

BAB II

LANDASAN TEORI

II.1 Citra

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran dari suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan. (Sutoyo, 2009:9)

Citra merupakan suatu keluaran dari suatu sistem perekam data yang bersifat optik, analog ataupun digital. Perekaman data citra dapat dibagi menjadi dua yaitu:

1. Citra Analog

Citra analog yaitu terdiri dari sinyal-sinyal elektromagnetik yang tidak dapat dibedakan sehingga pada umumnya tidak dapat ditentukan ukurannya. Citra analog mempunyai fungsi yang kontinu. Hasil perekaman citra analog dapat bersifat optik yakni berupa foto (film foto konvensional) dan bersifat sinyal video seperti gambar pada monitor televisi.

2. Citra Digital

Citra digital terdiri dari sinyal-sinyal yang dapat dibedakan dan mempunyai fungsi yang tidak kontinu yakni berupa titik-titik warna

pembentuk citra. Hasil perekaman citra digitak dapat disimpan pada suatu media magnetik.

II.1.1 Format Citra Digital

Format file citra standar yang digunakan saat ini terdiri dari beberapa jenis. Format-format ini digunakan dalam menyimpan citra dalam jenis. Format-format ini digunakan dalam menyimpan citra dalam sebuah file. Setiap format memiliki karakteristik masing-masing. Berikut adalah penjelasan beberapa format umum digunakan saat ini. (Darma, 2010:58)

1. Bitmap (.bmp)

Format .bmp adalah format penyimpanan standar tanpa kompresi yang umum dapat digunakan untuk menyimpan citra biner hingga citra berwarna. Format ini terdiri dari beberapa jenis yang setiap jenisnya ditentukan dengan jumlah bit yang digunakan untuk menyimpan sebuah nilai pixel.

2. Tagged Image Format (.tif, .tiff)

Format .tif merupakan format penyimpanan citra yang dapat digunakan untuk menyimpan citra bitmap hingga citra dengan warna gambar terkompresi. Format ini dapat digunakan untuk menyimpan citra yang tidak terkompresi dan juga citra terkompresi.

3. Portable Network Graphics (.png)

Format .png adalah format penyimpanan citra terkompresi. Format ini dapat digunakan pada citra grayscale, citra dengan palet warna, dan juga

citra fullcolor. Format .png juga mampu menyimpan informasi hingga kanal alpha dengan penyimpanan sebesar 1 hingga 16 bit per kanal.

4. JPEG (.jpg)

Format .jpg adalah format yang sangat umum digunakan saat ini khususnya untuk transmisi citra. Format ini digunakan untuk menyimpan citra hasil kompresi dengan metode JPEG.

5. MPEG (.mpg)

Format ini digunakan di dunia internet dan diperuntukkan sebagai format penyimpanan citra bergerak (video).

6. Graphics Interchange Format (.gif)

Format ini dapat digunakan pada citra warna dengan palet 8 bit. Penggunaan umumnya pada aplikasi web. Kualitas yang rendah menyebabkan format ini tidak terlalu populer di kalangan peneliti pengolahan citra digital.

7. RGB (.rgb)

Format ini merupakan format penyimpanan citra yang dibuat oleh silicon graphics untuk menyimpan citra berwarna.

8. RAS (.ras)

Format .ras digunakan untuk menyimpan citra dengan format RGB tanpa kompresi.

9. Postscript (.ps, .eps, .spfs)

Format ini diperkenalkan sebagai format untuk menyimpan citra buku elektronik. Dalam format ini, citra direpresentasikan ke dalam deret nilai desimal atau hexadesimal yang dikodekan ke dalam ASCII.

10. Portable Image File Format

Format ini memiliki beberapa bagian di antaranya adalah portable bitmap, portable graymap, portable pixmap, dan portable network map dengan format berturut-turut adalah .pbm, .pgm, .ppm dan .pnm. Format ini baik digunakan untuk menyimpan dan membaca kembali data citra.

11. PPM

PPM terdiri dari dua bagian umum yaitu bagian pendahuluan dan bagian data citra. Bagian pendahuluan memiliki tiga bagian kecil, yang pertama adalah pengenalan PPM yang dapat berupa p3 (untuk citra ASCII) dan p6 (untuk citra biner). Bagian pendahuluan yang kedua adalah ukuran panjang dan lebar citra. Bagian ketiga dari pendahuluan adalah nilai maksimum dari komponen warna. Keistimewaannya adalah dalam data citra dapat disimpan komentar dengan memberikan tanda '#' sebelum komentar.

12. PGM

Format ini hampir mirip dengan format PPM hanya saja format ini menyimpan informasi grayscale (satu nilai per pixel). Perngenal yang digunakan adalah p2 dan p5.

13. PBM

PBM digunakan untuk menyimpan citra biner. Hampir sama dengan PPM dan PGM, format PBM ini memiliki pendahuluan, hanya saja pendahuluannya tidak memiliki bagian ketiga (penjelasan nilai maksimum pixel). Penggenap yang digunakan adalah p1.

II.1.2 Elemen – elemen Citra Digital

Berikut adalah elemen – elemen citra digital (Sutoyo.T, dkk, 2009:24):

1. Kecerahan (*brightness*)

Kecerahan (*brightness*) merupakan intensitas cahaya yang di pancarkan pixel dari citra yang dapat ditangkap oleh sistem penglihatan. Kecerahan pada sebuah titik (pixel) di dalam citra merupakan intensitas rata-rata dari suatu area yang melingkupinya.

2. Kontras (*contrast*)

Kontras (*contrast*) menyatakan sebaran terang gelap dalam sebuah citra. Pada citra yang baik, komposisi gelap dan terang tersebar secara merata.

3. Kontur (*contour*)

Kontur (*contour*) adalah keadaan yang ditimbulkan oleh perubahan intensitas pada pixel – pixel yang bertetangga. Karena adanya perubahan intensitas inilah mata mampu mendeteksi tepi-tepi objek di dalam citra.

4. Warna

Warna sebagai persepsi yang ditangkap sistem visual terhadap panjang gelombang cahaya yang dipantulkan oleh objek.

5. Bentuk (*shape*)

Bentuk (*shape*) adalah properti intrinsik dari objek 3 dimensi, dengan pengertian bahwa bentuk merupakan properti intrinsik utama untuk sistem visual manusia.

6. Tekstur (*texture*)

Tekstur (*texture*) dicirikan sebagai distribusi spasial dari derajat keabuan di dalam sekumpulan pixel – pixel yang bertetangga. Tekstur adalah sifat – sifat atau karakteristik yang dimiliki oleh suatu daerah yang cukup besar sehingga secara alami sifat – sifat tadi dapat berulang dalam daerah tersebut. Tekstur adalah keteraturan pola – pola tertentu yang terbentuk dari susunan pixel – pixel dalam citra digital, Informasi tekstur dapat digunakan untuk membedakan sifat – sifat permukaan suatu benda dalam citra yang berhubungan dengan kasar dan halus, juga sifat – sifat spesifik dari kekasaran dan kehalusan permukaan tadi, yang sama sekali terlepas dari warna permukaan tersebut.

II.2 Foto

Foto merupakan istilah lain dari potret. Secara pengertian foto adalah gambar yang dibuat dengan kamera dan peralatan fotografi lainnya. Selain itu foto dan potret juga sering digunakan sebagai kiasan. Misalnya: “Foto/potret masa silam itu sering muncul kembali dalam benaknya”. Dalam hal ini, foto/ potret berarti bayangan, gambaran, atau kenangan.

Selain definisi diatas dan makna kiasan yang dapat diikaskan dengan kata foto. Secara kategorisasi foto juga harus dibedakan menjadi beraga. Kategorisasi ini bertujuan untuk memudahkan pembuatan dan pemanfaatannya, sesuai dengan standar kualitas bagi masing-masing keperluan. Ada banyak sekali kategori foto, antara lain: foto keluarga, foto dokumentasi, foto resmi, foto salon, foto seni, foto kriminal, foto porno, foto kedokteran (foto sinar X/rontgen), foto infra merah, foto bawah taut, foto satelit, foto udara, foto mikro, foto jurnalistik, dan lain-lain.

Selain itu, ada pula kategori foto berdasarkan ukuran. Misalnya pas foto, foto seluruh badan, foto KTP, paspor, foto postcard, dan lain-lain. Ada pula pembedaan sesuai dengan jenis kameranya. Misalnya foto analog (dengan film) dan foto digital. (Sumber: <http://www.gudangmateri.com/2011/06/pengertian-foto.html>)

II.3 Sejarah Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc, membeli Android Inc. Yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Hendset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia (Safaat, 2011:1).

Versi-versi Android yang pernah dirilis adalah sebagai berikut (Safaat, 2011: 11) :

1. Android versi 1.1

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

2. Android versi 1.5 (Cupcake)

Terdapat beberapa pembaruan termasuk juga penampilan beberapa fitur dalam versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem. Dirilis pada pertengahan Mei 2009.

3. Android versi 1.6 (Donut)

Donut (versi 1.6) dirilis pada September 2009 dengan menampilkan proses pencarian yang lebih baik dibandingkan sebelumnya, penggunaan baterai indikator dan control applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan gelri yang diintegrasikan; CDMA/EVDO, VPN, Gestures, dan Text-to-speech engine, kemampuan dial kontak; pengadaan resolusi WVGA.

4. Android versi 2.0/2.1 (Éclair)

Android ini diluncurkan pada 3 Desember 2009. Dilakukan perubahan, yaitu pengoptimalan *hardware*, perubahan *User Interface* (UI) dengan *browser* baru dan dukungan *HTML5*, daftar kontak yang baru, peningkatan Google Maps 3.1.2, dukungan *flash* untuk kamera 3,2 MP, *digital Zoom*, dan *Bluetooth 2.1*.

5. Android versi 2.2 (Froyo / Frozen Yoghurt)

Pada 20 Mei 2010 kembali diluncurkan ponsel *Android* dengan versi 2.2 (*Froyo*) perubahan yang dilakukan meliputi optimasi kecepatan, memori, dan kinerja sistem operasi secara keseluruhan, dukungan untuk menginstal aplikasi pada memori eksternal, dukungan *Adobe Flash 10.1* serta fungsi *USB tethering* maupun *Wi-Fi hotspot*.

6. Android versi 2.3 (Gingerbread)

1 Desember 2010, *Google* kembali meluncurkan versi terbaru yaitu *Android* versi 2.3. Pada versi ini terdapat peningkatan manajemen daya, control melalui aplikasi, penggunaan *multiple camera*, peningkatan *performa* serta penambahan sensor seperti *gyroscope*.

7. Android versi 3.0/3.1 (Honeycomb)

Versi ini berbeda dengan versi-versi sebelumnya. Versi ini dirancang khusus untuk *PC Tablet* sehingga memiliki *User Interface* yang berbeda dan mendukung ukuran layar yang lebih besar. Selain itu, pada versi ini memungkinkan penggunaan *multiprosesor* dan akselerasi

perangkat keras untuk grafis. SDK versi pertama pertama diluncurkan Februari 2011.

8. 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*.

II.3.1 Fitur dan Arsitektur Android

Fitur yang tersedia pada Android antara lain (Winarno, 2011:12):

1. Multiprocess dan App Widgets

OS nadroid tidak membatasi prosesor ke satu programsaja. Tapi sistem bisa mengatur prioritas aplikasi dan thread dalam satu aplikasi. Ini memungkinkan program background bisa dijalankan ketika user membiarkan peranti berjalan di background, misalnya ketika sedang main game. Sementara di belakngnya ada program yang tetap berjalan.

App Widget merupakan aplikasi mini yang bisa di- embed di aplikasi lain, seperti Home Screen. App Widget bisa memproses event, seperti memulai streaming audio, atau meng-update temperatur luar, sambil aplikasi lain tetap terus berjalan.

2. Touch, Gesture, dan Multitouch

Touch screen adalah antarmuka user yang sangat intuitif untuk peranti hand held.

Multitouch memberikan cara untuk berinteraksi dengan touch screen di lebih dari satu tangan pada saat yang bersamaan.

3. Keyboard Hard dan Soft

Peranti android juga memiliki keyboard, keyboard ini ada 2 jenis, yaitu hard/fisik dan soft yang berupa software.

Sistem operasi Android dibangun berdasarkan *kernel Linux* dan memiliki arsitekur sesuai dengan Gambar 2.1.



(Sumber: <http://developer.android.com/guide/basics/what-is-android.html>)

Gambar 2.1 Arsitekur Android

II.3.2 Pemrograman Android

Dalam membuat aplikasi smartphone berbasis Android, kita membutuhkan beberapa paket instalasi yang mendukung untuk kesuksesan pengerjaan aplikasi, antara lain (Safaat, 2011:13) :

1. Instalasi Java (J2SDK)

Android merupakan aplikasi yang dikembangkan dengan berbasis Java, sehingga sebelum kita melakukan *coding* aplikasi berbasis *Android*, komputer/PC kita harus sudah terinstal *program Java*.



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Mas Ardi>java -version
java version "1.7.0"
Java(TM) SE Runtime Environment (build 1.7.0-b147)
Java HotSpot(TM) Client VM (build 21.0-b17, mixed mode, sharing)

C:\Users\Mas Ardi>
```

Gambar 2.2 Java -version

2. Instalasi Android SDK

Android SDK (*Software Development Kit*) merupakan alat bantu dan API dalam mengembangkan aplikasi pada *platform Andorid* menggunakan bahasa pemrograman *Java*. Pada dasarnya, *Android SDK* tidak membutuhkan instalasi cukup dengan melakukan *copy* dan diletakkan di folder atau *drive* manapun langsung dapat dijalankan.

3. Instalasi *Eclipse Indogo*

Eclipse merupakan sebuah IDE yang bersifat *Free* (gratis) dan *Opensource* yang akan digunakan dalam melakukan aplikasi *Android*.



Gambar 2.3 Tampilan Awal *Eclipse Indigo*

4. Instalasi ADT (*Android Development Tools*) / *Plugins Eclipse*

ADT (*Android Development Tools*) merupakan sebuah *Plugins Eclipse* yang menghubungkan *Eclipse Juno* dengan *Android SDK* yang menjadikan *Eclipse Juno* dapat membuat *project* yang berbasis *Android* dan sebagai tempat *coding* aplikasi *Android* nantinya.

5. Membuat AVD (*Android Virtual Device*)

AVD atau yang sering dikenal dengan *Android Virtual Device* merupakan emulator untuk untuk menjalankan program aplikasi *Android* yang kita buat, AVD ini nantinya yang dijadikan sebagai tempat *test* dan menjalankan aplikasi yang akan dibuat. AVD berjalan di *Virtual Machine*.

II.4 UML (*Unified Modelling Language*)

Unified modeling language adalah bahasa standar yang digunakan untuk menjelaskan dan memvisualisasikan artifak dari proses analisa dan desain berorientasi obyek. UML menyediakan standar pada notasi dan diagram yang bisa

digunakan untuk memodelkan suatu sistem. UML dikembangkan oleh 3 pendekar ‘berorientasi obyek’, yaitu Grade Booch, Jim Rumbaugh, dan Ivar Jacobson. UML menjadi bahasa yang bisa digunakan untuk berkomunikasi dalam perspektif obyek antara user dengan developer, antara developer dengan developer, antara developer analisa dengan developer disain, dan antara developer disain dengan developer pemrograman. (Hermawan, 2004:7),

UML menyediakan beberapa notasi dan artifak standar yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisa dan desain. Artifak di dalam UML didefinisikan sebagai informasi dalam berbagai bentuk yang digunakan untuk dihasilkan dalam proses pengembangan perangkat lunak. Contohnya adalah source code yang dihasilkan oleh proses pemrograman.

Komponen UML terdiri dari *View* dan *Diagram*. Beberapa jenis *View* dalam UML antara lain : *use case view*, *logical view*, *component view*, dan *deployment view*. UML mendeskripsikan OOP (Object Oriented Programming) dengan beberapa diagram, diantaranya :

II.4.1 Use Case Diagram

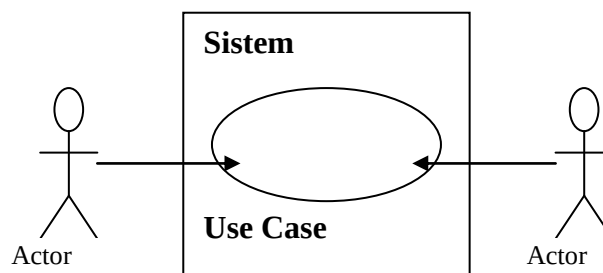
Use case adalah deskripsi fungsi dari sebuah sistem dari *perspektif* pengguna. *Use case* bekerja dengan cara deskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem yang disebut *scenario*. Setiap *scenario* mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras dan urutan waktu. Dengan demikian secara singkat bisa

dikatakan *use case* adalah serangkaian *scenario* yang digabungkan bersama-sama oleh pengguna tujuan umum pengguna.

Dalam pembicaraan tentang *use case*, pengguna biasanya disebut dengan *actor*. *Actor* adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem.

Model *use case* adalah bagian dari model *requirement*. Termasuk disini adalah problem domain object dan penjelasan tentang *user interface*. *Use case* memberikan spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari *perspektif user*.

Notasi *use case* menunjukkan 3 aspek dari sistem yaitu *actor use case* dan *system / sub system boundary*. *Actor* mewakili peran orang, *system* yang lain atau alat ketika berkomunikasi dengan *use case*. Ilustrasi *actor*, *use case* dan *system* ditunjukkan pada gambar 2.4



Gambar 2.4 Contoh Use Case Diagram

Sumber: (Widodo,2011)

Untuk mengidentifikasi *actor*, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. *Actor* adalah *abstraction* dari orang dan sistem yang lain mengaktifkan fungsi dari target

sistem. Orang atau sistem bila muncul dalam beberapa peran. Perlu dicatat bahwa *actor* berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*.

Use case adalah abstraksi dari interaksi antara sistem dan *actor*. Oleh karena itu sangat penting untuk memilih abstraksi yang cocok. *Use case* dibuat berdasarkan keperluan *actor*. *Use case* harus merupakan 'apa' yang dikerjakan *software* aplikasi, bukan 'bagaimana' *software* aplikasinya mengerjakannya. Setiap *use case* harus diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksinya dengan *actor*. Namun *use case* boleh terdiri dari beberapa kata dan tidak boleh ada dua *use case* yang memiliki nama yang sama.

II.4.2 Activity diagram

Activity diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaanya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku *paralel* sedangkan *flowchart* tidak bisa.

II.4.3 Class Diagram

Diagram kelas atau class diagram adalah inti dari proses pemodelan objek baik *forward engineering* maupun *reverse engineering* memanfaatkan diagram ini. *Forward engineering* adalah proses perubahan model menjadi kode program sedangkan *reverse engineering* sebaliknya merubah kode program menjadi model. (Widodo, 2011 : 37)

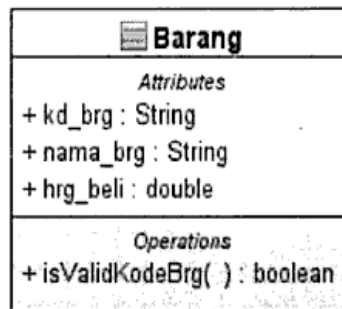
Kelas memiliki apa yang disebut *Atribut* dan metode atau operasi :

- a. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
- b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Susunan kelas suatu sistem yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

- a. Kelas main, kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
- b. Kelas yang menangani tampilan sistem, kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
- c. Kelas yang diambil dari pendefinisian *use case*, kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.
- d. Kelas yang diambil dari pendefinisian data, kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Jenis-jenis kelas diatas juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca *file* teks, dan lain sebagainya sesuai kebutuhan. Adapun contoh class diagram seperti gambar II.4.



Gambar 2.5 Contoh Class Diagram

Sumber: (Widodo,2011)

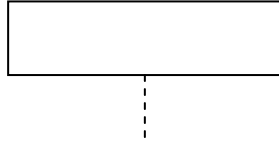
II.4.4 Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sebuah contoh objek dan pesan yang diletakkan diantara objek-objek ini didalam *use case*.

Komponen utama *Sequence diagram* terdiri dari atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*.

a. Objek / participant

Objek diletakkan di dekat bagian atas diagram dengan urutan dari kiri ke kanan. Mereka diatur dalam urutan guna menyederhanakan diagram. Setiap *participant* dihubungkan garis titik-titik yang disebut *lifeline*. Sepanjang *lifeline* ada kotak yang disebut *activation*. *Activation* mewakili sebuah eksekusi operasi dari *participant*. Panjang kotak ini berbanding lurus dengan durasi *activation*. *Activation* mewakili sebuah eksekusi operasi dari *participant*. Panjang kotak ini berbanding lurus dengan durasi *activation*. Bentuk *participant* dapat dilihat pada gambar 2.6



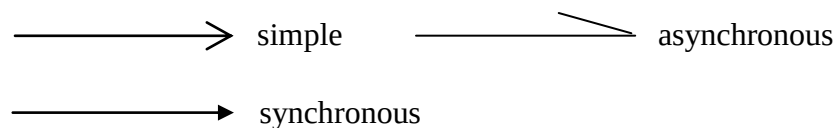
Gambar 2.6 Bentuk Participant

Sumber: (Widodo,2011)

b. *Message*

Sebuah *message* bergerak dari suatu *participant* ke *participant* yang lain dan dari *lifeline* ke *lifeline* yang lain. Sebuah *participant* bisa mengirim sebuah *message* kepada dirinya sendiri.

Sebuah *message* bisa jadi *simple*, *synchronous* atau *asynchoronous*. *Message* yang *simple* adalah sebuah perpindahan (transfer), contoh dari satu *participant* ke *participant* yang lainnya. Jika suatu *participant* mengirimkan sebuah *message* tersebut akan ditunggu sebelum di proses dengan urusannya. Namun jika *message asynchoronous* yang dikirimkan, maka jawabannya atas *message* tersebut tidak perlu ditunggu. Simbol *message* pada *squnence diagram*.



Gambar 2.7 Bentuk Message

Sumber: (Widodo,2011)

c. *Time*

Time adalah diagram yang mewakili waktu pada arah vertikal. Waktu dimulai dari atas ke bawah. *Message* yang lebih dekat dari atas akan dijalankan terlebih dahulu dibanding *message* yang lebih dekat kebawah.

Terdapat dua dimensi pada *sequence diagram* yaitu dimensi dari kiri ke kanan menunjukkan tata letak participant dan dimensi dari atas ke bawah menunjukkan lintasan waktu.