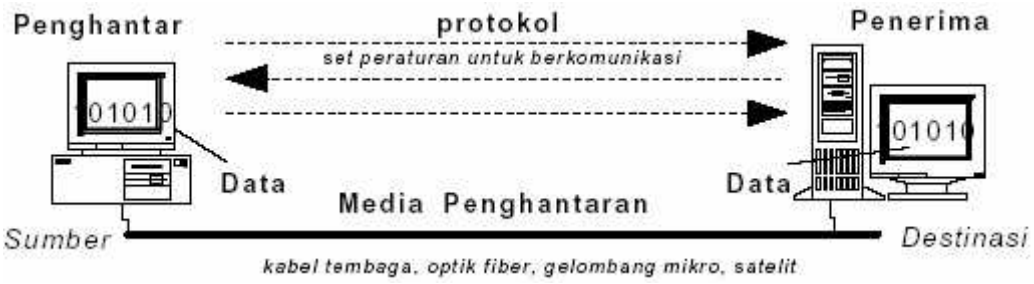
BAB II

LANDASAN TEORI

2. 1. Komunikasi Data

Komunikasi data adalah merupakan bagian dari telekomunikasi yang secara khusus berkenaan dengan transmisi atau pemindahan data dan informasi diantara komputer- komputer dan piranti-piranti yang lain dalam bentuk digital yang dikirimkan melalui media komunikasi data [6]. Data berarti informasi yang disajikan oleh isyarat digital. Komunikasi data merupakan bagian vital dari suatu masyarakat informasi karena sistem ini menyediakan infrastruktur yang memungkinkan komputer-komputer dapat berkomunikasi satu sama lain [Gambar 2.1].

1. Komponen Komunikasi Data
    a. **Pengirim**, adalah piranti yang mengirimkan data
    b. **Penerima**, adalah piranti yang menerima data
    c. **Data**, adalah informasi yang akan dipindahkan
    d. **Media pengiriman**, adalah media atau saluran yang digunakan untuk mengirimkan data
    e. **Protokol**, adalah aturan-aturan yang berfungsi untuk menyelaraskan hubungan



Gambar 2. 1 Komunikasi Data

2. 2. Keamanan Informasi

Keamanan informasi (*information security*) merupakan perlindungan terhadap informasi ketika informasi dikirim dari sebuah sistem ke sistem lainnya [7]. Sistem kemanan informasi memiliki empat tujuan yang sangat mendasar, yaitu:

1. *Confidentiality*

   Menjamin apakah informasi yang dikirim tersebut tidak dapat dibuka atau tidak dapat diketahui oleh orang lain yang tidak berhak. Untuk data yang penting, dibutuhkan sekali tingkat kerahasiaan yang tinggi, yang hanya bisa diakses oleh pihak-pihak tertentu saja (pihak yang berhak).

2. *Integrity*

   Menjamin keutuhan dan keaslian data, sehingga upaya pihak-pihak yang tidak bertanggung jawab untuk melakukan penduplikatan dan perusakan data dapat dihindari.

3. *Availability*

   Menjamin pengguna yang sah agar dapat mengakses informasi dan sumber miliknya sendiri. Tujuannya adalah untuk memastikan bahwa pihak-

pihak yang memang berhak tidak ditolak untuk mengakses informasi yang memang menjadi haknya.

4. *Legitimate Use*

Menjamin kepastian bahwa sumber tidak digunakan atau informasi tidak diakses oleh pihak-pihak yang tidak bertanggung jawab (pihak-pihak yang tidak berhak).

2. 3. Pengantar Multimedia

2. 3. 1. Definisi Multimedia

Multimedia dapat didefinisikan sebagai :

1. Multi [latin nouns] : banyak; bermacam-macam.
2. Medium : sesuatu yang dipakai untuk menyampaikan atau membawa sesuatu [latin]; alat untuk mendistribusikan dan mempresentasikan informasi [*American Heritage Electronic Dictionary*, 1991].

Multimedia dapat diartikan sebagai penggunaan beberapa media yang berbeda untuk menggabungkan dan menyampaikan informasi dalam bentuk text, audio, grafik, animasi, dan video [8].

Beberapa definisi menurut beberapa ahli :

1. Kombinasi dari komputer dan video (*Rosch, 1996*).
2. Kombinasi dari tiga elemen; suara, gambar, teks (*McComick, 1996*).
3. Kombinasi dari paling sedikit dua media input atau output. Media ini dapat berupa audio (suara, musik), animasi, video, teks, grafik dan gambar (*Turban dan kawan-kawan, 2002*)

4. Alat yang dapat menciptakan presentasi yang dinamis dan interaktif yang mengkombinasikan teks, grafik, animasi, audio dan video (*Robin dan Linda, 2001*)
5. Multimedia dalam konteks komputer menurut *Hofstetter* 2001 adalah pemanfaatan komputer untuk membuat dan menggabungkan teks, grafik, audio, video, dengan menggunakan *tool* yang memungkinkan pemakai berinteraksi, berkreasi, dan berkomunikasi.

2. 3. 2. Fungsi Multimedia

Multimedia dapat digunakan dalam :

1. Bidang periklanan yang efektif dan interaktif.
2. Bidang pendidikan dalam penyampaian bahan pengajaran secara interaktif dan dapat mempermudah pembelajaran karena didukung oleh berbagai aspek : suara, video, animasi, teks, dan grafik.
3. Bidang jaringan dan internet yang membantu dalam pembuatan website yang menarik, informatif, dan interaktif.

Multimedia juga mampu :

1. **Mengubah tempat kerja**. Dengan adanya *teleworking*, para pekerja dapat melakukan pekerjaannya tidak harus dari kantor.
2. **Mengubah cara belanja**. *Homeshopping/teleshopping* dapat dilakukan dengan menggunakan internet.
3. **Mengubah cara bisnis**. Banyak perusahaan menggunakan sistem jual beli *online*, bank menggunakan cara *online-banking*.

4. **Mengubah cara memperoleh informasi**. Orang-orang menggunakan internet dan berbagai *software* untuk mencari informasi.
5. **Mengubah cara belajar**. Sekolah mulai menggunakan komputer multimedia, belajar *online*, menggunakan *e-book*.

2. 4. File Multimedia

File komputer adalah bagian tertentu dari data yang ada pada sistem operasi computer [9]. Data ini dapat berisi apa saja dari program dieksekusi untuk pengguna menciptakan dokumen. Sebuah file komputer terdiri dari nama yang menciptakan identitas file dan ekstensi yang memberitahu sistem operasi dan program yang terkait apa jenis file itu. File memiliki awal dan akhir yang ditetapkan, yang berarti mereka berinteraksi dengan sistem komputer dalam cara yang dapat diprediksi.File yang paling umum adalah terdiri dari nama dan ekstensi. Awalnya, nama file terbatas sampai delapan karakter, tetapi sistem modern memungkinkan untuk nama file lebih panjang lagi. Nama ini umumnya ditentukan oleh pengguna atau program yang terkait dan sering memberikan indikasi tentang tujuan file.

Sebuah ekstensi file biasanya diikuti oleh periode 2-4 karakter. Ini bagian dari nama file yang digunakan oleh sistem operasi untuk internal katalogisasi. Cara perpanjangan digunakan bervariasi antara sistem operasi, beberapa memerlukan ekstensi dan beberapa benar-benar mengabaikannya. Kebanyakan sistem jatuh di antara-ekstensi membantu mengidentifikasi program yang digunakan dengan file dan bertindak sebagai jalan pintas ketika membaca.

Data dalam sebuah file komputer yang lengkap dan utuh. Informasi dalam file dapat dibuka dan dibaca, dengan menggunakan perangkat lunak yang tepat. Informasi yang terkandung dalam file mungkin memerlukan informasi lain untuk beroperasi, seperti file eksekusi membutuhkan sebuah file database untuk benar-benar mengeksekusi, tapi eksekusi dan file database adalah dua file yang terpisah.

Karena setiap file komputer berbeda untuk dirinya sendiri, ada karakteristik tertentu yang umum antara semua jenis file. Setiap file memiliki ukuran, bahkan jika ukuran itu adalah nol. Informasi yang disimpan dalam file memerlukan lokasi pada sistem komputer, sehingga setiap file akan diindeks oleh sistem.

Sebuah file komputer dapat dibuat atau dihapus, dipindahkan atau disalin, atau diubah. Seorang pengguna dapat melakukan ini dengan menggunakan program yang terkait dengan file-file atau melalui manajemen file dari sistem operasi itu sendiri. Operasi ini mungkin sulit dengan beberapa file karena interaksi mereka dengan sistem operasi, tetapi masih mungkin dilakukan.

2. 4. 1. File Teks

File teks memiliki beberapa jenis, yaitu :

1. *Plain Text (Unformatted Text).*
    a. Teks adalah data dalam bentuk karakter.
    b. Teks dalam hal ini adalah kode ASCII (*American Standard Code for Information Interchange*) dan ASCII *extension* seperti UNICODE murni.
    c. Contoh *plain text* adalah pada saat kita mengetik dengan menggunakan notepad (.txt).
    d. *Plain text* berjenis MIME *text/plain*.

e. Teks file tidak terenkripsi, tidak mengandung *embedded information*, seperti informasi font, tidak mengandung link, dan *inline-image*.

2. *Formatted Text (Rich Text Format).*

   a. Serangkaian karakter format yang telah didefinisikan. Contoh *rich text* adalah pada saat kita mengetik dengan menggunakan Wordpad (.rtf), Microsoft Word (.doc, .docx)

   b. *Rich Text Format* merupakan plain teks yang telah diformat sedemikian rupa dengan menggunakan aturan (tag/tanda) tertentu sehingga teks tersebut dapat ditebalkan (*bold*), miring (*italics*), digaris-bawahi (*underline*), diwarna, diganti font, dan lain-lain.

3. *Hypertext*

   Hypertext adalah teks yang memiliki fasilitas *linking*, contohnya :

   a. HTML (*Hypertext Markup Language*) merupakan standard bahasa yang digunakan untuk menampilkan dokumen web. Yang bisa kita lakukan dengan HTML yaitu :

      i. Mengontrol tampilan dari web page dan kontennya.

      ii. Mempublikasikan dokumen secara online sehingga bisa diakses dari seluruh dunia.

      iii. Membuat online form yang bisa digunakan untuk menangani pendaftaran, transaksi secara online.

      iv. Menambahkan obyek-obyek seperti gambar, audio, video dan juga java applet dalam dokumen HTML.

v. Mendukung *link* (sebuah hubungan dari satu dokumen ke dokumen yang lain) antar dokumen. *Link* pada umumnya berwarna biru, dan jika sudah pernah diklik berwarna ungu.

b. XML (*eXtensible Markup Language*) memiliki beberapa keuntungan dan keunggulan, yaitu :

i. *Simple*, karena XML tidak serumit HTML, strukturnya jelas dan sederhana.

ii. *Intelligence*, karena XML mampu menangani berbagai kompleksitas markup bertingkat-tingkat.

iii. *Portable*, karena memisahkan data dan presentasi.

iv. *Fast*, pencarian data cepat.

v. *Extensible*, dapat ditukar/digabung dengan dokumen XML lain.

vi. *Linking*, XML dapat melakukan *linking* yang lebih baik daripada HTML, bahkan dapat me*link* satu atau lebih poin dari dalam maupun luar data.

vii. *Maintenance*, XML mudah untuk diatur dan dipelihara, karena hanya berupa data, *stylesheet* dan *link* terpisah dari XML.

2. 4. 2. File Gambar

Gambar (*image*) merupakan suatu representasi spatial dari suatu obyek, dalam pandangan 2D atau 3D. Gambar digital merupakan suatu fungsi dengan nilai-nilai yang berupa intensitas cahaya pada tiap-tiap titik pada bidang yang telah diquantisasikan (diambil sampelnya pada interval diskrit) [10].

Titik dimana suatu gambar di-*sampling* disebut *picture element* (pixel). Nilai intensitas warna pada suatu pixel disebut *gray scale* level.

1 bit → binary-valued image (0 -1)

8 bits → gray level (0 – 255)

16 bits → high color (216)

24 bits → 224 true color

32 bits → true color (232)

Format gambar digital memiliki 2 parameter :

a. Spatial Resolution → pixels x pixels

b. Color Encoding → bits / pixel

Misal : terdapat gambar berukuran 100 pixels x 100 pixels dengan *color encoding* 24 bits dengan R=8bits, G=8bits, B=8bits per pixel, maka *color encoding* akan mampu mewakili 0 .. 16.777.215 (mewakili 16 juta warna) dan ruang disk yang dibutuhkan = 100 * 100 * 3 byte (karena RGB) = 30.000 bytes = 30 Kb atau 100 * 100 * 24 bits = 240.000 bits.

Tabel 2. 1 Tabel Resolusi display dan kebutuhan memory

Standard	Resolusi	Warna	Kebutuhan memory / frame (bytes)
VGA	640 x 480	8 bit	307.2 Kb
XGA	640 x 480 1024 x 768	16 bit 8 bit	614.4 Kb 786.432 Kb
SVGA	800 x 600 1024 x 768 1024 x 768	16 bit 8 bit 24 bit	960 Kb 786.432 Kb 2359.296 Kb

Jika suatu gambar disimpan maka yang disimpan adalah array 2D dimana masing-masing merepresentasikan data yang berhubungan dengan pixel tersebut.

Array[x,y] = warna pixel. Setiap pixel dapat mempunyai informasi tambahan yang berhubungan dengan pixel tersebut. Masing-masing gambar juga memiliki informasi tambahan seperti lebar X panjang gambar, kedalaman gambar, pembuat dan lain-lain.

File gambar memiliki beberapa format, yaitu :

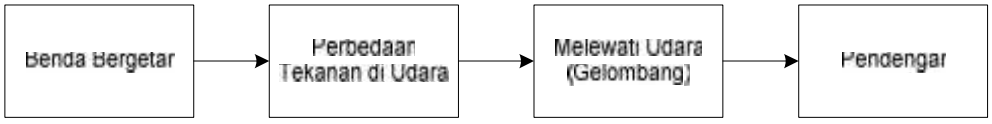
1. Bitmap (.bmp)
    a. Format gambar yang paling umum dan merupakan format standar windows.
    b. Ukuran filenya sangat besar karena bisa mencapai ukuran Megabytes.
    c. File ini merupakan format yang belum terkompresi dan menggunakan sistem warna RGB (Red, Green, Blue) dimana masing-masing warna pixelnya terdiri dari 3 komponen R, G, dan B yang dicampur jadi satu.
    d. File BMP tidak bisa (sangat jarang) digunakan di web (internet) karena ukurannya sangat besar.
2. Joint Photographic Expert Group (.jpeg/.jpg)
    a. Format JPG merupakan format yang paling terkenal sekarang ini. Hal ini karena sifatnya yang berukuran kecil (hanya puluhan/ratusan Kb saja) dan bersifat portable.
    b. File ini sering digunakan pada bidang fotografi untuk menyimpan file foto.
    c. File ini bisa digunakan di web (internet).

3. Graphics Interchange Format (.gif)
    a. Format GIF ini berukuran kecil dan mendukung gambar yang terdiri dari banyak frame sehingga bisa disebut dengan gambar animasi (gambar gerak).
    b. Format ini sering digunakan di internet untuk menampilkan gambar-gambar di web.
4. Potable Network Graphics (.png)
    a. Format yang standar dan sering digunakan di internet untuk menampilkan gambar atau pengiriman gambar. Ukuran file ini cukup kecil dan setara dengan ukuran gif dengan kualitas yang bagus. Namun tidak mendukung animasi (gambar bergerak).

Masih banyak lagi beberapa format file gambar yang ada pada saat ini, seperti TIFF (Tagged Image File Format), ICO (Icon), EMF (Enchanced Windows Metafile), PCX, ANI (Animation), Cur (Cursor), WBMP (WAP BMP), PSD (Adobe Photoshop Document), dan CDR (Corel Draw).

2. 4. 3. File Suara

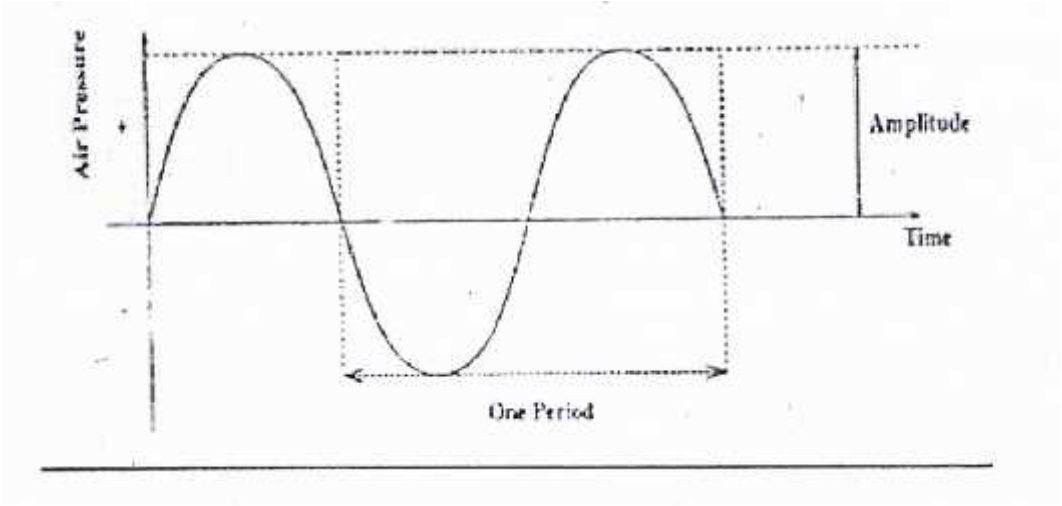
Suara adalah fenomena fisik yang dihasilkan oleh getaran benda atau getaran suatu benda yang berupa sinyal analog dengan amplitudo yang berubah secara kontinyu terhadap waktu [10].



Gambar 2. 2 Proses Suara

Suara berhubungan erat dengan rasa “mendengar”. Suara/bunyi biasanya merambat melalui udara [Gambar 2.2]. Suara/bunyi tidak bisa merambat melalui ruang hampa.

Konsep dasar pada suara adalah suara dihasilkan oleh getaran suatu benda. Selama bergetar, perbedaan tekanan terjadi di udara sekitarnya. Pola osilasi yang terjadi dinamakan sebagai “Gelombang”. Gelombang mempunyai pola sama yang berulang pada interval tertentu, yang disebut sebagai “Periode” [Gambar 2.3]. Contoh suara periodik adalah instrument musik, nyanyian burung, dan lain-lain. Sedangkan contoh suara non-periodik adalah batuk, percikan ombak, dan lain-lain.



Gambar 2. 3 Gelombang suara

Suara berkaitan erat dengan :

1. **Frekuensi**
   a. Banyaknya periode dalam 1 detik
   b. Satuan : *Hertz (Hz)* atau *cycles per second (cps)*
   c. Panjang gelombang suara (*wavelength*) dirumuskan = c/f

      Dimana c = kecepatan rambat bunyi

Dimana f = frekuensi

Contoh :

Berapakah panjang gelombang untuk gelombang suara yang memiliki kecepatan rambat 343 m/s dan frekuensi 20 KHz?

Jawab :

Wavelength = c/f = 343/20 = 17,15 mm

d. Berdasarkan frekuensi, suara dibagi menjadi :

i. Infrasound 0Hz – 20Hz
ii. Pendengaran manusia 20Hz – 20KHz
iii. Ultrasound 20KHz – 1GHz
iv. Hypersound 1GHz – 10THz

Manusia membuat suara dengan frekuensi : 50Hz – 10KHz. Sinyal suara musik memiliki frekuensi : 20Hz – 20KHz. Sistem multimedia menggunakan suara yang berada pada *range* pendengaran manusia.

Suara yang berada pada *range* pendengaran manusia sebagai “AUDIO” dan gelombangnya sebagai “ACCOUSTIC SIGNALS”. Suara diluar *range* pendengaran manusia dapat dikatakan sebagai “NOISE” (getaran yang tidak teratur dan tidak berurutan dalam berbagai frekuensi, tidak dapat didengarkan manusia).

2. Amplitudo

a. Keras lemahnya bunyi atau tinggi rendahnya gelombang.
b. Satuan amplitudo adalah decibel (db).

c. Bunyi mulai dapat merusak telinga jika tingkat volumenya lebih besar dari 85 dB dan pada ukuran 130 dB akan mampu membuat hancur gendang telinga.

3. Velocity

a. Kecepatan perambatan gelombang bunyi sampai ke telinga pendengar.
b. Satuan yang digunakan adalah m/s.
c. Pada udara kering dengan suhu 20°C (68°F)m kecepatan rambat suara sekitar 343 m/s.

Gelombang suara analog tidak dapat langsung direpresentasikan pada komputer. Komputer mengukur amplitudo pada satuan waktu tertentu untuk menghasilkan sejumlah angka. Tiap satuan pengukuran ini dinamakan "SAMPLE".

Analog to Digital Conversion (ADC) adalah proses mengubah amplitudo gelombang bunyi ke dalam waktu interval tertentu (disebut juga *sampling*), sehingga menghasilkan representasi digital dari suara.

Sampling rate adalah beberapa gelombang yang diambil dalam satu detik. Contohnya adalah jika kualitas CD Audio dikatakan memiliki frekuensi sebesar 44100 Hz, berarti jumlah sample sebesar 44100 per detik.

Nyquist Sampling rate adalah untuk memperoleh representasi akurat dari suatu sinyal analog secara *lossless*, amplitudonya harus diambil sample-nya setidaknya pada kecepatan (*rate*) sama atau lebih besar dari 2 kali lipat komponen frekuensi maksimum yang akan didengar. Misalnya, untuk sinyal analog dengan *bandwith* 15Hz – 10KHz → *sampling rate* = 2 x 10KHz = 20KHz.

Digital to Analog Converter (DAC) adalah proses mengubah digital audio menjadi sinyal analog. DAC biasanya hanya menerima sinyal digital *Pulse Code Modulation* (PCM). PCM adalah representasi digital dari sinyal analog, dimana gelombang disample secara beraturan berdasarkan interval waktu tertentu, yang kemudian akan diubah ke biner. Proses pengubahan ke biner disebut Quantisasi. Contoh DAC adalah soundcard, CDPlayer, Ipod, mp3player.

File suara memiliki beberapa format sebagai berikut :

1. **AAC (*Advanced Audio Coding*) [.m4a]**
   a. AAC bersifat *lossy compression* (data hasil kompresi tidak bisa dikembalikan lagi ke data sebelum dikompres secara sempurna, karena setelah dikompres terdapat data-data yang hilang).
   b. AAC merupakan *audio codec* yang menyempurnakan mp3 dalam hal medium dan high bit rates.

   Cara kerja :

   a. Bagian-bagian sinyal yang tidak relevan dibuang.
   b. Menghilangkan bagian-bangian sinyal yang redundan.
   c. Dilakukan proses MDCT (*Modified Discret Cosine Transform*) berdasarkan tingkat kekompleksitasan sinyal.
   d. Adanya penambahan *Internal Error Correction*.
   e. Kemudian, sinyal disimpan atau dipancarkan.

   Kelebihan AAC dari MP3

   a. Sample ratenya antara 8Hz – 96KHz, sedangkan MP3 16Hz – 48KHz.
   b. Memiliki 48 channel.

c. Suara lebih bagus untuk kualitas bit yang rendah (dibawah 16Hz).

2. WAVEFORM AUDIO [.wav]

a. WAV adalah format audio standar Microsoft dan IBM untuk PC.

b. WAV biasanya menggunakan coding PCM (*Pulse Code Modulation*).

c. WAV adalah data tidak terkompres sehingga seluruh sample Audio disimpan semuanya di harddisk.

d. Software yang dapat menciptakan WAV dari Analog Sound misalnya adalah Windows Sound Recorder.

e. WAV jarang sekali digunakan di internet karena ukurannya yang relatif besar.

f. Maksimal ukuran file WAV adalah 2Gb.

3. Audio Interchange File Format [.aif]

a. Merupakan format standar Macintosh.

b. Software pendukung : Apple QuickTime.

4. Audio CD [.cda]

a. Format untuk mendengarkan CD Audio.

b. CD Audio stereo berkualitas sama dengan PCM/WAV yang memiliki sampling rate 44100 Hz, 2 channel (stereo) pada 16 bit.

c. Durasi = 75 menit dan dynamic range = 95 dB.

5. Mpeg Audio Layer 3 [.mp3]

a. Merupakan file dengan *lossy compession*.

b. Sering digunakan di Internet karena ukurannya yang cukup kecil dibandingkan ukuran audio file yang tidak terkompresi.

c. Distandarisasi pada tahun 1991.

d. Kompresi dilakukan dengan menghilangkan bagian-bagian bunyi yang kurang berguna bagi pendengaran manusia.

e. Kompresi mp3 dengan kualitas 128 bits 44000 Hz biasanya akan menghasilkan file berukuran 3-4 Mb, tetapi unsur panjang pendeknya lagu juga akan berpengaruh.

6. MIDI (Music Instrument Digital Interface) [.midi]

Standard yang dibuat oleh perusahaan alat-alat musik elektronik berupa serangkaian spesifikasi agar berbagai instrumen dapat berkomunikasi.

MIDI = format data digital

Interface MIDI terdiri dari 2 komponen :

a. Perangkat keras

Hardware yang terhubung ke peralatan (alat instrumen / komputer)

b. Data Format

Pengkodean informasi

- Spesifikasi instrumen
- Awal / akhir nada
- Frekuensi
- Volume suara

2. 4. 4. File Video

Video adalah teknologi untuk menangkap, merekam, memproses, mentransmisikan dan menata ulang gambar bergerak. Biasanya menggunakan film seluloid, sinyal elektronik, atau media digital. Video sangat erat berkaitan dengan “penglihatan dan pendengaran”.

Digital Video adalah jenis sistem video recording yang bekerja menggunakan sistem digital dibandingkan dengan analog dalam hal representasi videonya. Biasanya digital video direkam dalam tape, kemudian didistribusikan melalui optical disc, misalnya VCD dan DVD. Salah satu alat yang dapat digunnakan untuk menghasilkan video digital adalah *camcorder*, yang digunakan untuk merekam gambar-gambar video dan audio, sehingga sebuah *camcorder* akan terdiri dari *camera* dan *recorder*.

Video kamera menggunakan 2 teknik, yaitu :

1. Interlanced
    a. Adalah metode untuk menampilkan image/gambar dalam *rasterscanned display service* seperti CRT televisi analog, yang ditampilkan bergantian antara garis ganjil dan genap secara cepat untuk setiap frame.
    b. *Refresh rate* yang disarankan untuk metode interlanced adalah antara 50 – 80 Hz
    c. Interlance digunakan di sistem televisi analog :
        i. PAL (50 *fields per second*, 625 *lines*, *even field drawn first*)
        ii. SECAM (50 *fields per second*, 625 *lines*)
        iii. NTSC (59,94 *fields per second*, 525 *lines*, *even field drawn first*)

2. Progressive scan

   a. Adalah metode untuk menampilkan, menyimpan, dan memancarkan gambar dimana setiap baris untuk setiap *frame* digambarkan secara berurutan.

   b. Biasa digunakan pada CRT monitor komputer.

File video memiliki beberapa format sebagai berikut :

A. MP4 / MPEG-4

a. Film atau video klip yang menggunakan kompresi MPEG-4, standar yang dikembangkan oleh *Moving Picture Experts Group* (MPEG), biasanya digunakan untuk berbagi file video di Internet.

b. Format video MPEG-4 menggunakan kompresi terpisah untuk track audio dan video.

c. Video dikompresi dengan MPEG-4 video encoding.

d. Audio dikompresi menggunakan kompresi AAC, jenis yang sama kompresi audio yang digunakan dalam file AAC.

B. AVI (*Audio Video Interleave*)

a. Menyimpan data video yang dapat dikodekan dalam berbagai codec, biasanya menggunakan kompresi kurang dari format yang sama seperti MPEG dan MOV.

C. FLV (*Flash Video*)

a. Flash-file video yang kompatibel diekspor oleh Eksportir Video Flash plug-in (termasuk dengan Adobe Flash) atau program lain dengan dukungan file FLV.

b. Terdiri dari header pendek, audio interleaved, video, dan metadata paket.
c. Audio dan video data disimpan dalam format yang sama yang digunakan oleh standar (. SWF) file Flash.

D. MKV (*Matroska Video*)

a. Format wadah video yang sama dengan format AVI, ASF, MOV .
b. Mendukung beberapa jenis codec audio dan video, juga termasuk SRT, SSA, atau USF (*Universal Subtitle Format*).
c. Format video Matroska menjadi populer di masyarakat penggemar subbing anime (di mana pengguna membuat sub judul) karena mendukung multiple audio track dan subtitle dalam file yang sama.

E. 3GP

a. Format audio dan video kontainer yang dikembangkan oleh *3rd Generation Partnership Project* (3GPP).
b. Dirancang sebagai format multimedia untuk transmisi file audio dan video antara ponsel 3G dan melalui internet.
c. Umum digunakan oleh ponsel yang mendukung video capture.

2. 5. Pengenalan Kriptografi

2. 5. 1. Definisi dan Sejarah

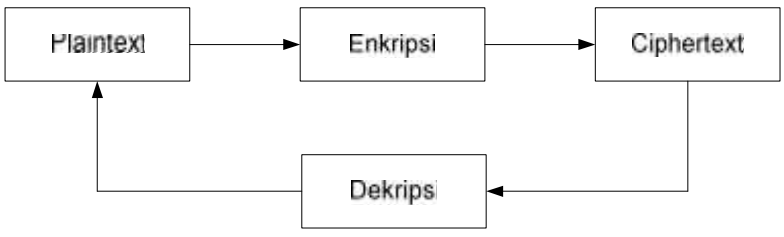
Kriptografi adalah ilmu untuk mempelajari penulisan secara rahasia dengan tujuan bahwa komunikasi dan data dapat dikodekan (*encode* / *encrypt*) dan didekodekan (*decode* / *decrypt*) kembali untuk mencegah pihak-pihak lain yang ingin mengetahui isinya, dengan menggunakan kode-kode dan aturan-aturan

tertentu dan metode lainnya sehingga hanya pihak yang berhak saja yang dapat mengetahui isi pesan sebenarnya [1].

Sebelum tahun 1970-an, teknologi kriptografi digunakan terbatas hanya untuk tujuan militer dan diplomatik. Akan tetapi kemudian bidang bisnis dan perorangan mulai menyadari pentingnya melindungi informasi berharga.

Kriptografi berasal dari bahasa Yunani yaitu kripto yang artinya tersembunyi dan grafia yang artinya sesuatu yang tertulis. Jadi kriptografi merupakan proses transformasi suatu data yang memiliki makna ke dalam bentuk data acak yang tidak memiliki makna, gunanya untuk menjaga kerahasiaan data dengan menerapkan rumus matematika tertentu, kemudian dengan transformasi balik, data acak ini dapat diubah kembali ke dalam bentuk data semula.

Sumber data yang akan ditransformasikan / dikodekan disebut *plaintext*. Proses untuk menyembunyikan isi sesungguhnya dari sebuah *plaintext* ke dalam bentuk yang hampir tidak dapat dibaca tanpa pengetahuan yang cukup dan sesuai disebut enkripsi. Sebuah sumber data yang telah ditransformasikan disebut *ciphertext*. Proses untuk mengembalikan *ciphertext* ke *plaintext* disebut dengan dekripsi. Proses enkripsi dan dekripsi dapat dilihat pada gambar 2.4 berikut ini.



Gambar 2. 4 Proses enkripsi dan dekripsi

Kriptografi mempunyai sebuah sejarah yang panjang dan menarik. Penghargaan untuk referensi non teknis terlengkap adalah *The Codebreakers* oleh

Kahn. Buku ini meliputi kriptografi dari penggunaannya yang pertama dan terbatas oleh bangsa Mesir 4000 tahun yang lalu, hingga abad XX dimana kriptografi memegang peranan penting sebagai akibat dari kedua perang dunia. Diselesaikan pada tahun 1963, buku Kahn mencakup aspek-aspek sejarah yang paling penting (pada waktu itu) pada perkembangan kriptografi. Para pengguna utama adalah mereka yang berhubungan dengan militer, pelayanan diplomatik dan pemerintahan secara umum. Kriptografi digunakan sebagai suatu alat untuk melindungi rahasia dan strategi-strategi nasional.

Perkembangan komputer dan sistem komunikasi pada tahun 1960-an mengakibatkan kebutuhan pihak swasta akan alat untuk melindungi informasi dalam bentuk digital dan untuk menyediakan layanan keamanan. Dimulai dengan usaha *Feistel* di IBM pada awal tahun 1970-an dan mencapai puncaknya pada tahun 1977 dengan pengadopsian sebagai Standar Pemrosesan Informasi pemerintah Amerika Serikat bagi enkripsi informasi rahasia, *DES*, *Data Encryption Standard*, adalah mekanisme kriptografi yang paling terkenal dalam sejarah. Metode ini menjadi alat standar untuk mengamankan *electronic commerce* bagi banyak institusi financial di seluruh dunia. Sejarah perkembangan kriptografi dapat dilihat pada tabel 2.2.

Tabel 2. 2 Sejarah Perkembangan Kriptografi

Tahun	Keterangan
~ 1900 A.D.	Pertama kali digunakannya teknik transformasi *cyptography* di "*tomb inscription*", merupakan penggunaan kriptografi yang pertama kali diketahui.
~ 475 A.D.	Sparta menggunakan kriptografi untuk komunikasi dan juga merancang alat untuk mengenkripsi (skytale) yang menghasilkan *transposition cipher*.
~ 350 A.D.	"Aenas The Tactician" mengeluarkan tulisan pertama mengenai keamanan komunikasi dan kriptografi.
~ 60 A.D.	Julius Caesar menjadi orang yang pertama kali yang diketahui menggunakan substitution *cipher*.
1412	"Treatise" tertua yang diketahui dalam kriptanalis yang diter*bit*kan oleh Alkalkas Handi (Mesir).
1917	Edward Hugh Hibern mengembangkan mesin motor yang pertama.
1971	IBM mengembangkan teknik enkripsi Lucifer.
1975	DES diumumkan (disetujui tahun 1977).
1976	Presentasi terbuka pertama tentang konsep public key oleh Diffie dan Helman.
1977	Merkle mengembangkan algoritma knapsack dan memberikan hadiah $100 bagi yang dapat memecahkan kuncinya (algoritma dengan satu kali pengulangan). Algoritma Rivest – Shamir – Aldeman (RSA) diumumkan kepada umum.

2. 5. 2. Aplikasi Kriptografi

2. 5. 2. 1. Confidentality dan Privacy

Confidentiality dan *privacy* terkait dengan kerahasiaan data atau informasi. Pada sistem *e-government* kerahasiaan data-data pribadi (*privacy*) sangat penting. Hal ini kurang mendapat perhatian di sistem *e-government* yang sudah ada. Bayangkan jika data pribadi anda, misalnya data KTP atau kartu keluarga, dapat diakses secara *online*. Maka setiap orang dapat melihat tempat dan tanggal lahir anda, alamat anda, dan data lainnya. Data ini dapat digunakan untuk melakukan penipuan dan pembobolan dengan mengaku-aku sebagai anda (atau keluarga anda).

Ancaman atau serangan terhadap kerahasiaan data ini dapat dilakukan dengan menggunakan penerobosan akses, penyadapan data (*sniffer, key logger*), *social engineering* (yaitu dengan menipu), dan melalui kebijakan yang tidak jelas (tidak ada).

Untuk itu kerahasiaan data ini perlu mendapat perhatian yang besar dalam implementasi sistem *e-government* di Indonesia. Proteksi terhadap data ini dapat dilakukan dengan menggunakan *firewall* (untuk membatasi akses), segmentasi jaringan (juga untuk membatasi akses), enkripsi (untuk menyandikan data sehingga tidak mudah disadap), serta kebijakan yang jelas mengenai kerahasiaan data tersebut.

Pengujian terhadap kerahasian data ini biasanya dilakukan secara berkala dengan berbagai metode. Ada beberapa jenis informasi yang tersedia didalam sebuah jaringan komputer. Setiap data yang berbeda pasti mempunyai grup pengguna yang berbeda pula dan data dapat dikelompokkan sehingga beberapa pembatasan kepada pengunaan data harus ditentukan. Pada umumnya data yang terdapat didalam suatu perusahaan bersifat rahasia dan tidak boleh diketahui oleh pihak ketiga yang bertujuan untuk menjaga rahasia perusahaan dan strategi perusahaan. *Backdoor*, sebagai contoh, melanggar kebijakan perusahaan dikarenakan menyediakan akses yang tidak diinginkan kedalam jaringan komputer perusahaan.

Kerahasiaan dapat ditingkatkan dan didalam beberapa kasus pengenkripsian data. Kontrol akses adalah cara yang lazim digunakan untuk membatasi akses kedalam sebuah jaringan komputer. Sebuah cara yang mudah tetapi mampu untuk

membatasi akses adalah dengan menggunakan kombinasi dari *username*-dan-*password* untuk proses otentikasi pengguna dan memberikan akses kepada pengguna (*user*) yang telah dikenali. Didalam beberapa lingkungan kerja keamanan jaringan komputer, ini dibahas dan dipisahkan dalam konteks otentikasi.

Dalam menjaga kerahasiaan data, kriptografi mentransformasikan data jelas (*plaintext*) ke dalam bentuk data sandi (*ciphertext*) yang tidak dapat dikenali. *Ciphertext* inilah yang kemudian dikirimkan oleh pengirim (*sender*) kepada penerima (*receiver*). Setelah sampai di penerima, *ciphertext* tersebut ditranformasikan kembali ke dalam bentuk *plaintext* agar dapat dikenali.

2. 5. 2. 2. Otentikasi (Authentication)

Otentikasi merupakan identifikasi yang dilakukan oleh masing – masing pihak yang saling berkomunikasi, maksudnya beberapa pihak yang berkomunikasi harus mengidentifikasi satu sama lainnya. Informasi yang didapat oleh suatu pihak dari pihak lain harus diidentifikasi untuk memastikan keaslian dari informasi yang diterima. Identifikasi terhadap suatu informasi dapat berupa tanggal pembuatan informasi, isi informasi, waktu kirim dan hal-hal lainnya yang berhubungan dengan informasi tersebut. Aspek ini berhubungan dengan metode untuk menyatakan bahwa informasi betul-betul asli, orang yang mengakses atau memberikan informasi adalah betul-betul orang yang dimaksud, atau *server* yang kita hubungi adalah betul-betul *server* yang asli.

Masalah pertama, membuktikan keaslian dokumen, dapat dilakukan dengan teknologi *watermarking* dan *digital signature*. *Watermarking* juga dapat

digunakan untuk menjaga "*intelectual property*", yaitu dengan menandai dokumen atau hasil karya dengan "tanda tangan" pembuat. Masalah kedua biasanya berhubungan dengan akses kontrol, yaitu berkaitan dengan pembatasan orang yang dapat mengakses informasi.

2. 5. 2. 3. Integritas (*Integrity*)

Aspek integritas (*integrity*) terkait dengan keutuhan data. Aspek ini menjamin bahwa data tidak boleh diubah tanpa ijin dari pihak yang memiliki hak. Acaman terhadap aspek integritas dilakukan dengan melalui penerobosan akses, pemalsuan (*spoofing*), virus yang mengubah atau menghapus data, dan *man in the middle attack* (yaitu penyerangan dengan memasukkan diri di tengah-tengah pengiriman data). Proteksi terhadap serangan ini dapat dilakukan dengan menggunakan *digital signature*, *digital certificate*, *message authentication code*, *hash function*, dan *checksum*. Pada prinsipnya mekanisme proteksi tersebut membuat kode sehingga perubahan satu bit pun akan mengubah kode.

2. 5. 2. 4. Nonrepudation

Aspek ini menjaga agar seseorang tidak dapat menyangkal telah melakukan sebuah transaksi. Sebagai contoh, seseorang yang mengirimkan email untuk memesan barang tidak dapat menyangkal bahwa dia telah mengirimkan email tersebut. Aspek ini sangat penting dalam hal *electronic commerce*. Penggunaan *digital signature*, *certificates*, dan teknologi kriptografi secara umum dapat menjaga aspek ini. Akan tetapi hal ini masih harus didukung oleh hukum sehingga status dari *digital signature* itu jelas legal.

Definisi dari *nonrepudition* adalah sebagai berikut :

1. Kemampuan untuk mencegah seorang pengirim untuk menyangkal kemudian bahwa dia telah mengirim pesan atau melakukan sebuah tindakan.
2. Proteksi dari penyangkalan oleh satu satu dari entitas yang terlibat didalam sebuah komunikasi yang turut serta secara keseluruhan atau sebagian dari komunikasi yang terjadi.

Proses tranformasi dari *plaintext* menjadi *ciphertext* disebut proses *encipherment* atau enkripsi (*encryption*), sedangkan proses mentransformasikan kembali *ciphertext* menjadi *plaintext* disebut proses dekripsi (*decryption*).

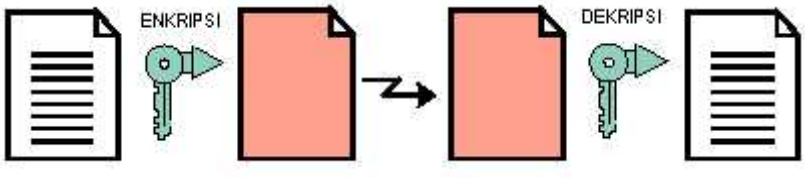
Untuk mengenkripsi dan mendekripsi data, kriptografi menggunakan suatu algoritma (*cipher*) dan kunci (key). *Cipher* adalah fungsi matematika yang digunakan untuk mengenkripsi dan mendekripsi. Sedangkan kunci merupakan sederetan bit yang diperlukan untuk mengenkripsi dan mendekripsi data.

2. 5. 3. Jenis-jenis Kriptografi

2. 5. 3. 1. Kunci Simetris

Ini adalah jenis kriptografi yang paling umum dipergunakan. Kunci untuk mengenkripsi pesan sama dengan untuk mendekripsi pesan yang telah terenkripsi. Jadi pembuat pesan dan penerimanya harus memiliki kunci yang sama persis. Siapapun yang memiliki kunci yang memiliki tersebut termasuk pihak-pihak yang tidak diinginkan dapat membuat dan membongkar rahasia *ciphertext*. Masalah yang paling jelas disini terkadang bukanlah masalah pengiriman *ciphertext*-nya, melainkan masalah bagaimana menyampaikan kunci simetris tersebut kepada

pihak yang diinginkan. Contoh algoritma kunci simetris yang terkenal adalah DES (*Data Encryption Standard*) dan RC-4 ada pada gambar 2.13.



Gambar 2. 5 Penggunaan Kunci simetris

Kelebihan kriptografi kunci simetri (kunci rahasia) ini adalah memiliki kecepatan proses enkripsi dan dekripsi yang jauh lebih cepat dibandingkan dengan sistem kriptografi lainnya. Kriptografi kunci simetri juga memiliki kelemahan. Adapun kelemahan dari kriptografi kunci simetri ialah :

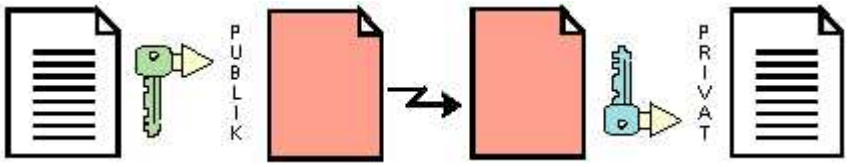
1. Masalah distribusi kunci. Bagaimana caranya agar kunci pembuka (untuk mendekripsi) tersebut bisa dengan aman sampai ke penerima. Padahal diketahui bahwa jaringan *internet* benar-benar tidak aman. Jika kunci sampai hilang atau dapat ditebak oleh pihak ketiga yang tidak bertanggungjawab, maka kriptosistem ini sudah tidak aman lagi.
2. Masalah efisiensi jumlah kunci. Jika ada pengguna sebanyak n, maka akan dibutuhkan n(n-1)/2 kunci, sehingga untuk jumlah pengguna yang sangat banyak, sistem ini tidak efisien.

2. 5. 3. 2. Kunci Asimetris

Pada pertengahan tahun 70-an Whitfield Diffie dan Martin Hellman menemukan teknik enkripsi asimetris yang merevolusi dunia kriptografi. Kunci asimetris adalah pasangan kunci-kunci kriptografi yang salah satunya dipergunakan untuk proses enkripsi dan yang satu lagi untuk dekripsi. Semua

orang yang mendapat kunci publik dapat menggunakan untuk mengenkripsi suatu pesan, sedangkan hanya satu orang saja yang memiliki rahasia tertentu dalam hal ini kunci privat untuk melakukan pembongkaran terhadap sandi yang dikirim untuknya.

Dengan cara seperti ini, jika Anto mengirim pesan untuk Badu, anto dapat merasa yakin bahwa pesan tersebut hanya dapat dibaca oleh Badu, karena hanya Badu yang bisa melakukan deskripsi dengan kunvi privatnya. Tentunya Anto harus memilki kunci publik Badu untuk melakukan enkripsi. Anto bisa mendapatkannya dari Badu, ataupun dari pihak ketiga seperti Tari. Gambaran penggunaan kunci asimetris dapat dilihat pada gambar 2.14.



Gambar 2. 6 Penggunaan Kunci Asimetris

Teknik enkripsi asimetris ini jauh lebih lambat ketimbang enkripsi dengan kunci simetris. Oleh karena itu, biasanya bukanlah pesan itu sendiri yang disandikan dengan kunci asimetris, namun hanya kunci simetrislah yang disandikan dengan kunci asimetris. Sedangkan pesannya dikirm setelah disandikan dengan kunci simetris tadi. Contoh algortima terkenal yang menggunakan kunci asimetris adalah RSA (merupakan singkatan penemunya yakni Rivest Shamir dan Adleman).

2. 6. Pengenalan Steganografi

Steganografi (*steganography*) adalah ilmu dan seni menyembunyikan pesan rahasia di dalam pesan lain sehingga keberadaan pesan rahasia tersebut tidak dapat diketahui [1]. Misalnya pengirim menyembunyikan pesan rahasia di dalam tulisan berikut :

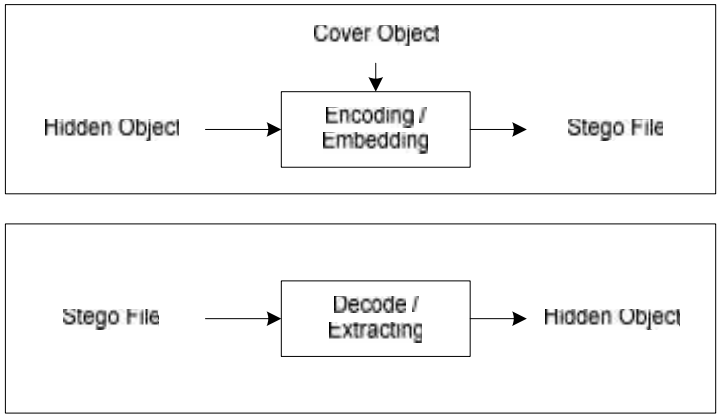
Gerakan **o**rang-**o**rang **d**ari **y**ordania **e**nggan **a**mbil **r**esiko

Yang dalam hal ini huruf awal setiap kata bila dirangkai akan membentuk pesan rahasia : **Good year**

Steganografi berasal dari Bahasa Yunani, yaitu "*steganos*" yang artinya "tulisan tersembunyi (*covered writing*)". Steganografi sangat kontras dengan kriptografi. Jika kriptografi merahasiakan makna pesan sementara eksistensi pesan tetap ada, maka steganografi menutupi keberadaan pesan. Steganografi dapat dipandang sebagai kelanjutan kriptografi dan dalam prakteknya pesan rahasia dienkripsi terlebih dahulu, kemudian cipherteks disembunyikan di dalam media lain sehingga pihak ketiga tidak menyadari keberadaanya. Pesan rahasia yang disembunyikan dapat diekstraksi kembali persis sama seperti aslinya.

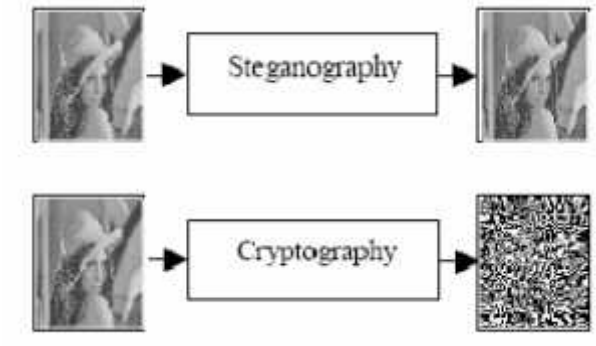
Steganografi membutuhkan dua properti yaitu media penampung dan pesan rahasia. Media penampung yang umum digunakan adalah gambar, suara, video atau teks. Pesan yang disembunyikan dapat berupa sebuah artikel, gambar, daftar barang, kode program, atau pesan lain. Steganografi memiliki 2 proses utama yaitu *embed* dan *extract*. Seperti yang terlihat pada gambar 2.7, proses *embed* merupakan proses penyisipan *hidden object* ke dalam sebuah *cover object*

sehingga menghasilkan *stego file*, sedangkan proses *extract* merupakan proses pengembalian *hidden object* secara utuh setelah disisipkan ke dalam *cover object*.



Gambar 2. 7 Proses Embed dan Extract

Keuntungan steganografi dibandingkan dengan kriptografi adalah bahwa pesan yang dikirim tidak menarik perhatian sehingga media penampung yang membawa pesan tidak menimbulkan kecurigaan bagi pihak ketiga. Ini berbeda dengan kriptografi dimana cipherteks menimbulkan kecurigaan bahwa pesan tersebut merupakan pesan rahasia. Perbedaannya dapat dilihat pada gambar 2.8.



Gambar 2. 8 Perbedaan Kriptografi dan Steganografi

Usia steganografi hampir setua usia kriptografi. Steganografi sudah dikenal oleh bangsa Yunani sejak lama. Herodatus, seorang penguasa Yunani, mengirimkan pesan rahasia menggunakan kepala budak atau prajurit sebagai

media. Caranya, rambut budak dibotaki, lalu pesan rahasia ditulis pada kulit kepala budak. Setelah rambut rambut budak tumbuh cukup banyak (yang berarti menutupi pesan rahasia), budak tersebut dikirim ke tempat tujuan pesan untuk membawa pesan rahasia di kepalanya. Di tempat penerima kepala budak dibotaki kembali untuk membaca pesan rahasia yang tersembunyi di balik rambutnya. Pesan tersebut berisi peringatan tentang invansi dari Bangsa Persia.

Bangsa Romawi mengenal steganografi dengan menggunakan tinta tak-tampak (*invisible ink*) untuk menulis pesan. Jika tinta digunakan untuk menulis maka tulisannya tidak tampak. Tulisan tersebut dapat dibaca dengan cara memanaskan kertas tersebut.

Selama Perang Dunia II, agen-agen spionase juga menggunakan steganografi untuk mengirim pesan. Caranya dengan menggunakan titik-titik yang sangat kecil sehingga keberadaannya tidak dapat dibedakan pada tulisan biasa yang diketik.

Saat ini steganografi sudah banyak diimplementasikan pada media digital. Steganografi digital menggunakan media digital sebagai penampung, seperti citra digital, video digital, atau audio. Informasi yang disembunyikan juga berbentuk digital seperti teks, citra, data audio, atau data video. Steganografi digital dapat dipakai di negara-negara yang menerapkan sensor ketat terhadap informasi atau di negara di mana enkripsi pesan dilarang. Pada negara-negara seperti itu informasi rahasia dapat disembunyikan dengan menggunakan steganografi.

2. 6. 1. Kriteria Steganografi

Dalam menyembunyikan pesan, ada beberapa kriteria dari steganografi digital yang harus dipenuhi antara lain :

1. Impercepbility

   Keberadaan pesan tidak dapat dipersepsi oleh indrawi. Jika pesan disisipkan ke dalam sebuah citra, citra yang telah disisipi pesan harus tidak dapat dibedakan dengan citra asli oleh mata. Begitu pula dengan suara, telinga haruslah mendapati perbedaan antara suara asli dan suara yang telah disisipi pesan.

2. Fidelity

   Mutu media penampung tidak berubah banyak akibat penyisipan. Perubahan yang terjadi harus tidak dapat dipersepsi oleh indrawi.

3. Recovery

   Pesan yang disembunyikan harus dapat diungkap kembali. Tujuan steganografi adalah menyembunyikan informasi, maka sewaktu-waktu informasi yang disembunyikan harus dapat diambil kembali untuk dapat digunakan lebih lanjut sesuai keperluan.

2. 6. 2. Media Cover

Media cover digunakan pada steganografi sebagai media untuk menampung pesan tersembunyi. Pesan yang disembunyikan tersebut dapat mempunyai hubungan atau bahkan tidak mempunyai hubungan sama sekali dengan media dimana pesan tersebut disisipkan atau pesan dapat menyediakan info penting tentang media, seperti informasi autentikasi, judul, tanggal dan waktu pembuatan,

hak cipta, nomor seri kamera digital yang digunakan untuk mengambil gambar, informasi mengenai isi dan akses terhadap citra dan lain sebagainya.

Secara teori, semua berkas yang ada dalam komputer dapat digunakan sebagai media dalam steganografi, asalkan berkas tersebut mempunyai bit-bit data redundan yang dapat dimodifikasi. Beberapa contoh media cover yang digunakan dalam teknik steganografi antara lain :

1. Teks

   Dalam algoritma steganografi yang menggunakan teks sebagai media penyisipannya biasanya digunakan teknik NLP sehingga teks yang telah disisipi pesan rahasia tidak akan mencurigakan untuk orang yang melihatnya.

2. Gambar

   Format ini paling sering digunakan, karena format ini merupakan salah satu format berkas yang sering dipertukarkan dalam dunia internet. Alasan lainnya adalah banyaknya tersedia algoritma steganografi untuk media penampung yang berupa citra/gambar.

3. Audio

   Format ini sering dipilih karena biasanya berkas dengan format ini berukuran relatif besar. Sehingga dapat menampung pesan rahasia dalam jumlah yang besar pula.

4. Video

   Format ini merupakan format dengan ukurn berkas yang relatif sangat besar namun jarang digunakan karena ukurannya yang terlalu besar sehingga

mengurangi kepraktisannya dan juga kurangnya algoritma yang mendukung format ini.

2. 6. 3. Least Significant Bit (LSB)

Teknik ini bekerja secara langsung kedalam domain spasial dari suatu citra. Istilah domain spasial sendiri mengacu pada pixel-pixel yang menyusun citra tersebut, yang artinya teknik ini beroperasi langsung pada pixel-pixel tersebut [10]. Metode ini merupakan metode yang paling umum digunakan untuk menyembunyikan pesan karena memanfaatkan Least Significant Bit (LSB). Walaupun banyak kekurangan pada metode ini, tetapi kemudahan implementasinya membuat metode ini tetap digunakan sampai sekarang.

Metode ini membutuhkan syarat, yaitu jika dilakukan kompresi pada stego, harus digunakan format lossless compression karena metode ini menggunakan bit-bit pada setiap pixel pada citra. Jika digunakan format lossy compression, pesan rahasia yang disembunyikan dapat hilang. Jika digunakan image 24 bit color sebagai cover, sebuah bit dari masing-masing komponen Red, Green, dan Blue dapat digunakan sehingga 3 bit dapat disimpan pada setiap pixel. Sebuah citra 800 x 600 pixel dapat digunakan untuk menyembunyikan 1.440.000 bit (180.000 bytes) data rahasia. Misalnya, di bawah ini terdapat 3 pixel dari citra 24 bit color:

(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)

Jika diinginkan untuk menyembunyikan karakter A (10000001) dihasilkan :

(00100111 11101000 11001000)
(00100110 11001000 11101000)
(11001000 00100110 11101001)

Dapat dilihat bahwa hanya 3 bit saja yang perlu diubah untuk menyembunyikan karakter A ini. Perubahan pada LSB ini akan terlalu kecil untuk terdeteksi oleh mata manusia sehingga pesan dapat disembunyikan secara efektif. Jika digunakan citra 8 bit color sebagai cover, hanya 1 bit saja dari setiap pixel warna yang dapat dimodifikasi sehingga pemilihan citra harus dilakukan dengan sangat hati-hati, karena perubahan LSB dapat menyebabkan terjadinya perubahan warna yang ditampilkan pada citra.

Akan lebih baik jika citra berupa citra grayscale karena perubahan warnanya akan lebih sulit dideteksi oleh mata manusia. Proses ekstraksi pesan dapat dengan mudah dilakukan dengan mengekstrak LSB dari masing-masing pixel pada stego secara berurutan dan menuliskannya ke output file yang berisi pesan tersebut.

2. 6. 4. Dynamic Cell Spreading (DCS)

Teknik Dynamic Cell Spreading (DCS) merupakan steganografi dengan menggunakan model proteksi terhadap deteksi yang dikembangkan oleh *Holger Ohmacht* dengan konsep dasar yaitu menyembunyikan file pesan (semua data elektronik) ke dalam media gambar (JPEG) [2]. Penyembunyian pesan dilakukan dengan cara menyisipkanya pada bit rendah LSB (*Least Significant Bit*) dari data pixel yang menyusun file tersebut menggunakan buffer memori sebagai media penyimpan sementara. Dalam proses penggabungan (stego) antara file gambar dengan teks, untuk file bitmap 24 bit maka setiap pixel (titik) pada gambar tersebut akan terdiri dari susunan tiga warna merah, hijau dan biru (RGB) yang

masing-masing disusun oleh bilangan 8 bit (byte) dari 0 sampai 255 atau dengan format biner 00000000 sampai 11111111. Dengan demikian pada setiap pixel file bitmap 24 bit kita dapat menyisipkan 3 bit data. Teknik Dynamic Cell Spreading ini sebenarnya hampir sama dengan metode LSB, hanya saja pada teknik ini membutuhkan media memori sebagai perantara untuk menghitung jumlah keseluruhan bit yang terdapat di dalam file sehingga memudahkan proses *embedding* itu sendiri.

2. 7. Digital Signature

Bedasarkan sejarahnya, penggunaan digital signature berawal dari penggunaan teknik kriptografi yang digunakan untuk mengamankan informasi yang hendak ditransmisikan/disampaikan kepada orang yang lain yang sudah digunakan sejak ratusan tahun yang lalu. Dalam suatu kriptografi suatu pesan dienkripsi (*encrypt*) dengan menggunakan suatu kunci (*key*). Hasil dari enkripsi ini adalah berupa *chipertext* tersebut kemudian ditransmisikan/diserahkan kepada tujuan yang dikehendakinya. Chipertext tersebut kemudian dibuka/didekripsi (*decrypt*) dengan suatu kunci untuk mendapatkan informasi yang telah enkripsi tersebut. Terdapat dua macam cara dalam melakukan enkripsi yaitu dengan menggunakan kriptografi simetris (*symetric crypthography/secret key crypthography*) dan kriptografi simetris (*asymetric crypthography*) yang kemudian lebih dikenal sebagai *public key crypthography*.

Secret key crypthografi atau yang dikenal sebagai kriptografi simetris, menggunakan kunci yang sama dalam melakukan enkripsi dan dekripsi terhadap suatu pesan (*message)*, disini pengirim dan penerima menggunakan kunci yang

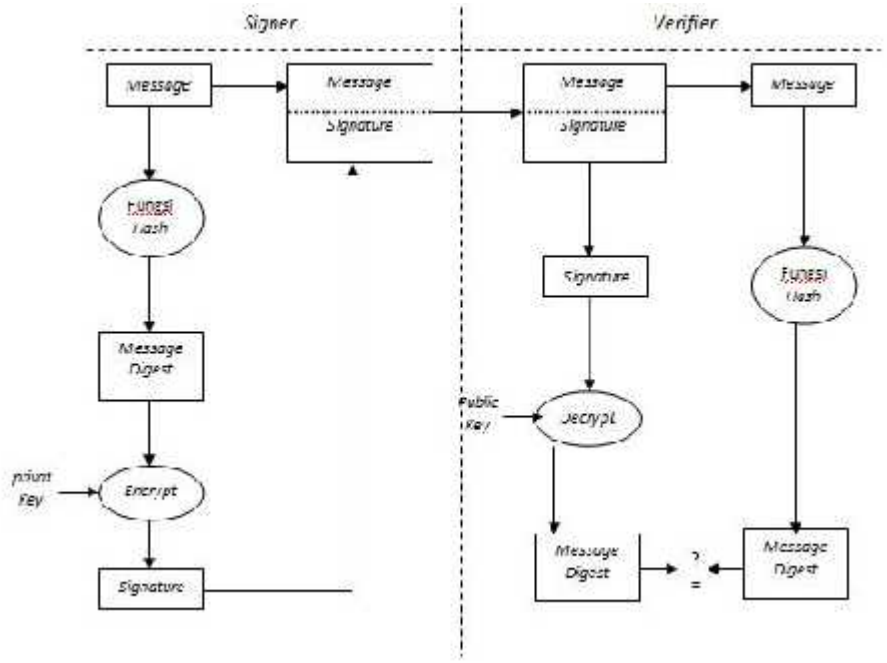
sama sehingga mereka harus menjaga kerahasian (*secret*) terhadap kuci tersebut. Salah satu algoritma yang terkenal dalam kriptografi simetris ini adalah *Data Encryption standard* (DES).

Tanda tangan digital (*digital signature*) adalah mekanisme otentikasi yang memungkinkan pembuat pesan menambahkan sebuah kode yang bertindak sebagai tanda tangannya [1]. Tanda tangan dihasilkan berdasarkan pesan yang diinginkan pesan yang ingin ditandatangani dan berubah-ubah sesuai dengan pesan. Tanda tangan digital dikirmkan bersama-sama dengan pesan kepada penerima.Tanda tangan digital memungkinkan penerima informasi untuk menguji terlebih dahulu keaslian informasi yang didapat dan juga untuk meyakinkan bahwa data yang diterima itu dalam keadaaan utuh. Oleh karena itu, tanda tangan digital kunci publik (*public key gital signature*) menyediakan layanan *authentication* (keaslian) dan *data integrity* (keutuhan data). Selain itu tanda tangan digital juga menyediakan layanan *non-repudiation* yang artinya melindungi pengirim dari sebuah klaim yamg menyatakan bahwa dia telah mengirim informasi padahal tidak.Tanda tangan digital memberikan pelayanan yang tujuannya sama dengan tanda tangan (berupa tulisan tangan biasa). Tetapi bagaimanapun juga tanda tangan berupa tulisan tangan relatif lebih mudah ditiru/dipalsukan oleh orang lain. Sedangkan tanda tangan digital hampir tudak mungkin dipalsukan, bahkan bisa berfungsi ganda yaitu sekaligus memperlihatkan sekilas isi informasi mengenai identitas yang menandatangani.

Bahkan beberapa orang lebih cenderung menggunakan tanda tangan digital dari pada menggunakan enkripsi biasa. Contohnya, seseorang mungkin tidak

perduli jika setiap orang mengetahui bahwa dia telah mentransfer uang sejumlah satu milyar ke rekening lain, tetapi orang tersebut pasti ingin tahu dan lebih peduli apakah *teller* bank tersebut benar-benar asli, benar-benar ada dan benar-benar sah. Perasaan tidak tenang akan *teller* gadungan dalam sistem transaksi lewat *internet* sangat wajar terjadi.

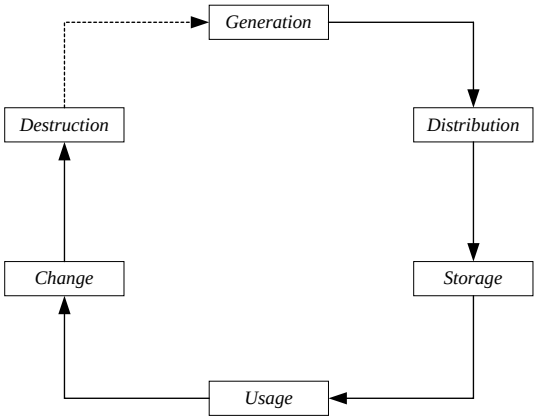
Pada konsep yang telah diketahui sebelumnya, proses enkripsi informasi biasanya menggunakan kunci publik. Tetapi pada konsep tanda tangan digital, informasi justru dibubuhi tanda tangan digital (dienkripsi) dengan kunci rahasia yang dimilki sumber. Apabila informasi tadi bisa diverifikasi (didekripsi) dengan kunci publik sumber yang telah tersebar ini berarti bahwa informasi tersebut adalah benar-benar asli dari sumber. Konsep tanda tangan digital dapat dilihat pada gambar 2.9.



Gambar 2. 9 Konsep Umum *Digital Signature*

2. 7. 1. Manajemen Kunci

Kekuatan sistem kriptografi secara total bergantung pada keamanan kunci. Kunci perlu dilindungi selama fase daur hidupnya. Daur hidup kunci dimulai dari pembangkitan kunci (*generation*) sampai kunci tidak diperlukan lagi untuk kemudian dihancurkan (*destruction*). Secara garis besar, daur hidup kunci digambarkan pada Gambar 2.10.



Gambar 2. 10 Daur hidup kunci

Tujuan manajemen kunci adalah menjaga keamanan dan integritas kunci pada semua fase di dalam daur hidupnya. Pada umumnya setiap kunci akhirnya diganti dengan kunci lain. Jadi, keseluruhan fase membentuk siklus (lingkaran) karena penghancuran kunci biasanya diikuti dengan penggantiannya dengan kunci baru (digambarkan dengan garis putus-putus). Manajemen kunci yang dibahas difokuskan pada algoritma kriptografi simetri karena manajemen kunci untuk algoritma kunci-publik sangat berbeda dengan algoritma simetri.

2. 7. 2. Digital Signature Algorithm (DSA)

Pada beberapa kasus seringkali otentikasi yang diperlukan tetapi kerahasiaan pesan tidak [1]. Maksudnya, pesan tidak perlu dienkripsikan, sebab

yang dibutuhkan hanya keotentikan pesan saja. Kebutuhan tersebut dapat dipenuhi dengan pemberian tandatangan *digital*. Algoritma kunci-publik dan fungsi hash dapat digunakan untuk kasus seperti ini. Tanda tangan *digital* adalah suatu nilai kriptografis yang bergantung pada pesan dan pengirim pesan.

Pada Agustus 1991, NIST (*The National of Standart and Technology*) mengumumkan standard untuk tanda tangan *digital* yang dinamakan *Digital Signature Standard* (*DSS*) yang terdiri dari dua komponen :

a. Algoritma tandatangan *digital* yang disebut *DSA* (*Digital Signature* Algorithm)

b. Fungsi Hash yang disebut *SHA* (*Secure Hash Algorithm*)

DSS adalah standard, sedangkan *DSA* adalah algoritma. Standard tersebut menggunakan algoritma *DSA* untuk penandatanganan pesan dan *SHA* untuk membangkitkan message digest dari pesan. *DSA* tidak dapat digunakan untuk enkripsi. *DSA* dispesifikasikan khusus untuk tanda tangan digital. *DSA* mempunyai dua fungsi utama:

1. Pembentukan tanda tangan (*signature generation*), dan

2. Pemeriksaan keabsahan tanda tangan (*signature verification*).

Sebagaimana halnya pada algoritma kriptografi kunci-publik, *DSA* menggunakan dua buah kunci, yaitu kunci publik dan kunci privat. Pembentukan tanda tangan menggunakan kunci rahasia privat, sedangkan verifikasi tanda tangan menggunakan kunci publik pengirim. DSA menggunakan fungsi *hash SHA* (*Secure Hash Algorithm*) untuk mengubah pesan menjadi *message digest* yang berukuran 160 bit.

DSA dikembangkan dari algoritma ElGamal. *DSA* mempunyai properti berupa parameter sebagai berikut:

1. p, adalah bilangan prima dengan panjang L bit, yang dalam hal ini $512 \geq L \leq 1024$ dan L harus kelipatan 64. Parameter p bersifat publik dan dapat digunakan bersama sama oleh orang di dalam kelompok.
2. q, bilangan prima 160 bit, merupakan faktor dari $p - 1$. Dengan kata lain, $(p - 1) \bmod q = 0$. Parameter q bersifat publik.
3. $g = h^{(p-1)/q} \bmod p$, yang dalam hal ini $h < p - 1$ sedemikian sehingga $h^{(p-1)/q} \bmod p > 1$. Parameter g bersifat publik.
4. x, adalah bilangan bulat kurang dari q. Parameter x adalah kunci privat.
5. $y = g^x \bmod p$, adalah kunci publik.
6. m, pesan yang akan diberi tanda-tangan.

Prosedur Pembangkitan Sepasang Kunci

1. Pilih bilangan prima p dan q, yang dalam hal ini $(p - 1) \bmod q = 0$.
2. Hitung $g = h^{(p-1)/q} \bmod p$, yang dalam hal ini $1 < h < p - 1$ dan $h^{(p-1)/q} \bmod p > 1$.
3. Tentukan kunci privat x, yang dalam hal ini $x < q$.
4. Hitung kunci publik $y = g^x \bmod p$.

   Jadi, prosedur di atas menghasilkan:

   kunci publik dinyatakan sebagai (p, q, g, y);

   kunci privat dinyatakan sebagai (p, q, g, x).

Prosedur Pembangkitan Tanda-tangan (*Signing*)

1. Ubah pesan *m* menjadi *message digest* dengan fungsi *hash* SHA, *H*.
2. Tentukan bilangan acak $k < q$.
3. Tanda-tangan dari pesan *m* adalah bilangan *r* dan *s*. Hitung *r* dan *s* sebagai berikut:

   $r = (g^k \bmod p) \bmod q$

   $s = (k^{-1} (H(m) + x * r)) \bmod q$
4. Kirim pesan *m* beserta tanda-tangan *r* dan *s*.

Prosedur Verifikasi Keabsahan Tanda-tangan (*Verifying*)

1. Hitung

   $w = s^{-1} \bmod q$

   $u1 = (H(m) * w) \bmod q$

   $u2 = (r * w) \bmod q$

   $v = ((g^{u1} * y^{u2}) \bmod p) \bmod q)$
2. Jika $v = r$, maka tanda-tangan sah, yang berarti bahwa pesan masih asli dan dikirim oleh pengirim yang benar.

2. 8. Fungsi HASH

Sebuah fungsi *hash* (*hash function* atau *hash algorithm*) adalah suatu cara untuk menghasilkan sebuah *digital* "*fingerprint*" kecil dari sembarang data [1]. Fungsi ini memecahkan dan mencampurkan data untuk menghasilkan *fingerprint* yang sering disebut sebagai nilai *hash* (*hash value*). Nilai *hash* ini sering direpresentasikan dengan sebuah *string* pendek dari huruf-huruf dan angka-angka yang kelihatan acak (berbentuk heksadesimal). Sebuah fungsi *hash* yang baik

adalah suatu fungsi yang tidak (jarang) memiliki output nilai *hash* yang sama untuk input yang berbeda.

SHA adalah fungsi *hash* satu-arah yang dibuat oleh *NIST* dan digunakan bersama *DSS* (*Digital Signature Standard*). Oleh *NSA*, *SHA* dinyatakan sebagai standard fungsi *hash* satu-arah. *SHA* didasarkan pada *MD4* yang dibuat oleh Ronald L. Rivest dari *MIT*.

SHA disebut aman (*secure*) karena ia dirancang sedemikian rupa sehingga secara komputasi tidak mungkin menemukan pesan yang berkoresponden dengan *message digest* yang diberikan. Algoritma SHA menerima masukan berupa pesan dengan ukuran maksimum 264 bit (2.147.483.648 *gigabyte*) dan menghasilkan *message digest* yang panjangnya 160 bit [Gambar 2.11], lebih panjang dari *message digest* yang dihasilkan oleh *MD5*. Langkah-langkah pembuatan *message digest* secara garis besar adalah sebagai berikut:

1. Penambahan bit-bit pengganjal (*padding bits*).
2. Penambahan nilai panjang pesan semula.
3. Inisialisasi penyangga (*buffer*) MD.
4. Pengolahan pesan dalam blok berukuran 512 bit.

Algorithm	Message Size (bits)	Block Size (bits)	Word Size (bits)	Message Digest Size (bits)	Security[2] (bits)
SHA-1	$< 2^{64}$	512	32	160	80
SHA-256	$< 2^{64}$	512	32	256	128
SHA-384	$< 2^{128}$	1024	64	384	192
SHA-512	$< 2^{128}$	1024	64	512	256

Gambar 2. 11 Daftar properti dari keempat SHA

2. 9. Pemodelan Sistem

Dalam suatu proses pembuatan software, analisa dan rancangan telah merupakan terminologi yang sangat tua. Pada saat masalah ditelusuri dan spesifikasi dinegisiasikan, dapat dikatakan bahwa kita berada pada tahap rancangan. Merancang adalah menemukan suatu cara untuk menyelesaikan masalah, salah satu tool/model untuk merancang pembangunan software yang berbasis *object-oriented* adalah UML. Alasan mengapa UML digunakan adalah, pertama, *scalability* dimana objek lebih mudah dipakai untuk menggambarkan sistem yang besar dan komplek. Kedua, *dynamic modeling*, dapat dipakai untuk pemodelan sistem dinamis dan real time [11]. Sebagaimana dalam tulisan pertama, akan dijelaskan konsep mengenai obyek, OOAD (*Obyek Oriented Analyst/ Design*) dan pengenalan UML, maka dalam tulisan kedua ini lebih ditekankan pada cara bagaimana UML digunakan dalam merancang sebuah pengembangan *software* yang disertai gambar atau contoh dari sebuah aplikasi.

2. 9. 1. UML (Unified Modelling Language)

Unified Modelling Language (UML) adalah sebuah ”bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan 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,

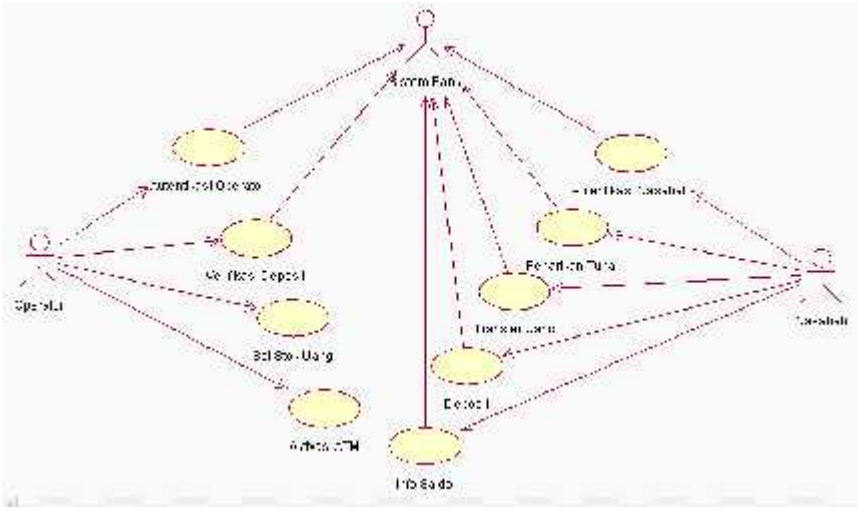
maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa bahasa berorientasi objek.

Seperti bahasa-bahasa lainnya , UML mendefinisikan notasi dan syntax. 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 di kombinasikan. Notasi Uml terutama diturunkan dari 3 notasi yang telah ada sebelumnya yaitu : Grady Booch OOD (*Object Oriented Design*), Jim Rumbaugh OMT(Object Modeling Technique) dan Ivar Jacobson OOSE (*Object Oriented Software Engineering*).

2. 9. 2. Use case

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.

Sebuah *use case* menggambarkan suatu urutan interaksi antara satu atau lebih aktor dan sistem. Dalam fase *requirements*, model *use case* mengambarkan sistem sebagai sebuah kotak hitam dan interaksi antara aktor dan sistem dalam suatu bentuk naratif, yang terdiri dari input user dan respon-respon sistem. Setiap use case menggambarkan perilaku sejumlah aspek sistem, tanpa mengurangi struktur internalnya. Selama pembuatan model use case secara pararel juga harus ditetapkan obyek-obyek yang terlibat dalam setiap use case. Contoh use case dapat dilihat pada gambar 2.12.



Gambar 2. 12 Use Case Diagram

2. 9. 3. 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).

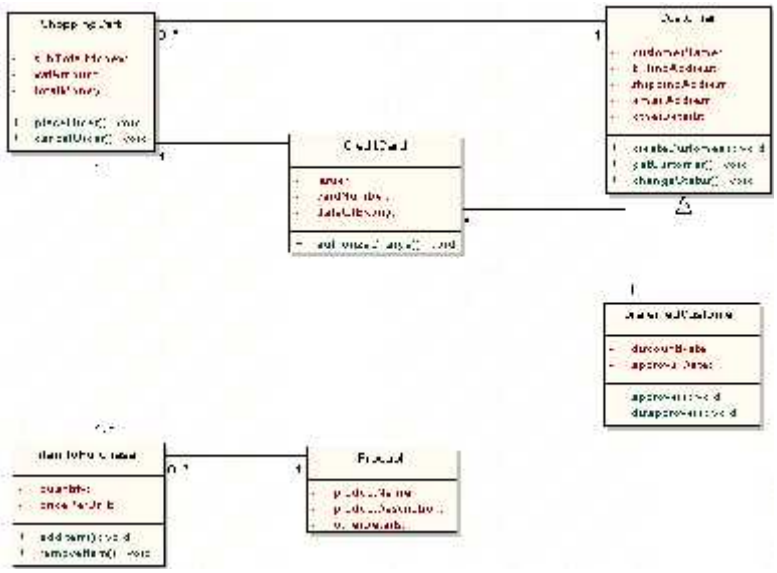
Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment,* pewarisan, asosiasi, dan lain-lain [Gambar 2.13].

Class memiliki tiga area pokok :

1. Nama (*danstereotype*)
2. Atribut
3. Metoda

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

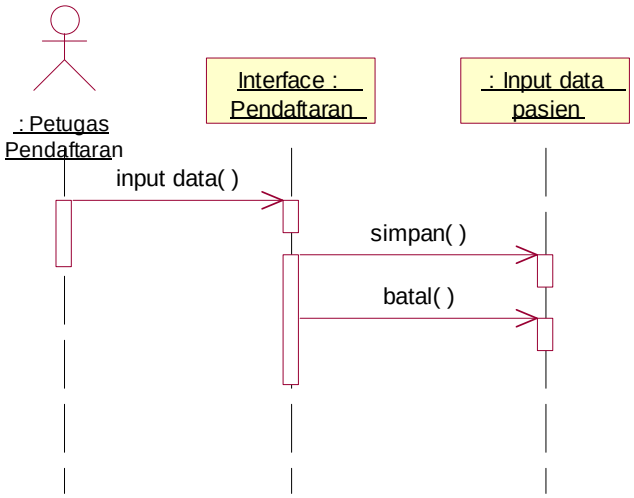


Gambar 2. 13 Class Diagram

2. 9. 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. Contoh *sequence diagram* dapat dilihat pada gambar 2.14.

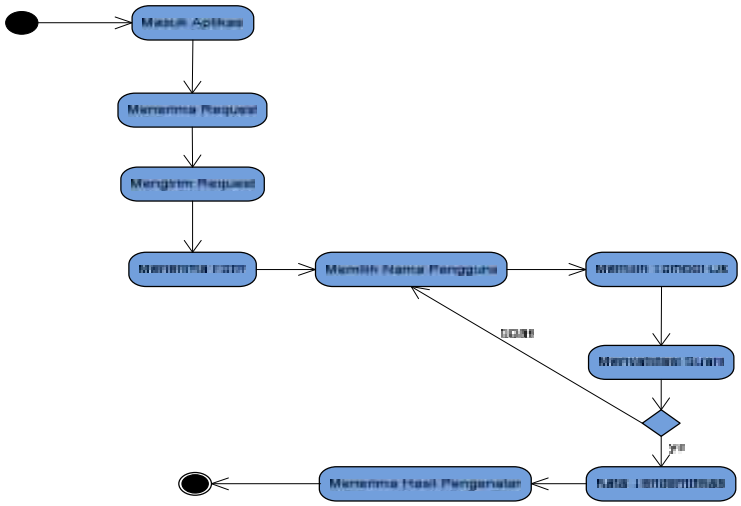


Gambar 2. 14 Sequence Diagram

2. 9. 5. 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.

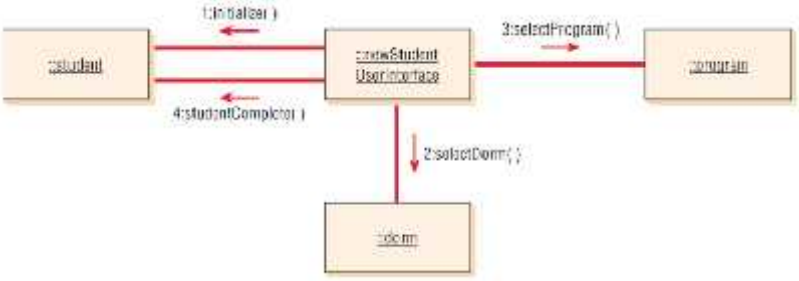
Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu activity diagram tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Contoh *activity diagram* dapat dilihat pada gambar 2.15.



Gambar 2. 15 Activity Diagram

2. 9. 6. Collaboration Diagram

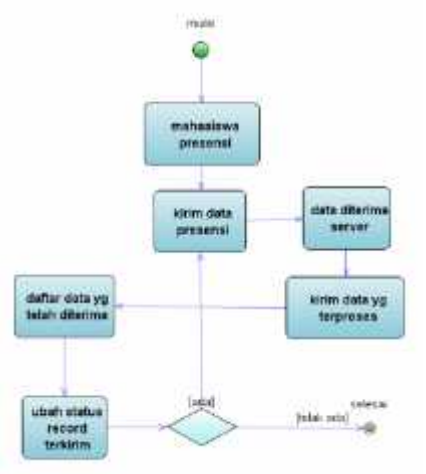
Menunjukan informasi yang sama seperti dalam sequence diagram. Perbedaannya adalah cara mengelompokan object sequence berdasarkan urutan nomber dari pesan. Contoh *collaboration diagram* dapat dilihat pada gambar 2.16.



Gambar 2. 16 Collaboration Diagram

2. 9. 7. State Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya statechart diagram menggambarkan class tertentu (satu class dapat memiliki lebih dari satu statechart diagram). Contoh *state diagram* dapat dilihat pada gambar 2.17.



Gambar 2. 17 State Diagram

2. 10. Sistem Operasi Java

Sistem operasi biasanya ditulis dalam sebuah kombinasi dari kode bahasa C dan assembly, terutama disebabkan oleh kelebihan performa dari bahasa tersebut dan memudahkan komunikasi dengan perangkat keras. Satu kesulitan dalam merancang sistem basis bahasa adalah dalam hal proteksi memori, yaitu memproteksi sistem operasi dari pemakai program yang sengaja memproteksi pemakai program lainnya. Sistem operasi tradisional mengharapkan pada tampilan perangkat keras untuk menyediakan proteksi memori. Sistem basis bahasa mengandalkan pada tampilan keamanan dari bahasa. Sebagai hasilnya, sistem basis bahasa menginginkan pada alat perangkat keras kecil, yang mungkin kekurangan tampilan perangkat keras yang menyediakan proteksi memori.

Java adalah sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990. Menurut definisi dari Sun, Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan. Kita lebih menyukai menyebut Java sebagai sebuah teknologi dibanding hanya sebuah bahasa

pemrograman, karena Java lebih lengkap dibanding sebuah bahasa pemrograman konvensional. Teknologi Java memiliki tiga komponen penting, yaitu:

a. Programming-language specification
b. Application-programming interface
c. Virtual-machine specification

2. 10. 1. Java2

Java2 adalah generasi kedua dari Java platform (generasi awalnya adalah Java Development Kit). Java berdiri di atas sebuah mesin interpreter yang diberi nama JVM. JVM inilah yang akan membaca bytecode dalam file .class dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu, bahasa Java disebut sebagai bahasa pemrograman yang portable karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM. Platform Java terdiri dari kumpulan library, JVM, kelas-kelas loader yang dipaket dalam sebuah lingkungan rutin Java, dan sebuah compiler, debuger, dan perangkat lain yang dipaket dalam Java Development Kit (JDK). Java2 adalah generasi yang sekarang sedang berkembang dari platform Java.

2. 11. Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah serangkaian kegiatan yang dilakukan dengan maksud untuk menemukan sebuah kesalahan. Hal ini juga dapat memastikan bahwa perangkat lunak dapat bekerja dengan baik dan sesuai dengan kebutuhan pengguna. Pengujian pada perangkat lunak terbagi menjadi 2, yaitu *white box testing* dan *black box testing*.

2. 11. 1. *White Box Testing*

White box testing adalah metode pengujian perangkat lunak yang menguji strukur internal atau cara kerja perangkat lunak [12]. Dengan menggunakan metode *white box*, penguji sistem akan memperoleh kasus uji yang :

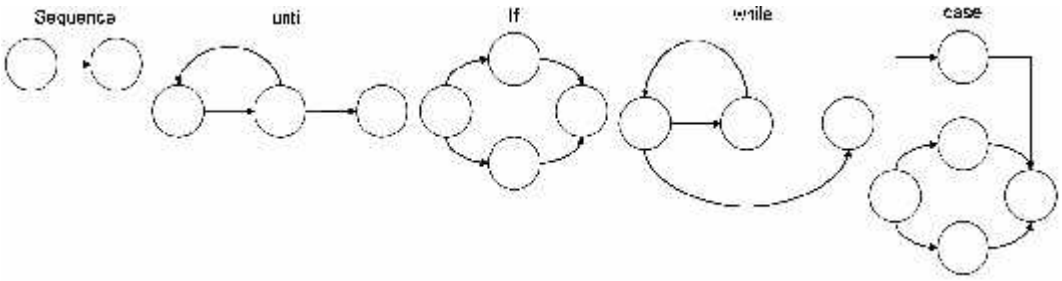
a. Menjamin seluruh *independent path* di dalam modul yamg dikerjakan sekurang-kurangnya 1 kali.
b. Mengerjakan seluruh keputusan logika.
c. Mengerjakan seluru loop yang sesuai dengan batasannya.
d. Mengerjakan seluruh struktur data internal yang menjamin validasi.

Menurut Ehmer Khan, metode pengujian *white box* meliputi :

a. *Control Flow Testing*
b. *Branch Testing*
c. *Basis Path Testing*
d. *Data Flow Testing*
e. *Loop Testing*

Metode *basis path testing* merupakan metode uji coba yang diusulkan oleh *Tom McCabe*. Metode ini memungkinkan perancang kasus uji mendapatkan ukuran kekompleksan logika dari perancangan prosedural dan menggunakan ukuran ini sebagai petunjuk untuk mendefinisikan basis set dari jalur pengerjaan. Kasus uji yang didapat digunakan untuk mengerjakan basis set yang menjamin pengerjaan setiap perintah minimal satu kali selama uji coba. Metode *basis path testing* meliputi :

a. *Flowgraph Notation*

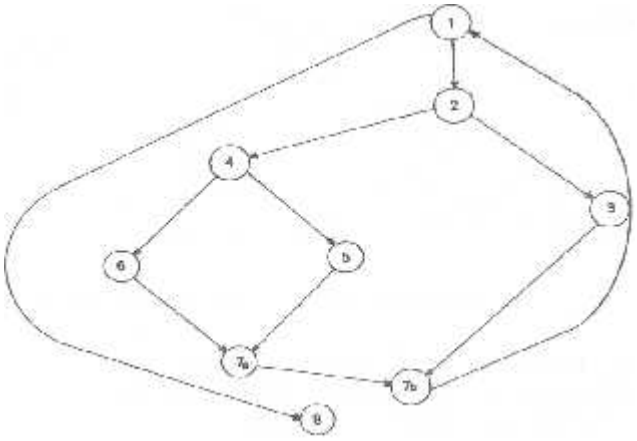


Gambar 2. 18 *Flowgraph Notation*

Pada Gambar 2.18, lingkaran *(node)*, menggambarkan satu/lebih perintah prosedural. Urutan proses dan keputusan dapat dipetakan dalam satu *node*. Tanda panah *(edge)*, menggambarkan aliran kontrol. Setiap *node* harus mempunyai tujuan *node*. *Region* adalah daerah yang dibatasi oleh edge dan node. Termasuk daerah diluar grafik alir. Contoh menterjemahkan *pseudo code* ke *flowgraph.*

Tabel 2. 3 Contoh Pseudo Code

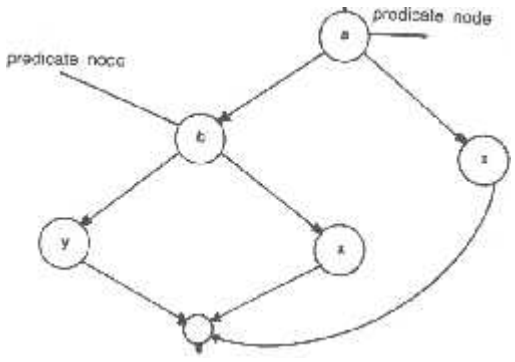
1	do while record masih ada baca record
2	if record ke 1 = 0
3	then proses record simpan di buffer naikan counter
4	else if record ke 2 = 0
5	then riser counter
6	proses record simpan pada file
7a	endif endif
7b	enddo
8	End



Gambar 2. 19 Contoh *Flowgraph* Dari *Pseudo Code*

Gambar 2.19 merupakan contoh *flowgraph notation* berdasarkan *pseudo code* pada tabel 2.3. Nomor pada *pseudo code* berhubungan dengan nomor *node*. Apabila ditemukan kondisi majemuk (*compound condition*) pada *pseudo code* pembuatan grafik alir menjadi rumit. Kondisi majemuk mungkin terjadi pada operator *Boolean* (*AND, OR, NAND, NOR*) yang dipakai pada perintah *if*. Contoh :

```
if A or B then
     procedure x
  else
     procedure y
endif
```



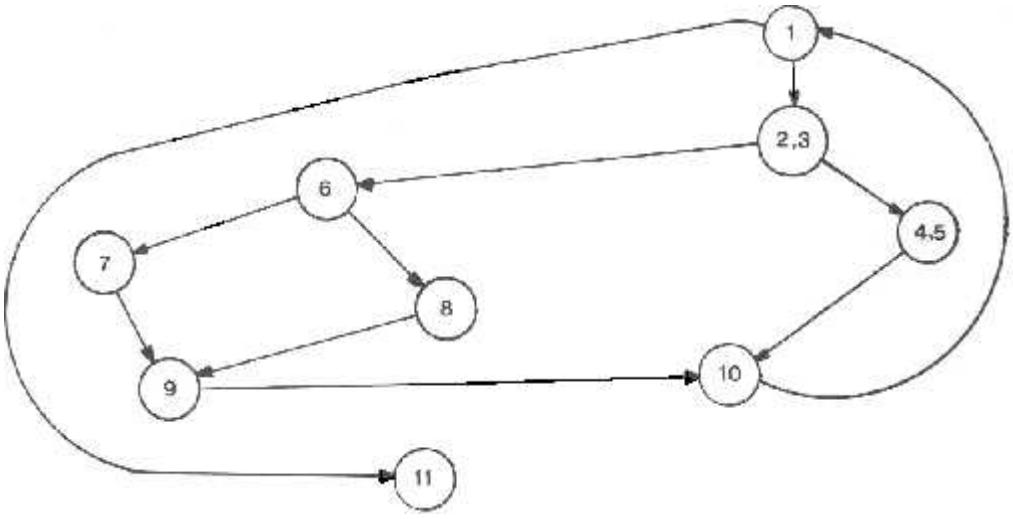
Gambar 2. 20 Logika Gabungan

Pada Gambar 2.20, *node* dibuat terpisah untuk masing-masing kondisi A dan B dari pernyataan IF A OR B. Masing-masing *node* berisi kondisi yang disebut *predicate node* dan mempunyai karakteristik dua atau lebih edge dari *node*.

b. *Cyclomatic Complexity*

Cyclomatic complexity adalah metrik *software* yang menyediakan ukuran kuantitatif dari kekompleksan logikal program. Apabila digunakan dalam konteks metode uji coba *basis path*, nilai yang dihitung untuk *cyclomatic complexity* menentukan jumlah jalur independen dalam basis set suatu program dan memberi batas atas untuk jumlah uji coba yang harus dikerjakan untuk menjamin bahwa seluruh perintah sekurang-kurangnya telah dikerjakan sekali.

Jalur *independent* adalah jalur yang melintasi atau melalui program dimana sekurang-kurangnya terdapat proses perintah yang baru atau kondisi yang baru.



Gambar 2. 21 Contoh *Flowgraph* Untuk Menghitung *Independent Path*

Dari gambar 2.21, didapatkan jalur *independent* :

Path 1 = 1 - 11

Path 2 = 1 - 2 - 3 - 4 - 5 - 10 - 1 - 11

Path 3 = 1 - 2 - 3 - 6 - 8 - 9 - 10 - 1 - 11

Path 4 = 1 - 2 - 3 - 6 - 7 - 9 - 10 - 1 - 11

Path 1,2,3,4 yang telah didefinisikan merupakan basis set untuk diagram alir.

Cyclomatic complexity digunakan untuk mencari jumlah *path* dalam satu *flowgraph*. Dapat dipergunakan rumusan sebagai berikut :

1. Jumlah region grafik alir sesuai dengan *cyclomatic complexity*.
2. *Cyclomatix complexity* V(G) untuk grafik alir dihitung dengan rumus:

   V(G) = E - N + 2

   Dimana:

   E = jumlah *edge* pada grafik alir

   N = jumlah *node* pada grafik alir
3. Cyclomatix complexity V(G) juga dapat dihitung dengan rumus:

   V(G) = P + 1

   Dimana P = jumlah predicate node pada grafik alir

Pada Gambar 2.21 dapat dihitung *cyclomatic complexity*:

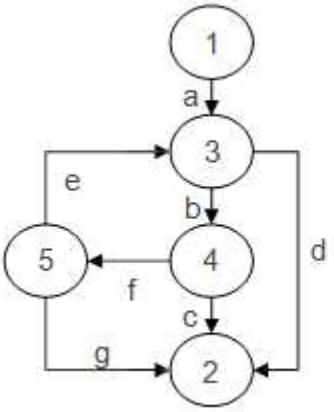
1. *Flowgraph* mempunyai 4 region
2. V(G) = 11 edge - 9 node + 2 = 4
3. V(G) = 3 predicate node + 1 = 4

Jadi cyclomatic complexity untuk flowgraph adalah 4

c. *Graph Matrix*

Graph matrix merupakan *software* yang dikembangkan untuk membantu uji coba *basis path* atau struktur data. *Graph matrix* adalah *matrix* empat persegi yang mempunyai ukuran yang sama dengan jumlah node pada *flow graph*. Masing-masing baris dan kolom mempunyai hubungan dengan node yang telah ditentukan dan pemasukan data *matrix* berhubungan dengan hubungan (*edge*) antar *node.*

Contoh sederhana pemakaian *graph matrix* dapat dilihat pada Gambar 2.22.



	1	2	3	4	5
1			a		
2					
3				b	
4		c			f
5		g	e		

Gambar 2. 22 *Graph Matrix*

Hubungan bobot menyediakan tambahan informasi tentang aliran kontrol. Secara simpel hubungan bobot dapat diberi nilai 1 jika ada hubungan antara node atau nilai 0 jika tidak ada hubungan. Dapat juga hubungan bobot diberi tanda dengan:

1. Kemungkinan link (*edge*) dikerjakan
2. Waktu yang digunakan untuk proses selama traversal dari link

3. Memori yang diperlukan selama *traversal link*
4. Sumber daya yang diperlukan selama *traversal link*

2. 11. 2. *Black Box Testing*

Black box testing terfokus pada spesifikasi fungsional dari perangkat lunak. Penguji dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program. *Black box testing* bukanlah solusi alternatif dari *white box testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *white box testing* [13].

Black box testing cenderung untuk menemukan hal-hal berikut:

a. Fungsi yang tidak benar atau tidak ada
b. Kesalahan antarmuka *(interface errors)*
c. Kesalahan pada struktur data dan akses basis data
d. Kesalahan performansi *(performance errors)*
e. Kesalahan inisialisasi dan terminasi.

Berikut ini adalah metode-metode yang digunakan di dalam *black box testing* :

a. *Equivalence Partitioning*

i. Membagi input menjadi kelas-kelas data yang dapat digunakan untuk menggenerasi kasus uji.
ii. Bertujuan untuk menemukan kelas-kelas kesalahan.
iii. Berdasarkan pada kesamaan kelas-kelas kondisi input.
iv. Sebuah kelas yang ekivalen merepresentasikan kumpulan status/kondisi yang valid atau tidak valid

v. Sebuah kondisi input dapat berupa nilai numerik yang spesifik, rentang nilai, kumpulan nilai yang berkaitan, atau kondisi *boolean.*

b. *Boundary Value Analysis/Limit Testing*

i. Banyak kesalahan terjadi pada kesalahan masukan.

ii. BVA mengijinkan untuk menyeleksi kasus uji yang menguji batasan nilai input.

iii. BVA merupakan komplemen dari *equivalence partitioning.* Lebih pada memilih elemen-elemen di dalam kelas ekivalen pada bagian sisi batas dari kelas.

c. *Comparison Testing*

i. Pada beberapa aplikasi *reliability* dari sebuah perangkat lunak sangat penting.

ii. Redundansi perangkat keras dan perangkat lunak mungkin digunakan untuk meminimalisir kesalahan *(error).*

d. *Sample Testing*

i. Melibatkan beberapa nilai yang terpilih dari sebuah kelas ekivalen

ii. Mengintegrasikan nilai pada kasus uji

iii. Nilai-nilai yang terpilih mungkin dipilih dengan urutan tertentu atau interval tertentu

e. *Robustness Testing*

i. Data input dipilih diluar spesifikasi yang telah didefinisikan

ii. Tujuan dari pengujian ini adalah membuktikan bahwa tidak ada kesalahan jika masukan tidak valid

f. *Behavior Testing*

Hasil uji tidak dapat dievaluasi jika hanya melakukan pengujian sekali, tapi dapat dievaluasi jika pengujian dilakukan beberapa kali, misalnya pada pengujian struktur data stack

g. *Requirement Testing*

i. Spesifikasi kebutuhan yang terasosiasi dengan perangkat lunak (*input*/*output*/fungsi/performansi) diidentifikasi pada tahap spesifikasi kebutuhan dan desain.

ii. Requirement testing melibatkan pembuatan kasus uji untuk setiap spesifikasi kebutuhan yang terkait dengan program

iii. Untuk memfasilitasinya, setiap spesifikasi kebutuhan bisa ditelusuri dengan kasus uji dengan menggunakan *traceability matrix*

h. *Performance Testing*

i. Mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan misalnya: aliran data, ukuran pemakaian memori, kecepatan eksekusi, dll

ii. Untuk mencari tahu beban kerja atau kondisi konfigurasi program

iii. Spesifikasi mengenai performansi didefinisikan pada saat tahap spesifikasi atau desain

iv. Dapat digunakan untuk menguji batasan lingkungan program

i. Uji Ketahanan *(Endurance Testing)*

Endurance Testing melibatkan kasus uji yang diulang-ulang dengan jumlah tertentu dengan tujuan untuk mengevaluasi program apakah sesuai dengan spesifikasi kebutuhan.

j. Uji Sebab-Akibat *(Cause-Effect Relationship Testing)*

i. Teknik ini merupakan suplemen dari *equivalence testing* dengan menyediakan cara untuk memilih kombinasi data input

ii. Melibatkan kondisi input *(cause)* dan kondisi output *(effect)* untuk mencegah pendefinisian kasus uji yang terlalu banyak