

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Transfer Data**

Aplikasi adalah suatu sub kelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Contoh utama aplikasi adalah pengolah kata, lembar kerja, memanipulasi foto, merancang rumah dan pemutar media. Beberapa aplikasi yang digabung bersama menjadi suatu paket disebut sebagai suatu paket atau *suite* aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *Open Office.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja dan beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antar muka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Sering kali, mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat ditenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah. (Dahlan Abdullah ; 2013 : 152)

Jenis-jenis Software Aplikasi :

1. *Software* aplikasi hiburan, contohnya yaitu *winamp* untuk mendengarkan musik, *games* dan sebagainya untuk hiburan.
2. *Software* aplikasi pendidikan yaitu *software* digunakan untuk mempelajari atau mereferensikan tentang pendidikan atau pengetahuan.

3. *Software* aplikasi bisnis yaitu *software* yang digunakan untuk aplikasi bisnis
4. *Software* aplikasi khusus
5. *Software* aplikasi untuk produktivitas kerja.

## II.2. *Game* (Permainan)

Teori permainan pertama kali ditemukan oleh sekelompok ahli Matematika pada tahun 1944. Teori itu dikemukakan oleh *John von Neumann and Oskar Morgenstern* yang berisi : “Permainan terdiri atas sekumpulan peraturan yang membangun situasi bersaing dari dua sampai beberapa orang atau kelompok dengan memilih strategi yang dibangun untuk memaksimalkan kemenangan sendiri atau pun untuk meminimalkan kemenangan lawan. (Arizonanataliya Heryanti ; 2013 : 56)

Permainan merupakan bagian dari bermain dan bermain juga bagian dari permainan keduanya saling berhubungan. Permainan dalam hal ini merujuk pada pengertian kelincahan intelektual (*Intellectual Playability Game*) yang juga bisa diartikan sebagai arena keputusan dan aksi pemainnya. Dalam *Game*, ada target - target yang ingin dicapai pemainnya. Permainan adalah kegiatan yang kompleks yang didalamnya terdapat peraturan, bermain dan budaya. Sebuah permainan adalah sebuah sistem dimana pemain terlibat dalam konflik buatan. Disini pemain berinteraksi dengan sistem dan konflik dalam permainan merupakan rekayasa atau buatan. Dalam permainan terdapat peraturan yang bertujuan untuk membatasi perilaku pemain dan menentukan permainan. Mendefinisikan apakah yang dimaksud dengan *Game*, tidak cukup dengan hanya melihat kamus bahasa.

Terdapat banyak makna dalam kata „*Game*“. Yang jelas *Game* secara naluri adalah merupakan bagian dari kehidupan manusia. Makna sekilas dari *Game* memberikan pengertian bahwa *Game* merupakan suatu aktifitas yang tidak dilakukan dengan sungguh-sungguh. Untuk mengetahui apa yang sesungguhnya disebut dengan *Game*, maka paling tidak kita dapat memahaminya dari adanya sejumlah pengertian *Game* yang biasa kita alami dalam kehidupan.(M. Rosidi Zamroni ; 2013 : 489)

### **II.2.1. Jenis Klasifikasikan *Game* Berdasarkan *Platform***

*Game* dapat diklasifikasikan menjadi beberapa jenis, yaitu berdasarkan *platform* yang digunakan, dimensi, dan *genre* dari *game* itu sendiri. *Platform* merupakan kombinasi spesifik dari komponen elektronik atau perangkat keras komputer dengan perangkat lunak yang memungkinkan *game* untuk beroperasi. Berdasarkan *platform* yang digunakan, *game* dapat dibagi menjadi beberapa jenis, diantaranya :

1. *Arcade games*, yaitu *game* yang biasanya memiliki *box* atau mesin yang memang khusus di desain untuk jenis *video games* tertentu, bahkan memiliki fitur yang dapat membuat pemainnya lebih merasa masuk dan menikmati , seperti pistol, kursi khusus, sensor gerakan, sensor injakkan dan stir mobil. *Arcade games* biasanya berada di daerah/tempat khusus, sebagai contoh di Indonesia dikenal dengan sebutan ding-dong.
2. *PC games*, yaitu *game* yang dimainkan menggunakan komputer pribadi.

3. *Console games*, yaitu *game* yang dimainkan menggunakan konsol tertentu, seperti *Playstation 2*, *Playstation 3*, *XBOX 360*, dan *Nintendo Wii*.
4. *Handheld games*, yaitu *game* yang dimainkan di konsol khusus *video game* yang dapat dibawa kemana-mana (*portable*), sebagai contoh *Nintendo DS* dan *Sony PSP*.
5. *Mobile games*, yaitu *game* yang dapat dimainkan khusus untuk *mobile phone* atau PDA. (Mahardika Abdi Prawira Tanjung ; 2012 : 1)

## II.2.2. *Game Dimensi*

Dimensi adalah angka yang berhubungan dengan sifat metrik atau topologi dari suatu objek. Berdasarkan dimensi dari objek-objeknya, *game* dapat dibagi menjadi beberapa jenis, diantaranya :

### 1. *Game 2D*

*Game* dua dimensi atau 2D merupakan suatu konsep dimana semua objek berada pada satu bidang datar. Gerakan pada *game 2D* dibatasi hanya horizontal dan vertikal atau secara koordinat gerakan pemain dibatasi hanya dapat bergerak pada sumbu X dan Y. Pada *game 2D* terdapat dua pergerakan kamera. Pertama adalah kamera statis dimana gambar latar (*background*) dan tempat *game 2D* tidak bergerak sama sekali, contoh dalam jenis ini adalah tetris. Kedua adalah *Side Scrolling* dimana *game* yang kita mainkan mempunyai kamera yang dapat bergeser ke kanan atau ke kiri dengan kecepatan sesuai dengan gerakan dan kecepatan karakter yang kita mainkan/gerakkan pada *game* tersebut, contoh *game*

yang termasuk pada jenis ini adalah *Super Mario Bross*, *Sonic* dan *Megaman*. (Mahardika Abdi Prawira Tanjung ; 2012 : 1)

## 2. *Game 2.5D*

Setelah *game 2D*, muncul *game* dengan tampilan 3D datar (*3D Plane*). *Game* seperti ini bukan 2D tapi tidak juga full 3D. Teori grafik seperti ini disebut dengan 2.5D atau *pseudo-3D*, sedangkan pada istilah *game* lebih dikenal dengan istilah *isometric*, *diametric* atau *trimetric projection*. Biasanya *gameplay* dari *game 2.5D* mirip 2D dimana kita hanya bisa bergerak secara horizontal dan vertikal, namun beberapa objek menggunakan teknik *rendering* secara 3D. Salah satu hal yang membuat *developer game* membuat *game* seperti ini karena pemrosesan 3D secara total memerlukan banyak waktu dan biaya untuk membuatnya. (Mahardika Abdi Prawira Tanjung ; 2012 : 1)

## 3. *Game 3D*

*Game 3D* menggunakan tiga dimensional representasi geometris data (X, Y, Z) yang disimpan dalam komputer untuk keperluan perhitungan dan *rendering* gambar 2D. Dalam sebuah *game 3D*, pemain bisa melihat sebuah objek dari sudut 360°. Terdapat tiga dasar dalam pembuatan sebuah objek 3D, yaitu *3D modeling*, *3D rendering*, dan *3D computer graphics software*. (Mahardika Abdi Prawira Tanjung ; 2012 : 1)

### **II.2.3. Prinsip-Prinsip Dan Komponen Utama Permainan**

#### 1. Prinsip-prinsip Permainan

Adapun prinsip-prinsip permainan menurut Yumarlin MZ (2013:77) adalah sebagai berikut :

- a. Dimainkan dua orang atau lebih secara interaktif
- b. Mempunyai tujuan-tujuan tertentu
- c. Adanya pemenang dalam setiap permainan

#### 2. Komponen Utama Permainan

Menurut Yumarlin MZ (2013:77), menyatakan bahwa setiap permainan harus mempunyai empat komponen utama, yaitu:

- a. Adanya pemain, biasanya lebih dari dua orang
- b. Adanya lingkungan dimana para pemain berinteraksi
- c. Adanya aturan-aturan main,dan
- d. Adanya tujuan tertentu yang ingin dicapai

### **II.2.4. Kelebihan Dan Kekurangan Permainan**

#### 1. Kelebihan permainan

Karakteristik suatu permainan dapat dilihat dari segi warna, desain bentuk, dan cara bermainnya. Selain itu permainan mempunyai kelebihan dan kekurangan. Kelebihan permainan antara lain Yumarlin MZ (2013:77):

- a. Permainan adalah sesuatu yang menyenangkan untuk dilakukan dan sesuatu yang menghibur.
- b. Permainan memungkinkan adanya partisipasi aktif dari siswa untuk belajar.

- c. Permainan dapat memberikan umpan balik langsung.
  - d. Permainan memungkinkan penerapan konsep-konsep ataupun peran-peran ke dalam situasi dan peranan yang sebenarnya di masyarakat.
  - e. Permainan bersifat luwes.
  - f. Permainan dapat dengan mudah dibuat dan diperbanyak.
2. Kekurangan permainan

Sedangkan menurut Yumarlin MZ (2013:77) permainan juga mempunyai kekurangan, antara lain:

- a. Karena asyik, atau karena belum mengenai aturan / teknis pelaksanaan.
- b. Dalam mensimulasikan situasi social permainan cenderung terlalu menyederhanakan konteks sosialnya sehingga tidak mustahil siswa justru memperoleh kesan yang salah.
- c. Kebanyakan permainan hanya melibatkan beberapa orang siswa saja, padahal keterlibatan seluruh siswa / warga belajar amatlah penting agar proses belajar bias lebih efektif dan efisien.

#### **II.2.5. Jenis-Jenis Permainan**

Menurut Yumarlin MZ (2013:78) ada beberapa macam permainan yang memiliki aturan-aturan tertentu dan tujuan tertentu pula. Adapun macam-macam permainan tersebut adalah sebagai berikut :

### 1. Permainan Individual

Permainan ini peserta didik memainkan untuk menguji kemampuan sendiri karena sebagian besar permainan itu dilakukannya sendiri. Peserta didik bermain tanpa menghiraukan apa yang dilakukan oleh peserta didik lain disekitarnya. Contoh permainan individual adalah lompat tali, menyusun *puzzle*, menyusun balok-balok, dsb.

### 2. Permainan Beregu

Permainan beregu ini mempunyai aturan-aturan yang diberikan sebelum permainan dimulai. Aturan permainan harus dimengerti oleh setiap pemain dan bersedia mengikuti aturan permainan.

### 3. Permainan Kooperatif

Permainan ini ditandai dengan adanya kerjasama atau pembagian tugas dan pembagian peran antara peserta didik yang terlibat dalam permainan tersebut untuk mencapai tujuan dari kegiatan bermain. Permainan kerjasama dapat dilihat saat peserta didik mengerjakan suatu proyek atau tugas secara bersama-sama dalam kelompok kecil atau kelompok besar sekaligus. Bermain dengan bekerjasama ini bias dimulai oleh peserta didik sendiri atau dengan arahan dari guru. Permainan ini dapat mengembangkan keterampilan sosial dan konstruktif bagi peserta didik. Dalam permainan ini peserta didik dapat berperan serta dalam usaha untuk belajar memecahkan masalah secara bersamasama.

### 4. Permainan Sosial

Permainan sosial adalah kegiatan bermain peserta didik dengan teman-temannya sendiri. Pada permainan ini peserta didik berpartisipasi dalam kegiatan

bermain dengan peserta didik lainnya sesuai perannya masing-masing yang sudah disepakati sebelumnya. Contohnya seperti permainan polisi dengan pencuri, atau lompat tali beregu.

#### 5. Permainan dengan aturan tertentu

Permainan ini ditandai dengan adanya kegiatan bermain yang menggunakan aturan-aturan tertentu. Dalam permainan ini peserta didik diharapkan dapat bersikap sportif. Contoh dari permainan ini adalah sepak bola, permainan ular tangga, monopoli, gobak sodor, dsb.

### **II.3. *Android***

*Android* adalah sistem operasi untuk telepon seluler yang berbasis *Linux*, yang mencakup *system* operasi, *middleware* dan aplikasi. *Android* tidak terikat ke satu merek telepon seluler. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri hingga dapat digunakan oleh berbagai peranti *mobile*. Beberapa fitur utama dari *Android* antara lain *WiFi hotspot*, *Multi-touch*, *Multitasking*, *GPS*, *support java*, mendukung banyak jaringan (GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, *Bluetooth*, Wi-Fi, LTE, and WiMAX) dan juga kemampuan dasar telepon seluler pada umumnya. (Alicia Sinsuw ; 2013 : 2)

#### **II.3.1. Sistem Operasi *Android***

Adapun sistem operasi *android* menurut *Alicia Sinsum* pada tahun 2013 adalah sebagai berikut :

## 1. *Android OS*

*Android OS* adalah sistem operasi yang berbasis *Linux*, sistem operasi *open source*. Selain *Android Software Development Kit (SDK)* untuk pengembangan aplikasi, *android* juga tersedia bebas dalam bentuk sistem operasi. Hal ini yang menyebabkan *vendor-vendor smartphone* begitu berminat untuk memproduksi *smartphone* dan komputer tablet berbasis *Android*. Saat ini *Android OS* sudah menyebar bukan hanya di *smartphone* saja, tetapi juga di komputer *tablet*.

## 2. *Android SDK (Software Development Kit)*

*Android SDK* adalah *tools API (Application Programming Interface)* yang diperlukan untuk mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman *Java*. Beberapa fitur-fitur *Android* yang paling penting adalah mesin *Virtual Dalvik* yang dioptimalkan untuk perangkat *mobile*, *integrated browser* berdasarkan *engine open source WebKit*, Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3D berdasarkan spesifikasi *opengl ES 1.0* (Opsional akselerasi perangkat keras), kemudian *SQLite* untuk penyimpanan data (*database*). Fitur-fitur *android* lainnya termasuk media yang mendukung *audio*, *video*, dan gambar, juga ada fitur *bluetooth*, *EDGE*, *3G* dan *WiFi*, dengan fitur kamera, *GPS*, dan kompas. Selanjutnya fitur yang juga turut disediakan adalah lingkungan *Development* yang lengkap dan kaya termasuk perangkat emulator, *tools* untuk debugging, profil dan kinerja memori, dan *plugin* untuk *IDE Eclipse*.

### 3. AVD (*Android Virtual Device*)

*Android Virtual Device* merupakan emulator untuk menjalankan aplikasi *android*. Setiap AVD terdiri dari sebuah profil perangkat keras yang dapat mengatur pilihan untuk menentukan fitur *hardware* emulator. Misalnya, menentukan apakah menggunakan perangkat kamera, apakah menggunakan *keyboard* QWERTY fisik atau tidak, berapa banyak memori internal, dan lain-lain. AVD juga memiliki sebuah pemetaan versi *Android*, maksudnya kita menentukan versi dari *platform Android* akan berjalan pada emulator. Pilihan lain dari AVD, misalnya menentukan skin yang kita ingin gunakan pada emulator, yang memungkinkan untuk menentukan dimensi layar, tampilan, dan sebagainya. Kita juga dapat menentukan *SD Card virtual* untuk digunakan dengan di emulator.

## II.4. Jaringan

Pengertian jaringan komputer merupakan sekumpulan komputer serta perangkatperangkat lain pendukung komputer yang saling terhubung dalam suatu kesatuan. Media jaringan komputer dapat melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling melakukan pertukaran informasi seperti dokumen dan data. Dapat juga melakukan pencetakan pada printer yang sama dan bersama-sama memakai perangkat keras dan perangkat lunak yang terhubung dengan jaringan. (Choirul Muallifah ; 2013 : 2)

#### **II.4.1. Jenis-Jenis Jaringan**

Berdasarkan jangkauan area atau lokasi jaringan menurut Choirul Muallifah tahun 2013 dibedakan menjadi 3 jenis, yaitu :

1. LAN (*Local Area Network*)

Merupakan jaringan lokal yang dibuat pada area tertutup. Misalkan dalam satu gedung atau dalam satu ruangan. LAN biasa digunakan untuk jaringan kecil yang menggunakan resourcebersama. Seperti penggunaan printer secara bersama. LAN dapat menggunakan media komunikasi seperti kabel dan *wireless*.

2. MAN (*Metropolitan Area Network*)

Merupakan jaringan antara LAN satu dengan LAN lain yang dipisahkan daerah lokasi yang cukup jauh. Contoh penggunaan MAN adalah hubungan antara kantor pusat dengan kantor cabang yang ada di daerah-daerah. Dapat dikatakan MAN merupakan pengembangan dari LAN.

3. WAN (*Wide Area Network*)

Merupakan jaringan yang cakupannya lebih luas dari pada MAN. Cakupan WAN meliputi satu kawasan, satu negara, satu pulau, bahkan satu benua. Metode yang digunakan WAN hamper sama dengan LAN dan MAN.

#### **II.4.2. Manfaat dan Keuntungan Jaringan**

Adapun manfaat dan keuntungan membangun jaringan menurut Choirul Muallifah tahun 2013 adalah sebagai berikut :

1. Manfaat Jaringan Komputer:
  - a. *Sharing resources*
  - b. Media komunikasi
  - c. Integrasi data
  - d. Pengembangan dan pemeliharaan
  - e. Keamanan data
  - f. Sumber daya lebih efisien dan informasi terkini
2. Keuntungan Membangun Jaringan Komputer Yaitu:
  - a. Dapat berbagi peralatan (*peripheral*) dan penggunaanya, seperti printer, *harddisk*, modem.
  - b. Memudahkan bertukar data diantara pengguna komputer tanpa harus menggunakan disket.
  - c. Kita dapat menggunakan programprogram yang ada di komputer pusat.
  - d. Bisa mengirim dan menerima *email* dari *internet* dan mencari informasi lain melalui fasilitas *internet*.

## **II.5. Bluetooth**

Menurut Feri (Feri Djuandi, 2011)Arduino adalah sebuah physical computing yang bersifat open source. Dikutip dari artikel yang ditulis oleh Tim Diytech (Tim DiyTech, 2012)dijelaskan bahwa Bluetooth adalah teknologi komunikasi tanpa kabel yang menyediakan layanan komunikasi secara real-timeantar perangkat Bluetoothdengan jarak layanan yang lebih jauh dari media infra merah. Teknologi Bluetooth banyak digunakan sebagai media pertukaran

data pada berbagai perangkat smartphonetermasuk Android. Tertulis pada artikel yang ditulis Tim Wikipedia (Tim Wikipedia, 2008)Android merupakan sistem operasi milik Google berbasis Linux yang dirancang untuk smartphonedan tablet dengan layar sentuh. Bluetooth bekerja menggunakan frekuensi radio. Beda dengan inframerah yang mendasarkan diri pada gelombang cahaya. Jaringan Bluetooth bekerja pada frekuensi 2.402 Giga Hertz sampai dengan 2.480 Giga Hertz. Dibangkitkan dengan daya listrik kecil sehingga membatasi daya jangkauhanya hanya sampai 10 meter. Penetapan frekuensi ini telah distandardisasi secara *internasional* untuk peralatan elektronik yang dipakai untuk kepentingan industri, ilmiah, dan medis. Kecepatan *transfer* data *Bluetooth* rilis 1.0 adalah 1 mega bit per detik (Mbps), sedangkan versi 2.0 mampu menangani pertukaran data hingga 3 Mbps. Kalau dalam satu waktu bisa terjadi koneksi antara 8 peralatan *Bluetooth* secara simultan, lalu bagaimana bisa peralatan *Bluetooth* tidak mengganggu satu sama lain? Hal itu disebabkan karena masing-masing peralatan tersebut membangkitkan sinyal sangat lemah melalui listrik berdaya 1 miliwatt yang akan mengacak penggunaan 79 frekuensi sebanyak 1600 kali dalam satu detik. Jadi, akan sangat kecil kemungkinan masing-masing alat menggunakan frekuensi yang sama dalam satu waktu. Sepasang peralatan *Bluetooth* yang telah tersambung akan membentuk *Personal Area Network*, disebut juga piconet dan mengacak frekuensi. Akan terjadi transaksi dan percakapan antar peralatan secara otomatis apakah ada data yang hendak dipertukarkan dan pihak manakah yang akan mengontrol komunikasi

## II.6. Java

*Java* memiliki cara kerja yang unik dibandingkan dengan bahasa perograman lainnya yaitu bahasa perograman *java* bekerja menggunakan *interptreter* dan juga *compiler* dalam proses pembuatan program, *Interpreter java* dikenal sebagai perograman *bytecode* yaitu dengan cara kerja mengubah paket *class* pada *java* dengan *extensi*. *Java* menjadi *.class*, hal ini dikenal sebagai *class bytecode*, yaitunya *class* yang dihasilkan agar program dapat dijalankan pada semua jenis perangkat dan juga *platform*. (Defni, Indri Rahmayun ; 2014 : 64)

## II.7. Data Flow Diagram (DFD)

Data *Flow* Diagram (DFD) adalah alat pembuatan model yang memungkinkan professional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama *Bubble chart*, *Bubble diagram*, model proses, diagram alur kerja, atau model fungsi.

DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.

DFD ini merupakan alat perancangan system yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa

maupun rancangan ,sistem yang mudah dikomunikasikan oleh professional sistem kepada pemakai maupun pembuat program. (Dahlan Abdullah ; 2013 : 154)

## **II.8. UML (*Unified Modelling Language*)**

*Unified Modelling Language* (UML) adalah sebuah "bahasa" yg 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 objek seperti C++, *Java*, C# atau *VB.NET*. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: *Grady Booch OOD (Object-Oriented Design)*, Jim Rumbaugh *OMT (Object Modeling Technique)*, dan Ivar Jacobson *OOSE (Object-Oriented Software Engineering)*. Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi

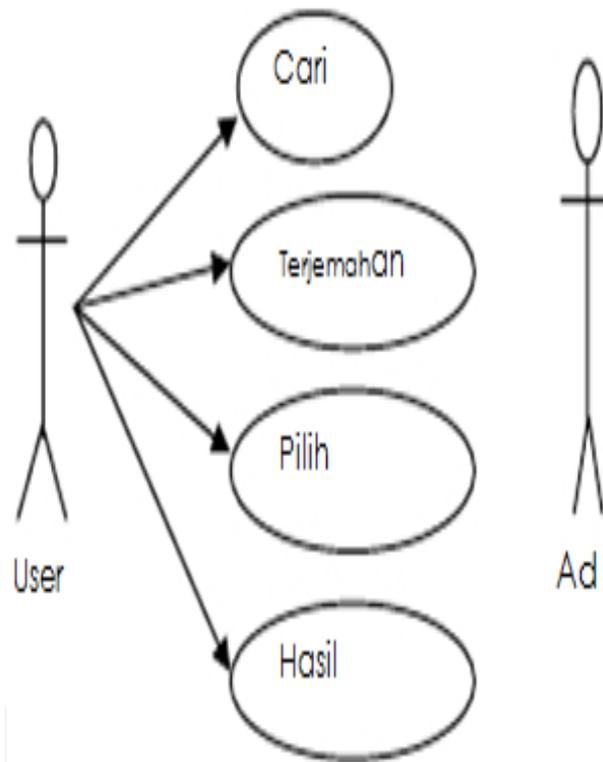
pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: *metodologi booch*, *metodologi coad*, *metodologi OOSE*, *metodologi OMT*, *metodologi shlaer-mellor*, *metodologi wirfs-brock*, dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan. Dimulai pada bulan Oktober 1994 *Booch*, *Rumbaugh* dan *Jacobson*, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan memelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease *draft* pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh *Object Management Group* (OMG – <http://www.omg.org>). Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003. *Booch*, *Rumbaugh* dan *Jacobson* menyusun tiga buku serial tentang UML pada tahun 1999. Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek. (Yuni Sugiarti ; 2013 : 33)

Dalam pembuatan skripsi 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 belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang common. Sebuah *use case* juga dapat meng-*extend* *use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. (Yuni Sugiarti ; 2013 : 41)

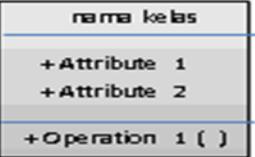


**Gambar II.1. Use Case Diagram**

(Sumber : Junaedi Siregar ; 2013 : 76)

## 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 :

Simbol	Deskripsi
 <p>Package</p>	Package merupakan sebuah bungkus dari satu atau lebih kelas
 <p>Operasi</p> <p>nama kelas</p> <p>+ Attribute 1</p> <p>+ Attribute 2</p> <p>+ Operation 1 ( )</p>	Kelas pada struktur sistem
 <p>Antarmuka / interface</p> <p>interface</p>	sama dengan konsep interface dalam pemrograman berorientasi objek
 <p>Asosiasi</p> <p>1 1..*</p>	relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
 <p>Asosiasi berarah/directed asosiasi</p>	relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
 <p>Generalisasi</p>	relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
 <p>Kebergantungan / defedency</p>	relasi antar kelas dengan makna kebergantungan antar kelas
 <p>Agregasi</p>	relasi antar kelas dengan makna semua-bagian (whole-part)

**Gambar II.2. Class Diagram**

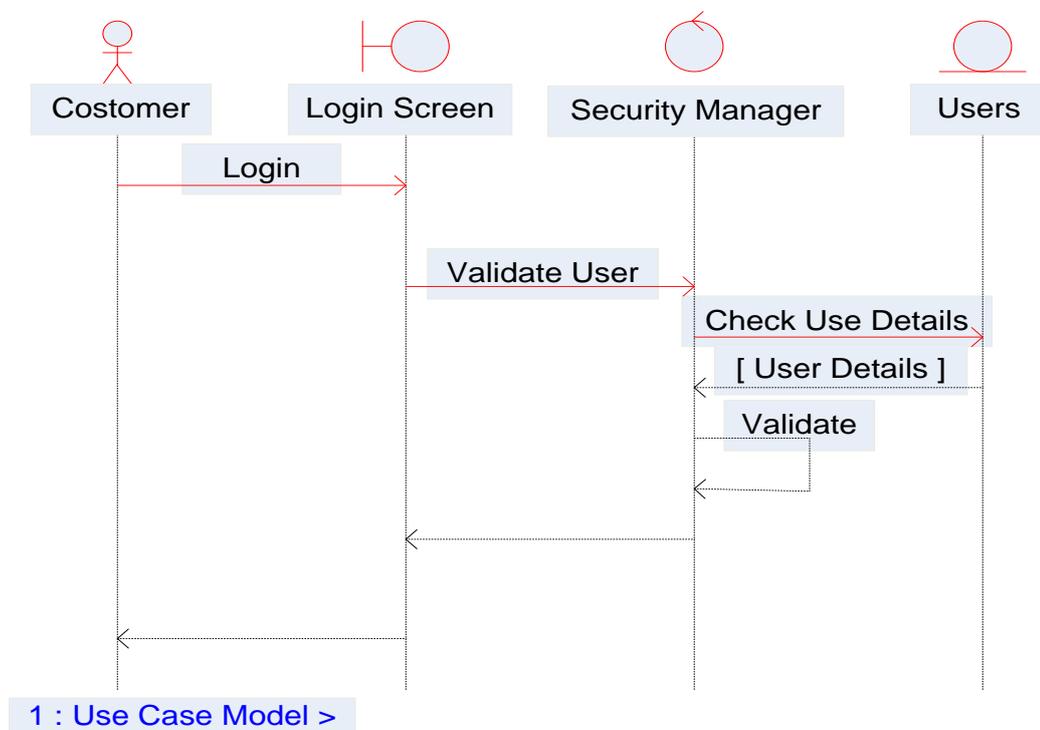
(Sumber : Yuni Sugiarti ; 2013 : 59)

### 3. Sequence Diagram

Diagram *Sequence* menggambarkan kelakuan/prilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk 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.

Banyaknya diagram *sequence* yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya 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.3. Contoh Sequence Diagram**

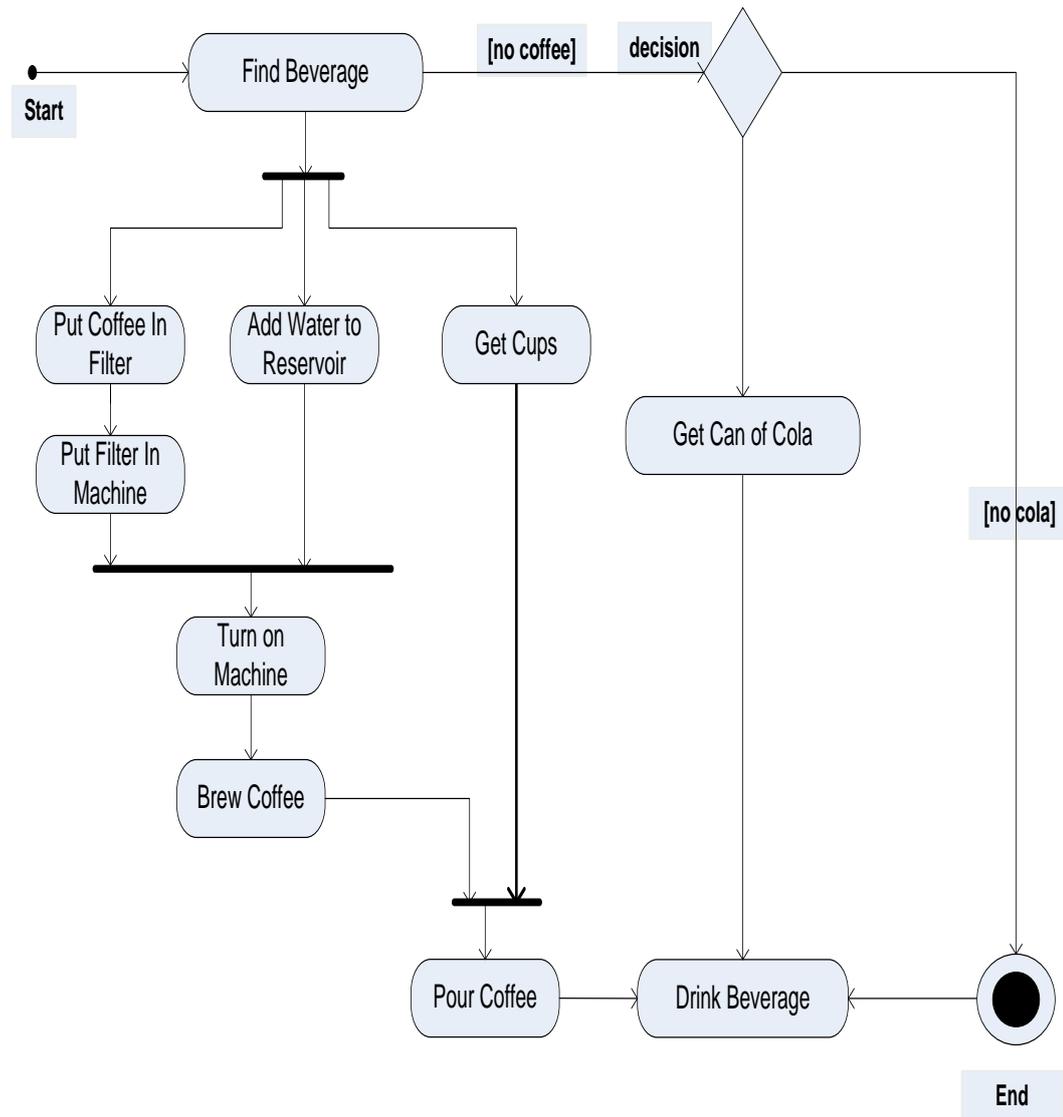
(Sumber : Yuni Sugiarti ; 2013 : 63)

#### 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, 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 II.4. Activity Diagram**

*(Sumber : Yuni Sugiarti ; 2013 : 76)*