BAB II

LANDASAN TEORI

2.1 Pengertian Aplikasi

Aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media (Ridwan).

2.2 Pengertian Game

Game berasal dari kata bahasa inggris yang memiliki arti dasar Permainan. Permainan dalam hal ini merujuk pada pengertian "kelincahan intelektual" (intellectual playability). Game juga bisa diartikan sebagai arena keputusan dan aksi pemainnya. Ada target-target yang ingin dicapai pemainnya. Kelincahan intelektual, pada tingkat tertentu, merupakan ukuran sejauh mana game itu menarik untuk dimainkan secara maksimal. Pada awalnya, game identik dengan permainan anak-anak. Kita selalu berpikir game merupakan suatu kegiatan yang dilakukan oleh anak-anak yang dapat menyenangkan hati mereka.

Dengan kata lain, segala bentuk kegiatan yang memerlukan pemikiran, kelincahan intelektual dan pencapaian terhadap target tertentu dapat dikatakan sebagai game. Tetapi yang akan dibahas pada kesempatan ini adalah game yang terdapat di komputer, baik off line maupun online. Saat ini perkembangan games di komputer sangat cepat. Para pengelola industri game berlomba-lomba untuk menciptakan game yang lebih nyata dan menarik untuk para pemainnya. Hal inilah yang membuat perkembangan games di komputer sangat cepat. Sehingga games bukan hanya sekedar permainan untuk mengisi waktu luang atau sekedar hobi. Melainkan sebuah cara untuk meningkatkan kreatifitas dan tingkat intelektual para penggunanya. Jadi, bermain game adalah suatu proses "fine tuning" (atau penyamaan frekuensi) dari logika berpikir anak-anak kita dengan logika berpikir aplikasi komputer yang canggih tadi. Pada saat bersamaan, game juga secara nyata mempertajam daya analisis para penggunanya untuk mengolah informasi dan mengambil keputusan cepat yang jitu. Namun, tentu saja kenyataan juga harus kita masukkan kedalam perhitungan. Kenyataan itu diantaranya adalah kecanduan para pemain / penggunanya yang akut terhadap permainan komputer semacam ini. Mereka bisa lupa segala-galanya akan tugas mereka yang lain termasuk tugas menuntut ilmu. Menurut Agustinus Nilwan dalam bukunya "Pemrograman Animasi dan Game Profesional" terbitan Elex Media Komputindo, game di artikan sebagai suatu aktivitas tersetruktur atau juga digunakan sebagai alat pembelajaran. Sebuah game bisa dikarakteristikan dari apa pemain lakukan misalnya :

A. Peralatan

Misal : bola, kartu, papan, atau sebuah Komputer.

B. Peraturan

Peraturan digunakan untuk menentukan giliran pemain, hak dan keharusan masing-masing pemain, dan tujuan permainan.

C. Skill, Strategi dan Keberuntungan

Game dengan dengan skill, contohnya dengan kekuatan fisik, misal gulat, menembak dan kekuatan mental seperti catur.

D. Single Player Game (pemain satu orang) dan Double Player (lebih dari satu pemain)

Jika pemain tunggal, pemain harus bermain dengan keahlian, berpacu dengan waktu dan keberuntungan sedangkan pemain double, pemain diharuskan untuk menggunakan suatu strategi dan kekompakan sesama pemain, untuk mencapai tujuan tertentu atau sebaliknya pemain harus berlomba dengan pemain lainnnya untuk mencapai sesuatu tujuan (Risa).

2.3 Pengertian Game TicTacJawa

Game TicTacJawa adalah sebuah permainan yang memakai logika sebagai dasarnya, hampir di semua belahan bumi mengenal permainan ini, mungkin di tempat lain mempunyai nama Tictactoe atau catur jawa, ada pula yang memberi nama permainan ini S.O.S, berbeda nama tidak jauh berbeda dari *game puzzle* lainnya. Cara memainkannya, kita hanya menempatkan icon S atau X dan

O kita agar sejajar dalam 9 titik yang disediakan, akan tetapi tidak semudah yang dibayangkan jika lawan kita memiliki taktik yang lebih bagus dari kita (Bisma Rangga).

2.4 Pengertian Aplikasi Mobile

Kata mobile mempunyai arti bergerak atau berpindah. Sehingga diperoleh pengertian bahwa aplikasi bergerak merupakan aplikasi yang dapat dijalankan walaupun pengguna berpindah atau karena pengguna berpindah.

Pemrograman aplikasi bergerak tidak banyak berbeda dengan pemrograman aplikasi bergerak tidak banyak berbeda dengan pemrograman konvensional pada PC. Aspek karakteristik dari perangkat bergerak sering mempengaruhi arsitektur dan implementasi dari aplikasi tersebut. Dalam pemrograman aplikasi bergerak berbagai aspek teknis perangkat lebih menonjol karena memiliki banyak keterbatasan dibandingkan computer konvensional atau PC. (Budi Darytamo dkk, 2007:1)

Perangkat bergerak (mobile) memiliki banyak jenis dalam hal ukuran, desain dan layout, tetapi memiliki kesamaan karakteristik yang sangat berbeda dari sitem desktop.

1. Ukuran yang kecil

   Perangkat mobile memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.

2. Memory yang kecil

Perangkat mobile juga memiliki memory yang kecil, yaitu primary (RAM) dan secondary (disk). Pembatasan ini adalah salah satu faktor yang mempengaruhi penulisan program untuk berbagai jenis dari perangkat ini. Dengan pembatasan jumlah dari memory, pertimbangan-pertimbangan khusus harus diambil untuk memelihara pemakaian dari sumber daya yang mahal ini.

3. Daya proses yang terbatas

   Sistem mobile tidaklah setangguh rekan mereka yaitu desktop. Ukuran, teknologi dan biaya adalah beberapa faktor yang mempengaruhi status dari sumber daya ini. Seperti harddisk dan RAM, anda dapat menemukan mereka dalam ukuran yang pas dengan sebuah kemasan kecil.

4. Mengkonsumsi daya yang rendah

   Perangkat mobile menghabiskan sedikit daya dibandingkan dengan mesin desktop. Perangkat ini harus menghemat daya karena mereka berjalan pada keadaan dimana daya yang disediakan dibatasi oleh baterai-baterai.

5. Kuat dan dapat diandalkan

   Karena perangkat mobile selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapim benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.

6. Koneksi yang terbatas

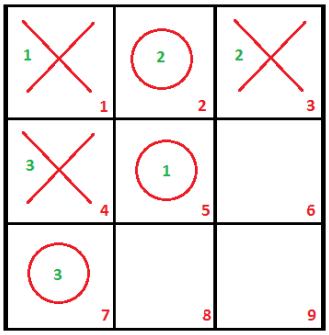
   Perangkat mobile memiliki bandwith rendah, beberapa dari mereka bahkan tidak tersambung. Kebanyakan dari mereka menggunakan koneksi wireless.

7. Masa hidup yang pendek

   Perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan selalu menyala. Coba ambil kasus sebuah handphone, mereka booting dalam hitungan detik dan selalu menyala meski pada malam hari.

2.5 Algoritma Aplikasi

Pada aplikasi TicTacJawa terdapat 2 pelaku, yaitu *Player* 1 dan *Player* 2 dimana mereka masing-masing bersaing untuk mencari kemenangan. Dan untuk memenangkan permainan ketiga objek harus sejajar baik vertical, horizontal ataupun diagonal juga hanya diperbolehkan pindah 1 langkah. Metode algoritma dalam permainan ini diaplikasikan pada permainan yang melibatkan 2 pelaku. Permainan TicTacJawa di deskripsikan dengan sejumlah aturan premis. Dengan itu, kita dapat mengetahui, pada titik-titik tertentu permainan, langkah-langkah yang mungkin berikutnya. Permainan tersebut memberi karakteristik yang sama, yakni "Permainan dengan penuh informasi" Setiap pemain mengetahui semua langkah-langkah yang mungkin dari pemain lawannya.



Gambar 2.1 Contoh Langkah Kemungkinan

Terlihat Pada gambar 2.1, Terdapat 9 buah kotak dimana kotak-kotak tersebut akan di isi dengan 2 bentuk objek yang berbeda dimana 1 objek ditandai dengan X dan objek yang selanjutnya ditandai dengan O, player 1 di tandai dengan objek X dan player 2 dengan O, setiap player dibatasi dengan masing-masing 3 buah objek, dengan permulaan bebas menempatkan objek masing-masing di dalam 9 kotak yang tersedia, jika masing-masing player selesai menempatkan ketiga objeknya masing-masing dan hasilnya belum ada 1 pihak yang memenangkan permainan, maka seperti yang terlihat di gambar 2.1 terdapat objek X 1, X 2 dan X 3, untuk player 1 serta O 1, O 2 dan O 3 untuk player 2, kemudian objek X 1 dikarenakan terkalang oleh O 1, O 2 dan X 3 maka X 1 tidak bisa bergerak dan yang memungkinkan bergerak X 2 ke kotak 6 dan X 3 ke kotak 8, jika yang digerakan X 2 dari kotak 3 ke kotak 6, maka dengan kosongnya kotak 3, objek O 2 dari kotak 2 akan mengisi ke kotak 3, dengan adanya objek O 1 di kotak 5 dan O 3 di kotak 7, maka dengan masuknya objek O 2 ke kotak 3 berakhirlah permainan yang dimenangkan oleh objek O.

2.6 Pengertian Hiburan

Hiburan adalah segala sesuatu yang baik dan berbentuk kata-kata, tempat, benda, perilaku yang dapat menjadi penghibur atau pelipur hati yang susah atau sedih.

Pada umumnya hiburan dapat berupa musik, film, opera, drama, ataupun berupa permainan bahkan olahraga. Berwisata juga dapat dikatakan sebagai upaya hiburan dengan menjelajahi alam ataupun mempelajari budaya. Mengisi kegiatan di waktu senggang seperti membuat kerajinan, keterampilan, membaca juga dapat dikatagorikan sebagai hiburan, bagi orang tertentu yang memiliki sifat workaholic, bekerja adalah hiburan dibandingkan dengan berdiam diri, hiburan bisa dikatakan sebagai cara untuk mengatasi rasa jenuh, demikian pula yang bersifat kepada sesuatu yang menyenangkan untuk seseorang, ada kemungkinan apa yang orang lain anggap sebagai hiburan, justru kita anggap sebagai suatu hal yang hanya bisa menjadikannya sebagai sesuatu yang membuat jenuh dan bosan, akan tetapi sebuah hiburan hanya bisa dirasakan ketika perasaan kita merasakan suatu hal yang nyaman, apapun itu selama membuat perasaan kita nyaman yang akan selalu kita ngin lakukan bisa disebut sebuah hiburan.

2.7 Metode Pendekatan Sistem

Metode pendekatan sistem yang digunakan oleh penulis adalah pendekatan dengan *Object Oriented* yang menggunakan OOAD (*Object Oriented Analisis and Design*).

2.7.1. Pengertian OOAD

OOAD adalah metode analisis yang memeriksa requirements dari sudut pandang kelas kelas dan objek yang ditemui dalam ruang lingkup

permasalahan yang mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek system atau subsistem. OOAD merupakan cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas.

2.7.2. Konsep Dasar OOAD

OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memerika requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup tertentu. Sedangkan OOD adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem.

2.7.2.1 Objek (*Object)*

- Objek adalah benda secara fisik dan konseptual yang ada di sekitar kita. Sebuah objek memiliki keadaan sesaat yang disebut state.
- State dari sebuah objek adalah kondisi dari objek atau himpunan keadaan yang menggambarkan objek tersebut. State dinyatakan dengan nilai dari atribut objeknya.

- Atribut adalah nilai internal suatu objek yang mencerminkan karakteristik objek, kondisi sesaat, koneksi dengan objek lain dan identitas.
- Behaviour (perilaku objek) mendefinisikan bagaimana sebuah objek bertindak dan memberi reaksi. Behaviour ditentukan oleh himpunan semua atau beberapa operasi yang dapat dilakukan oleh objek tersebut, yang dicerminkan oleh interface, service, dan method dari objek tersebut.
- Interface adalah pintu untuk mengakses service dari objek
- Service adalah fungsi yang dapat dikerjakan oleh sebuah objek
- Method adalah mekanisme internal objek yang mencerminkan perilaku objek tersebut

2.7.2.2 Kelas (*Class)*

Class adalah himpunan objek yang sejenis yaitu mempunyai sifat (atribut), perilaku umum (operasi), relasi umum dengan objek lain dan semantik umum. Class adalah abstraksi dari objek dalam dunia nyata. Class menetapkan spesifikasi perilaku dan atribut dari objek tersebut.

2.7.2.3 Kotak Hitam (*Black Box)*

Sebuah objek adalah kotak hitam. Konsep ini menjadi dasar implementasi objek. Dalam operasi OO hanya developer yang dapat memahami detail proses yang ada didalam kotak tersebut, sedangkan

user tidak perlu mengetahui apa yang dilakukan yang penting mereka dapat menggunakan objek untuk memproses kebutuhan mereka. Kotak hitam berisi kode dan data.

- Encapsulation, yaitu proses menyembunyikan detail implementasi sebuah objek. Untuk mengakses data objek tersebut adalah melalui interface. Untuk berkomunikasi dengan objek digunakan message.
- Message adalah permintaan agar objek menerima untuk membawa metode yang ditunjukkan oleh perilaku dan mengembalikan result dari aksi tersebut kepada objek pengirim (sender)

2.7.2.4 Asosiasi dan Agregasi

- Asosiasi adalah hubungan yang mempunyai makna antara sejumlah objek. Asosiasi digambarkan dengan sebuah garis penghubung diantara objeknya. Contohnya : Asosiasi karyawan dengan unit kerja. Setiap karyawan bekerja di satu unit kerja, sedangkan unit kerja dapat memiliki beberapa karyawan.
- Agregasi adalah bentuk khusus sebuah asosiasi yang menggambarkan seluruh bagian pada suatu objek merupakan bagian dari objek yang lain. Contohnya : Kopling dan piston adalah bagian dari mesin, sedangkan mesin, roda, body merupakan bagian dari sebuah mobil.

2.7.3 Pengertian Object

Object adalah sesuatu yang dapat dilihat, disentuh atau dirasakan dan digunakan pengguna serta akan disimpan data dan perilakunya. Bisa berupa:

- Orang, tempat, benda atau kejadian
- Pegawai, pelanggan, guru, dosen, mahasiswa, murid.
- Gudang, kantor, bangunan, ruangan.
- Kendaraan, produk, konputer, video.

2.7.4 Karakteristik Object

- Identitas berarti bahwa data diukur mempunyai nilai tertentu yang membedakan entitas disebut Objek.
- Objek dapat kongkrit, seperti halnya arsip dalam sistem, atau konseptual seperti kebijakan penjadualan dalam multiprocessing pada sistem operasi.
- Setiap objek mempunyai sifat yang melekat pada identitasnya.
- Dua objek dapat berbeda walaupun bila semua nilai atributnya identik.

2.7.4.1 Kelas Object

- Kelas merupakan gambaran sekumpulan Objek yang terbagi dalam atribut, operasi, metode, hubungan, dan makna yang sama.
- Suatu kegiatan mengumpulkan data (atribut) dan perilaku (operasi) yang mempunyai struktur data sama ke dalam satu grup.

- Kelas Objek merupakan wadah bagi Objek. Dapat digunakan untuk menciptakan Objek.
- Objek mewakili fakta/keterangan dari sebuah kelas.

2.7.5 Metodologi Berorientasi Objek

Metodologi adalah cara systematis untuk mengerjakan analisys and design. Dengan metodologi, pihak yang membangun system software dapat merencanakan dan mengulangi pekerjaan dilain waktu. Metodologi juga menghilangkan perbedaan notasi untuk suatu hal yang sama karena setiap oarng akan berbicara dalam bahasa yang sama.

2.7.5.1 Karakteristik Metodologi Berorientasi Objek

1. Encapsulation (pengkapsulan) merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses, data dan prosedur atau fungsi dikemas bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya, data terlindung dari prosedur atau objek lain, kecuali prosedur yang berada dalam objek itu sendiri
2. Inheritance (pewarisan) adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metode dari induknya langsung. Atribut dan metode dari objek induk diturunkan kepada anak objek, demikian seterusnya, Inheritance mempunyai arti bahwa atribut dan operasi akan dimiliki bersama di antara kelas yang

mempunyai hubungan secara hirarki, suatu kelas dapat ditentukan secara umum, kemudian ditentukan spesifik menjadi subkelas. Setiap subkelas mempunyai hubungan atau mewarisi semua sifat yang dimiliki oleh kelas induknya, dan ditambah dengan sifat unik yang dimilikinya.

3. Polymorphism (polimorfisme) yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda, Polimorfisme mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda, kemampuan objek-objek yang berbeda untuk melakukan metode yang sesuai dalam merespon pesan yang sama, seleksi dari metode yang sesuai, bergantung pada kelas yang seharusnya menciptakan objek.

2.8 Teknik Pemodelan OOAD

Model Objek :

- Model objek Menggambarkan struktur statis dari suatu objek dalam sistem dan relasinya
- Model objek berisi diagram objek. Diagram objek adalah graph dimana nodenya adalah kelas yang mempunyai relasi antar kelas.

Model Dinamik

- Model dinamik menggambarkan aspek dari sistem yang berubah setiap saat.
- Model dinamik dipergunakan untuk menyatakan aspek kontrol dari sistem.

- Model dinamik berisi state diagram. State diagram adalah graph dimana nodenya adalah state dan arc adalah tarnsisi antara state yang disebabkan oleh event.

Model Fungsional

- Model fungsional menggambrakan transformasi nilai data di dalam sistem.
- Model fungsional berisi data flow diagram. DFD adalah suatu *graph* dimana *node*nya menyatakan proses dan *arc*nya adalah aliran data.

2.9 Alat Bantu Analisis dan Perancangan

2.9.1 UML (*Unified Modeling Languages*)

Menurut Bambang Hariyanto. Ir.,MT. (2004:17) UML adalah bahasa grafis untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi.

Tujuan UML

1. Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan pertukaran model-model yang berarti.
2. Menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep-konsep inti.

3. Mendukung spesifikasi independent bahasa pemprograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk pemahaman bahasa pemodelan.

2.9.1.1 Bagian-Bagian dari UML

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 belanja, dan sebagainya.

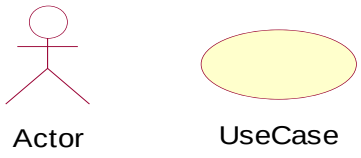
Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

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 behavior- nya sendiri.
- Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain



Gambar 2.2 *Use Case Diagram*

(Munawar. 2005. *Pemodelan visual dengan UML*. Graha Ilmu. Yogyakarta.)

2. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

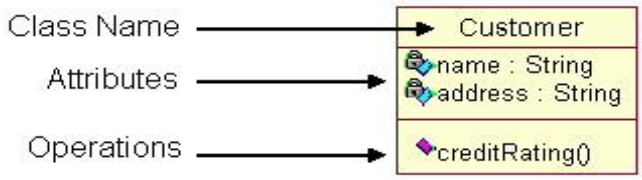
Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti

containment, pewarisan, asosiasi, dan lain-lain. Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

Untuk Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. Private, tidak dapat dipanggil dari luar class yang bersangkutan
2. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
3. Public, dapat dipanggil oleh siapa saja



Gambar 2.3 Bagian-bagian dari *class*

3. Activity Diagram

Activity diagrams 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.

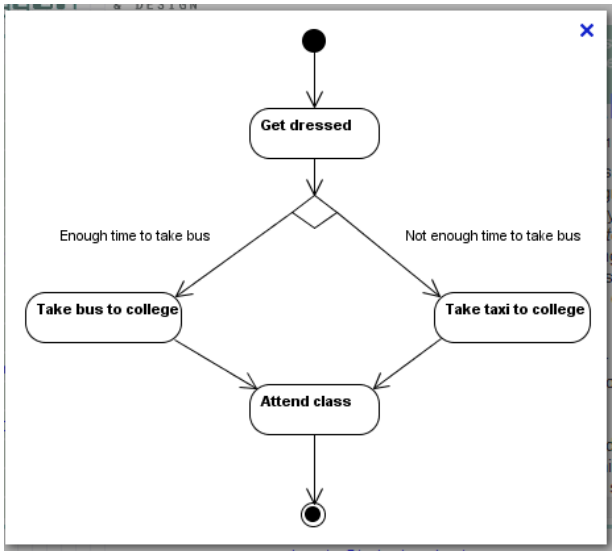
Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-

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. Decision digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (fork dan join) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity diagram dapat dibagi menjadi beberapa object swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.



Gambar 2.4. *Activity Diagram*

(Sumber : http://www.albertoaden.wordpress.com/ msf-fase-planning-planning-phase/ 04 Agustus 2012)

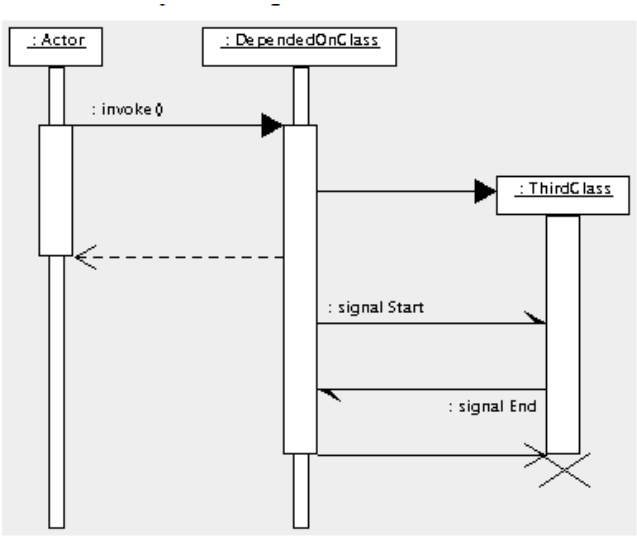
4. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan

perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

- Masing-masing objek, termasuk aktor, memiliki lifeline vertikal.
- Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class.
- Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah massage.
- Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk objek boundary, controller dan persistent entity.



Gambar 2.5. *Sequence Diagram*

(Sumber : http://argouml-stats.tigris.org/documentation/manual-0.32/ch19.html/ 04 juli 2012)

2.9.2 Tools Yang Mendukung UML

2.9.2.1 Rational Rose

Dalam penelitian ini, *Rational Rose* berfungsi sebagai *tool* untuk pemodelan sistem yang menggambarkan proses-proses yang ada pada sistem ini. *Rational Rose* adalah kakas (*tools)* pemodelan visual untuk pengembangan system berbasis objek yang sangat handal untuk digunakan sebagai bantuan bagi para pengembang dalam melakukan analisis dan perancangan sistem.

Rational Rose digunakan untuk melakukan pemodelan sistem sebelum pengembang menulis kode-kode dalam bahasa pemrograman tertentu. Ia juga membantu analisis system dengan cara pengembang membuat diagram *use case* untuk melihat fungsionalitas sistem secara keseluruhan sesuai dengan harapan dan keinginan pengguna. Kemudian, ia juga menuntut pengembang untuk mengembangkan *Interaction* diagram untuk melihat bagaimana objek-objek saling bekerjasama dalam menyediakan fungsionalitas yang diperlukan (Adi Nugroho, 2005:20).

2.10 Pengenalan Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri dan digunakan oleh bermacam perangkat mobile. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan hardware, software, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode–kode Android di bawah lisensi Apache. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar–benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD).

2.10.1 Sejarah Android

Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, di antaranya Andy Rubin, Rich Miner, Nick Sears, dan

Chris White. Saat itu banyak yang menganggap fungsi Android Inc. hanyalah sebagai perangkat lunak pada telepon seluler.

Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler. Versi android terbaru yaitu Android versi 3.0. juga sudah bergabung dengan beberapa smart mobile seperti Nokia, Sony Ericsson, dan lainnya.

2.10.1.1 Produk Awal (2007-2008)

Sekitar September 2007, sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler (Nexus One, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya).

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android yang diantaranya ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat mobile yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan

fitur baru. Telepon seluler pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008.

2.10.1.2 Android 1.1



Gambar 2.6 Android 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.

2.9.1.3 Android 1.5 (Cupcake)



Gambar 2.7 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, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, Bluetooth A2DP support, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

2.10.1.4 Android 1.6 (Donut)



Gambar 2.8 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 dintegrasikan, Gestures, dan Text-to-speech engine, kemampuan dial kontak, teknologi text to change speech

2.10.1.5 Android 2.0/2.1 (Eclair)



Gambar 2.9 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 dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1. Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah Shazam, Backgrounds, dan WeatherBug. Sistem operasi Android dalam situs Internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh MySpace dan Facebook.

2.10.1.6 Android 2.2 (Froyo : Frozen Yoghurt)



Gambar 2.10 Froyo (Frozen Yoghurt)

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi- versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 JavaScript engine yang dipakai Google Chrome. yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

2.10.1.7 Android 2.3 (Gingerbread)



Gambar 2.11 Gingerbread

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

2.10.1.8 Android 3.0 (Honeycomb)



Gambar 2.12 Honeycomb

Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom.3.

2.10.1.9 Android versi 4.0 (Ice Cream Sandwich)



Gambar 2.13 Ice Cream Sandwich

Android versi 4.0 (ICS :Ice Cream Sandwich) Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC.

2.10.1.10 Android 4.1 (Jelly Bean)



Gambar 2.14 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.

2.10.2 Fitur Android

Fitur Android Fitur yang tersedia di Android adalah:

1. Kerangka aplikasi: itu memungkinkan penggunaan dan penghapusan komponen yang tersedia.
2. Dalvik mesin virtual: mesin virtual dioptimalkan untuk perangkat mobile.
3. Grafik: grafik di 2D dan grafis 3D berdasarkan pustaka OpenGL.
4. SQLite: untuk penyimpanan data.
5. Mendukung media: audio, video, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
6. GSM, Bluetooth, EDGE, 3G, dan WiFi (hardware dependent)
7. Kamera, Global Positioning System (GPS), kompas, dan accelerometer (tergantung hardware)

2.10.3 Kelebihan dan Kekurangan Android

2.10.3.1 Kelebihan Android

Beberapa kelebihan dan keunggulan Android adalah :

1. Android bersifat open source, karena berbasis linux. Sehingga banyak developer yang ingin mengembangkan Android.

2. Merupakan realisasi dari Cloud Computing, karena semua fitur Google sudah terintegrasi dengan mobile yang terpasang Android. Contact pada telepon seluler pun akan tersimpan secara otomatis pada account google.
3. Dengan tersedianya Android Market, cara mendapatkan OS Android ini pun terbilang mudah. Fleksibel, karena bisa digunakan pada banyak platform hardware.
4. Android memberikan pilihan untuk memilih Hardware yang digunakan.
5. Android dapat mengubah Pengaturan dengan Cepat
6. Android dapat menjalankan beberapa Aplikasi pada waktu yang bersamaan atau disebut juga Multitasking.
7. Kemudahan dalam Notifikasi, karena setiap ada SMS, Email, atau bahkan artikel terbaru dari RSS Reader, akan selalu ada notifikasi di Home Screen Ponsel Android.

2.10.3.2 Kekurangan Android

Adapun beberapa kekurangan Android adalah :

1. Keterbergantungan pada koneksi internet yang simultan.
2. Kurang tersedianya Aplikasi tambahan pihak ketiga.
3. Content Providers yang menyediakan aplikasi untuk mengakses data dari aplikasi lain (seperti Kontak), atau untuk data sharing.
4. Resource Manager menyediakan akses ke resource non-code seperti grafik dan file layout.

5. Notification Manager yang memungkinkan semua aplikasi untuk menampilkan custom alert pada status bar.
6. Activity Manager yang mengelola live cycle dari aplikasi dan menyediakan navigasi backstack.

2.10.4 Libraries

Android memuat seperangkat C / C++ libraries yang digunakan oleh berbagai komponen Android. Beberapa libraries core-nya adalah :

1. System C library, implementasi turunan BSD dari system C library standar (libc) yang disesuaikan untuk perangkat berbasis Linux.
2. Media Libraries, berbasis Packet Video's OpenCORE. Support pada library playback, rekaman video dan audio, serta file gambar statis, termasuk MPEG4, MP3, AMR, JPG dan PNG.
3. Surface Manager, mengelola akses ke tampilan subsistem.
4. LibWebCore, modern web browser engine yang terintegrasi pada Android dan tampilan web browser.
5. SGL, engine grafis 2D.
6. 3D libraries, implementasi berbasis OpenGL 1.0 API, library menggunakan hardware 3D.
7. FreeType, bitmap dan vektor font rendering.
8. SQLite, database relasional engine yang handal dan ringan tersedia untuk semua aplikasi.

2.10.5 Android Runtime

Setiap aplikasi Android berjalan dalam prosesnya sendiri, contohnya Dalvik Virtual Mesin. Dalvik VM mengeksekisi file dalam format .dex yang dioptimalkan untuk meminimalisir kapasitas memori. VM ini di compile oleh Java Compiler dan ditransformasikan dalam format .dex oleh dx tool.

VM Dalvik bergantung pada kernel Linux untuk fungsionalitas dasar seperti threading dan manajemen memori tingkat rendah.

2.10.6 Linux Kernel Android

Bergatung pada Linux versi 2.6 untuk layanan sistem core seperti keamanan, manajemen memori, manajemen proses, network stack dan driver model. Kernel juga bertindak sebagai layer abstraksi antara hardware dan seluruh software stack.

2.11 Perangkat Lunak yang Digunakan

2.11.1 Eclipse

Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent). Berikut ini adalah sifat dari Eclipse:

- Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
- Mulit-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa

pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.

- Multi-role: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan open source, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan plug-in.

2.11.2 Arsitektur Eclipse

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah kernel, yang mengangkat plug-in. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari plug-in yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan Rich Client Platform (RCP). Berikut ini adalah komponen yang membentuk RCP:

- Core platform
- OSGi
- SWT (Standard Widget Toolkit)
- JFace
- Eclipse Workbench

Secara standar Eclipse selalu dilengkapi dengan JDT (Java Development Tools), plug-in yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (Plug-in Development Environment) untuk mengembangkan plug-in baru. Eclipse beserta plug-in-nya diimplementasikan dalam bahasa pemrograman Java.

Konsep Eclipse adalah IDE yang terbuka (open), mudah diperluas (extensible) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat plug-in CDT(C/C++ Development Tools). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, plug-in UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat plug-in sesuai dengan keinginannya. Salah satu situs yang menawarkan plug-in secara gratis seperti Eclipse downloads by project.

2.11.3 Adobe Illustrator

Adobe Illustrator merupakan salah satu produk keluaran Adobe yang begitu popular dan sering kali digunakan dalam menghasilkan lukisan-lukisan vector. Ia selalu digunakan oleh pereka-pereka grafik untuk menghasilkan lakaran-lakaran seperti kartun, gambar rajah, carta dan logo. Tidak seperti file-file image yang

lain, Illustrator menggunakan persamaan matematik untuk melukis sesuatu bentuk itu. Ini menjadikan grafik dalam bentuk vektor boleh diubah saiz dalam skala yang besar tanpa menjejaskan resolusi sesuatu image itu (image pecah).

Kelebihan Grafik Dalam Format Vektor

Boleh diubah size tanpa menjejaskan resolusi, garisan sentiasa dalam bentuk yang tepat dan kemas, boleh di cetak pada resolusi yang tinggi, size file yang kecil.

Kelamahan Grafik Dalam Format Vektor

Lukisan nampak agak kurang hidup dan lebih kepada kartun. Sukar untuk menghasilkan image yang kelihatan sama seperti objek asal