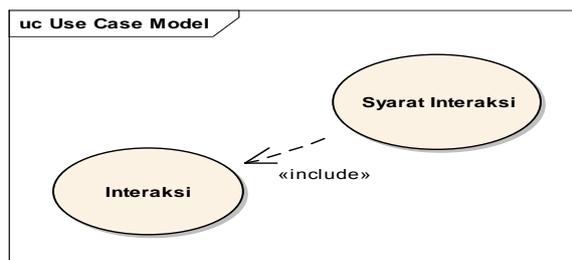


Gambar 2.3 *Actor* dan *use case* dihubungkan dengan *association*

4. *Include*

*Include* adalah garis putus-putus yang menghubungkan antara satu *use case* dengan *use case* lainnya yang menunjukkan bahwa suatu tindakan pengguna kepada aplikasi perlu disertai tindakan lainnya atau *include*

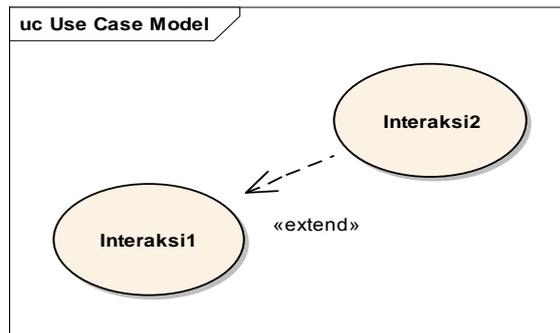
juga dapat diartikan sebagai pemisahan interaksi yang terlalu luas untuk digambarkan hanya dengan 1 *use case* sehingga 1 *use case* yang terlalu luas dapat dibagi menjadi beberapa *use case* yang dihubungkan dengan garis *include* kepada *use case* yang utama. Pada Gambar 2.4 menunjukkan elemen *include* pada *use case* yang menghubungkan *use case* syarat interaksi dengan *use case* interaksi, pada Gambar 2.4 dicontohkan sebuah skema *use case* diagram yang menunjukkan agar perangkat lunak dapat menjalankan keinginan dari *actor* maka *actor* juga perlu melakukan interaksi terhadap *use case* syarat interaksi.



Gambar 2.4 *Include* penghubung antara syarat interaksi kepada interaksi

##### 5. *Extends*

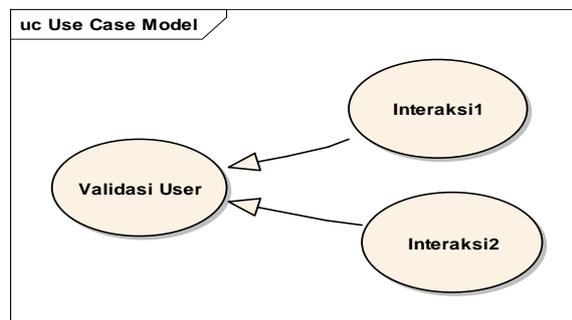
Relasi *use case* tambahan kesebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu. Gambar 2.5 menggambarkan bahwa *use case* interaksi 2 dapat berdiri sendiri walau tanpa *use case* interaksi 1.



Gambar 2.5 Extends pada use case diagram

#### 6. Generalization

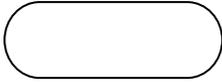
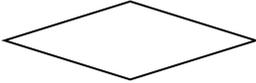
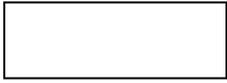
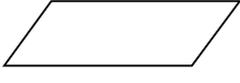
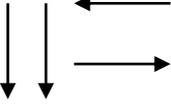
Hubungan generalisasi dan spesialisasi antara dua buah *use case*. Gambar 2.6 menunjukkan generalisasi dari *use case* interaksi 1 dan interaksi 2 dengan *use case* validasi user.



Gambar 2.6 Generalization pada use case diagram

#### O. Flowchart

Sistem *flowchart* dapat didefinisikan sebagai bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem. Simbol-simbol yang digunakan antara lain :

Simbol <i>Flowchart</i>	Fungsi
	<p><b>Terminal</b></p> <p>Simbol ini digunakan untuk mengawali atau mengakhiri suatu proses.</p>
	<p><b>Preparation</b></p> <p>Digunakan untuk mempersiapkan nilai awal dari suatu variabel yang akan diproses dan digunakan untuk proses <i>loop</i>.</p>
	<p><b>Decision</b></p> <p>Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban/aksi.</p>
	<p><b>Proses</b></p> <p>Untuk menggambarkan suatu proses yang sedang dieksekusi.</p>
	<p><b>Input / Output</b></p> <p>Digunakan untuk menggambarkan proses <i>input</i> (<i>read</i>) maupun proses <i>output</i> (<i>print</i>).</p>
	<p><b>Subroutine</b></p> <p>Menggambarkan proses pemanggilan subprogram dari mainprogram.</p>
	<p><b>Flow line</b></p> <p>Merupakan gambaran arus proses dari suatu kegiatan ke kegiatan lain.</p>

***Connector***



Sebagai penghubung antara suatu proses dengan proses lainnya yang ada pada suatu lembar halaman.

---

***Page connector***



Sebagai penghubung antara suatu proses dengan proses lainnya, tetapi berbeda lembar halaman.

---

***Manual operation***



Simbol yang digunakan untuk menggambarkan suatu kegiatan atau proses yang bersifat manual.

---

***Printer***



Menggambarkan suatu dokumen atau suatu kegiatan mencetak suatu informasi.

---

***Console***



Simbol yang digunakan untuk menampilkan data atau informasi melalui monitor atau CRT (*Cathode Ray Tube*).

---

***Disk***



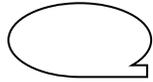
Untuk menggambarkan suatu kegiatan membaca atau menulis data menggunakan media *magnetic disk*.

---

***Manual input***



Menggambarkan proses pemasukan data melalui manual.



***Tape***

Untuk menggambarkan suatu kegiatan membaca atau menulis data menggunakan media *magnetic disk*.

---

Tabel 2.1 Simbol-simbol *flowchart*.

Sumber : (Tata Sutabri, S.Kom, 2004)