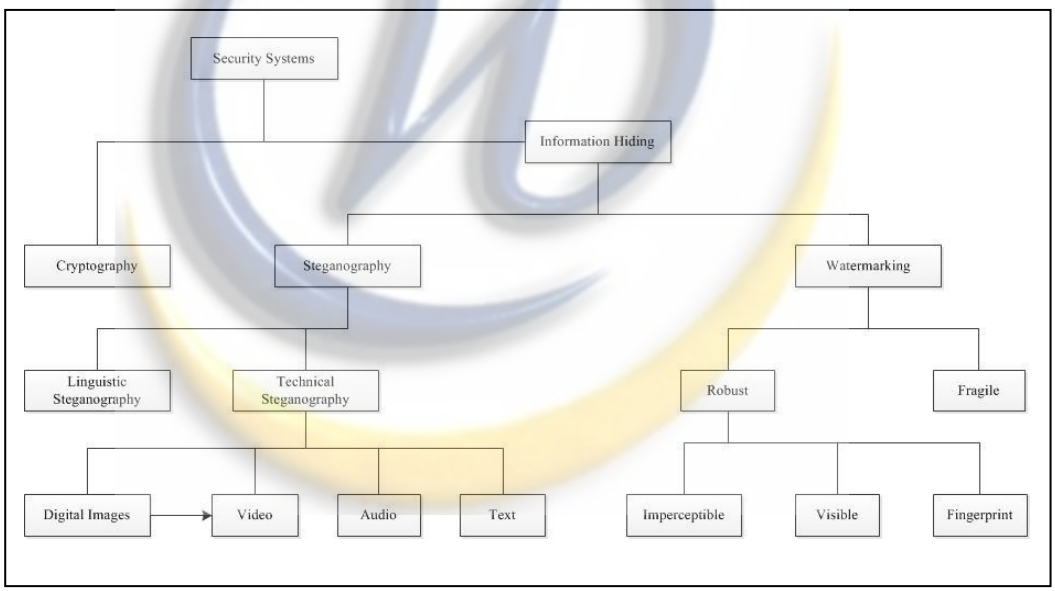
BAB II
LANDASAN TEORI

Pada bab ini akan diuraikan tentang teori-teori yang melandasi penulisan Laporan Penelitian ini.

2.1 Steganografi

Steganografi adalah teknik untuk menyembunyikan informasi. Nama steganografi diambil dari kerja yang dilakukan Trithemus (1462-1516) yang berjudul "Steganographia". Steganografi berasal dari bahasa Yunani yang berarti "tulisan tertutup", dengan kata lain yaitu menyembunyikan informasi dibalik informasi lainnya.[12]



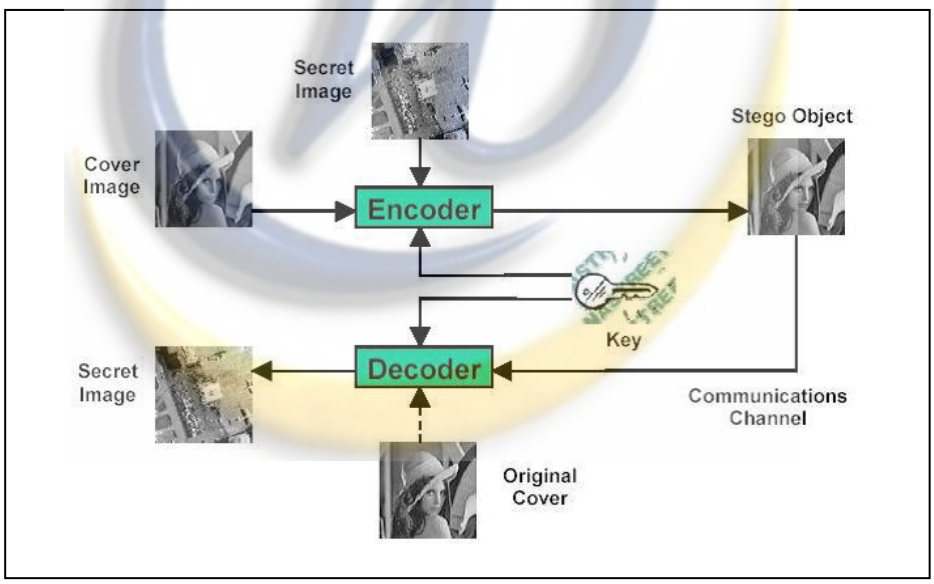
Gambar 2.1 Klasifikasi Teknik Penyembunyian Informasi.[12]

Dapat dilihat dari gambar 2.1 yang berisi klasifikasi teknik penyembunyian informasi, bahwa steganografi merupakan salah satu teknik yang berada didalamnya.

2.2 *Digital Watermarking*

Secara hierarkis, *watermarking* merupakan suatu proses yang berakar pada konsep ilmu *steganography*. *Steganography* sendiri sudah dikenal sejak jaman Mesir kuno. Menurut Popa, *steganography* dapat dibagi menjadi 2 (dua) bagian yaitu *protection against detection* (*data hiding*) dan *protection against removal* (*document marking*). *Watermarking* merupakan salah satu jenis dari *document marking*.

Watermarking merupakan teknik penyisipan data ke dalam elemen *multimedia* seperti citra, audio atau *video*. Data yang disisipkan tersebut kemudian harus dapat diekstrak atau dideteksi berada di dalam *multimedia* tersebut. Dari pengertian tersebut, terdapat dua proses utama dalam *watermarking*, yaitu proses menyisipkan data (*encode*) dan proses mengekstrak data (*decode*). Secara umum proses yang terjadi dalam *watermarking* terlihat dalam gambar berikut ini.



Gambar 2.2 Proses Watermarking.[3]

Dari gambar tersebut terlihat bahwa gambar asli akan di-*encode* dengan menambahkan gambar rahasia dan kunci tertentu. Gambar yang sudah di-*encode*, selanjutnya dapat ditransfer melalui suatu jalur komunikasi dan jika akan diperiksa keasliannya atau ingin mendapatkan gambar aslinya, maka dilakukan proses *decoder*. Proses *decoder key* (kunci) yang sama dengan kunci pada proses *encode*.

Menurut beberapa peneliti, teknik *digital watermarking* dapat diaplikasikan dalam berbagai hal, antara lain:

1. ***Ownership Assertion***. Kepemilikan dari dokumen *multimedia* dapat dilindungi dengan menambahkan *watermark* yang berisi informasi pemilik dari dokumen *multimedia*. Pemilik juga dapat mempublikasikan dokumen *multimedia* yang sudah disisipi *watermark* dengan aman tanpa harus mempublikasikan dokumen *multimedia* yang asli. Jika terjadi terjadi klaim dari orang lain mengenai dokumen *multimedia* tersebut, tentu dapat diketahui secara otentik siapa pemilik sebenarnya.[5]
2. ***Fingerprinting***. *Watermarking* dan *fingerprinting* pada dasarnya sama, hanya saja pada *fingerprinting,* penyisipan *watermark* biasanya bersifat unik untuk suatu dokumen *multimedia*. Dokumen *multimedia* yang sama dapat memiliki *fingerprint* yang berbeda.[5]
3. ***Copy prevention or control***. Teknik *watermarking* juga dapat dilakukan untuk mencegah dokumen *multimedia* untuk diduplikasi dengan *hardware* atau *software* tertentu. Misalnya untuk mencegah suatu dokumen *multimedia* yang tersimpan dalam *CD* atau *DVD* agar tidak diduplikasi dengan *CD* atau *DVD copier*. [5]
4. ***Fraud and tamper detection***. *Watermarking* juga dapat digunakan untuk mendeteksi adanya pembajakan terhadap suatu dokumen *digital*. [5]
5. ***ID card security***. Informasi berupa passport atau ID juga dapat disertakan sebagai *watermark* ke dalam foto orang yang bersangkutan, sehingga jika suatu saat dokumen seperti passport dimanipulasi oleh orang lain dengan mengganti fotonya maka dapat dideteksi. [5]

Secara umum, teknik *watermarking* yang baik harus memenuhi kriteria sebagai berikut:

1. ***Imperceptibility***. Secara kasat mata manusia, antara media asli dan media yang sudah disisipi *watermark* harus tidak dapat dibedakan. [11]

2. ***Trustworthiness***. *Watermark* harus dapat menjamin kepemilikan asli dari media tersebut, artinya *watermark* harus sulit untuk dipalsukan. [11]
3. ***Robustness***. *Watermark* yang dihasilkan harus tangguh dan tahan terhadap perubahan yang terjadi pada media. [11]

2.2.1 Jenis-jenis *Digital Watermarking*

Berikut ini teknik-teknik *watermarking* untuk jenis media *text*, *image*, audio dan *video*.

1. Text Watermarking

Proses *watermarking* terhadap dokumen teks sebenarnya telah dilakukan di dalam kehidupan sehari-hari, misalnya dengan mencetak dokumen teks pada media khusus seperti kertas segel. Namun untuk melakukan *watermarking* pada teks yang tersimpan dalam dokumen *digital*, teknik yang dilakukan tidaklah sama. Berikut ini beberapa teknik *watermarking* terhadap teks. [11]

a. Line Shift Coding Protocol
b. Word Shift Coding Protocol
c. White Space Manipulation
d. *Text Content*

2. Image Watermarking

Watermarking terhadap gambar (*image*) paling banyak dilakukan untuk melindungi gambar seperti foto. Saat ini cukup banyak teknik maupun algoritma *watermarking* terhadap gambar yang ditawarkan. Beberapa diantaranya sebagai berikut: [11]

a. Simple Watermarking
b. Least Significant Bit Hiding (Image Hiding)
c. Hue Saturation Lightness (HSL)
d. Discrete Wavelet Transformation (DWT)
e. Spread Spectrum Watermarking

3. Audio Watermarking

Watermarking audio dapat diartikan sebagai suatu teknik penyembunyian data atau informasi rahasia ke dalam suatu data audio untuk "ditumpangi" (audio *host*), tetapi orang lain tidak menyadari keberadaan data tambahan pada data *host*-nya. Jadi seolah-olah tidak ada perbedaan antara audio *host* sebelum dan sesudah proses *watermarking*. Metode *watermarking* dikembangkan sebagai salah satu jawaban untuk menentukan keabsahan pencipta atau pendistribusi suatu data digital. [1]

2.2.2 Jenis-jenis *Audio Watermarking*

Audio watermarking mempunyai berbagai metode yang dapat diimplementasikan pada file audio berdasarkan domain frekuensi dan domain waktu.

1. Domain Frekuensi

Metode ini bekerja dengan cara mengubah *spectral content* dalam domain frekuensi dari sinyal. Misalnya dengan cara membuang komponen frekuensi tertentu atau menambahkan data sebagai derau dengan amplitudo rendah sehingga tidak terdengar.[1] Beberapa teknik yang termasuk dalam metode ini adalah:

a. *Phase Coding*
b. *Frequency Band Modification*
c. *Spread Spectrum*

2. Domain waktu

Metode ini bekerja dengan cara mengubah data audio dalam domain waktu yang akan disisipkan *watermark*. Contohnya dengan mengubah LSB (*Least Significant Bit*) dari data tersebut.[1] Beberapa teknik algoritma yang termasuk dalam metode ini adalah:

a. *Compressed-domain Watermarking*
b. *Bit Dithering*
c. *Echo Hiding*

2.3 *Echo Hiding*

Teknik ini merupakan salah satu metode pada *audio watermarking* yang bekerja sesuai domain waktu. Berikut penjelasan mengenai *Echo Hiding*.

2.3.1 Definisi *Echo Hiding*

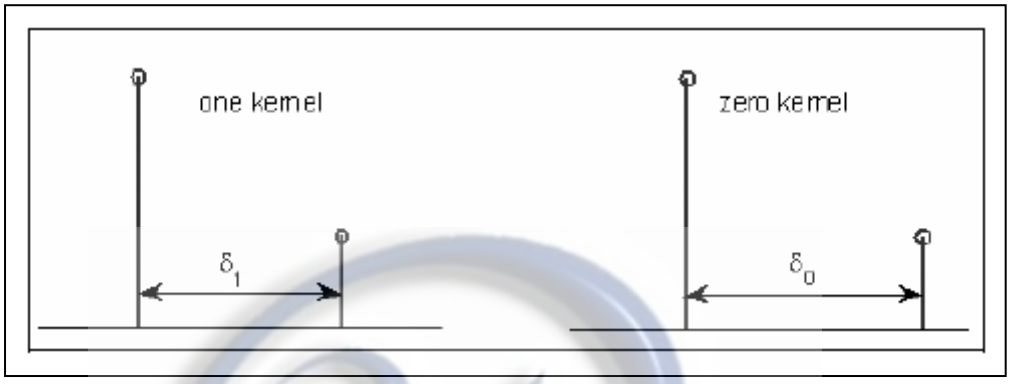
Metode *Echo Hiding* dilakukan dengan menambahkan data pada sinyal suara penampung dengan memunculkan *echo*. Data yang akan disembunyikan dalam bentuk *echo* dinyatakan dengan variasi dari tiga parameter, yaitu amplitudo awal, *decay rate*, dan *offset (delay)*. Amplitudo awal menyatakan amplitudo asal dari data suara tersebut, *decay rate* menyatakan seberapa besar *echo* yang akan diciptakan, dan *offset* menyatakan jarak antara sinyal suara dengan *echo* dalam bentuk fasa sudut dalam persamaan analog. Jika *offset* dari sinyal asal dan *echo* berkurang, maka kedua sinyal akan bercampur. *Echo* ini akan terdengar sebagai resonansi. [8]

2.3.2 Cara Kerja Echo Hiding

Echo Hiding merupakan metode untuk menyembunyikan informasi di dalam *file* audio. Metode ini menggunakan *echo* yang ada di dalam *file* audio untuk mencoba menyembunyikan informasi. Pesan akan disembunyikan dengan memvariasikan tiga parameter dalam *echo* yaitu besar amplitudo awal, tingkat penurunan atenuasi, dan *offset*. Ketiga parameter tersebut diatur sedemikian rupa di bawah pendengaran manusia sehingga tidak mudah untuk dideteksi. Sebagai tambahan, *offset* divariasikan untuk merepresentasikan *binary* pesan yang disembunyikan. Nilai *offset* pertama merepresentasikan nilai *binary* 1 dan nilai *offset* kedua merepresentasikan *binary* 0. [8]

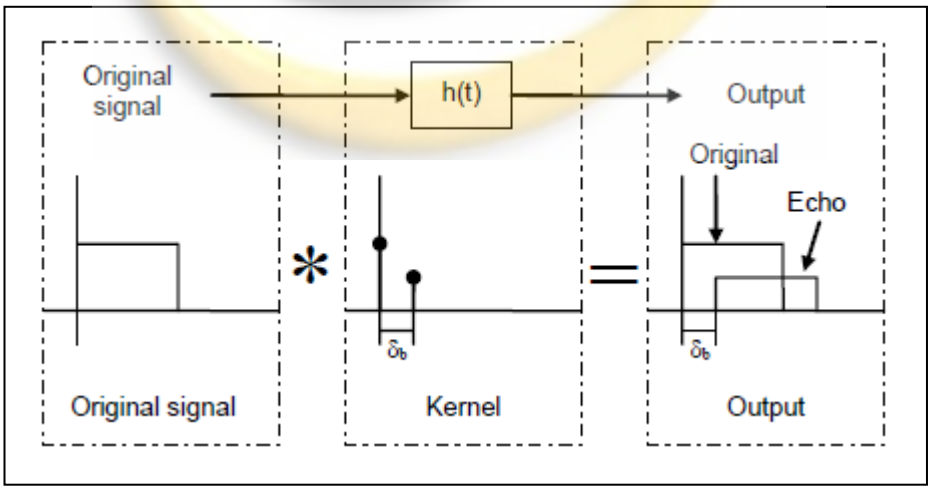
Echo Hiding menempatkan informasi sisipan pada sinyal asli (yang selanjutnya disebut *cover audio*) dengan menggunakan sebuah "*echo*." Pada kasus-kasus tertentu, telinga manusia tidak dapat mendengar sinyal asli dan *echo* secara bersamaan, melainkan hanya berupa sinyal distorsi tunggal. Hal ini sulit ditentukan secara tepat, ini tergantung pada kualitas rekaman sinyal asli, tipe suara yang di-*echo* dan pendengar. Fungsi sistem yang digunakan pada domain

waktu adalah *discrete time exponential* yang cara membedakannya hanya pada *delay* antar impuls. Untuk membentuk *echo* hanya menggunakan dua buah impuls yang disebut kernel. Kernel "satu" dibuat dengan delay δ1 detik sedangkan kernel "nol" dibuat dengan delay δ0 detik. [8]



Gambar 2.3 Kernel.[8]

Jika hanya 1 *echo* yang dihasilkan dari sinyal asli, hanya 1 bit informasi yang dapat di *encoding*. Karena itu, sinyal awal dibagi – bagi ke dalam beberapa blok sebelum proses *encoding* dimulai. Ketika proses *encoding* telah selesai, blok – blok tersebut digabungkan kembali membentuk sinyal baru. [8]



Gambar 2.4 Proses Pembentukan *Echo*.[8]

Kemudian algoritma yang digunakan untuk mengenkripsi setiap blok adalah sebagai berikut :

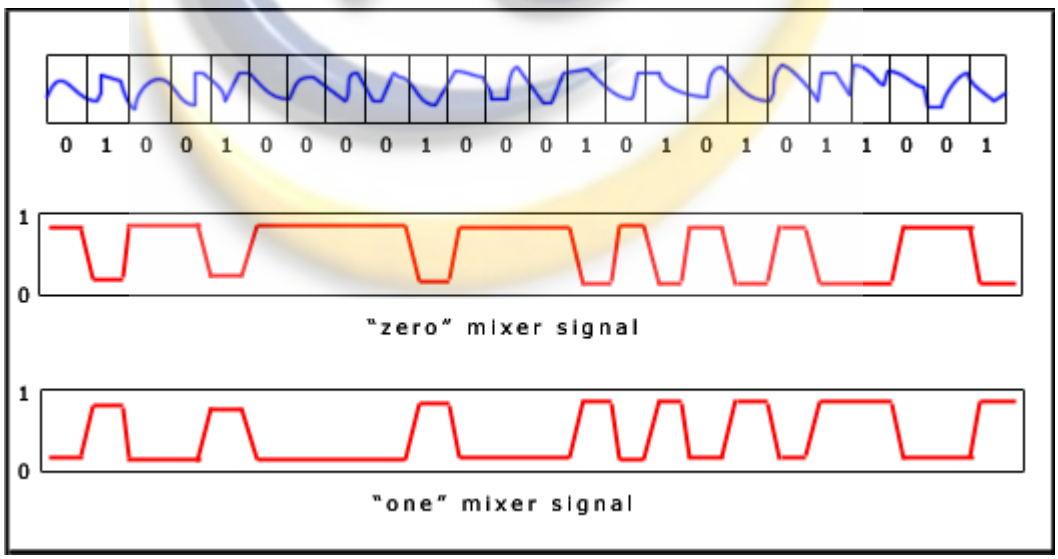
```
init(Block blocks[]) {
  for (int i=0; i < blocks.length; i++) {
    if (blocks[i].echoValue() == 0)
      blocks[i] = offset0(blocks[i]);
    else
      blocks[i] = offset1(blocks[i]);
  }
}

Block offset0(Block block) {
  return (block + (block - OFFSET_0));
}

Block offset1(Block block) {
  return (block + (block - OFFSET_1));
}
```

Gambar 2.5 Algoritma Pembentukan *Echo*.[10]

Blok – blok tersebut dikombinasikan untuk menghasilkan sinyal baru.



Gambar 2.6 Contoh Blok Sinyal dan Sinyal *Mixer*.[10]

Sinyal *echo* "1" kemudian dikali dengan sinyal *mixer* "1" dan sinyal *echo* "0" dikali dengan sinyal *mixer* "0". Kemudian kedua hasil tersebut dijumlahkan untuk mendapatkan sinyal akhir. Dengan adanya *offset* dari *echo* dan sinyal asli maka *echo* akan tercampur dengan sinyal aslinya.[10]

2.4 Jenis-Jenis File Audio

Secara umum, ada 3 kelompok utama format file audio, yaitu:

1. Format file audio tanpa kompresi, seperti file WAV, AIFF, AU dan raw header-less PCM.
2. Format file audio dengan kompresi *lossy*, seperti MP3, Vorbis, Mousepack, AAC, ATRAC, dan WMA *Lossy*.
3. Format file audio dengan kompresi *lossless*, seperti FLAC, MPEG-4 SLS, MPEG-4 ALS, MPEG-4 DST, WMA *Lossless*.

2.5 WAV (WAVeform Audio Format)

WAV merupakan file audio yang tidak terkompresi dan berukuran besar. File ini berbasis track audio, sehingga *encoder* dalam menghasilkan file WAV tidak ditentukan oleh kekuatan streaming *processor/mcu/cpu, bit-rate* yang dihasilkan tergantung *bit-rate* modul *sound card* dan *storage* yang digunakan.[6]

File WAV bisa disebut file universal, karena bisa dikompresi ke beberapa jenis file audio. Karena data-data digit kode audio didalam file WAV sangat lengkap secara menyeluruh, sehingga *encoder* file lebih mudah menerjemahkan algoritma data audio ke format lainnya. [6]

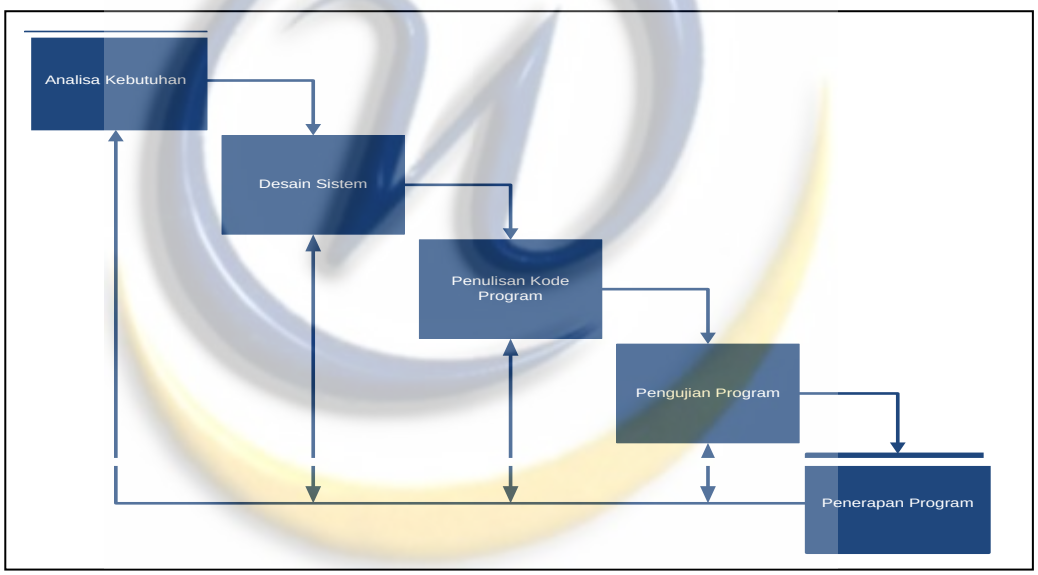
WAV adalah standar audio yang dikembangkan oleh Microsoft dan IBM sebagai standar untuk menyimpan file audio pada PC. WAV ini adalah format utama untuk menyimpan data audio mentah pada Windows dan menggunakan metode yang sama dengan AIFF Apple untuk menyimpan data. File WAV adalah file audio yang tidak terkompres sehingga seluruh sample audio disimpan semuanya di media penyimpanan dalam bentuk digital. Karena ukurannya yang besar, file WAV jarang digunakan sebagai file audio di Internet. [9]

WAV menggunakan teknik *pulse-code modulation* (PCM) yang tidak dikompres. Dengan cara ini , detil tidak hilang ketika audio analog didigitalkan dan disimpan. Ini membuat format WAV (menggunakan PCM) menjadi pilihan untuk mengedit audio *high-fidelity*. Akan tetapi untuk keperluan mengoleksi musik, transfer via internet dan,memainkan di-*player portable*, format ini kurang

popular dibandingkan dengan MP3, Ogg Vorbis dan VMA yang dikarenakan ukuran file yang sangat besar. [9]

2.6 Pembangunan Sistem dengan *Waterfall*

Dalam pembangunan sistem ini digunakan metode pengembangan *waterfall*. Inti dari metode *waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Jadi jika langkah satu belum dikerjakan maka tidak akan bisa melakukan pengerjaan langkah 2, 3 dan seterusnya. Secara otomatis tahapan ke-3 akan bisa dilakukan jika tahap ke-1 dan ke-2 sudah dilakukan.[2]



Gambar 2.7 Metode *Waterfall* [2]

Secara garis besar metode *waterfall* mempunyai langkah-langkah sebagai berikut :

1. Analisa

   Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau studi literatur. Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan sistem. [2]

2. *Design*

Proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada : struktur data, arsitektur perangkat lunak, representasi *interface*, dan *detail* (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*.[2]

3. *Coding*

Coding merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh *programmer* yang akan meterjemahkan transaksi yang diminta oleh *user*. Tahapan ini lah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini.[2]

4. *Testing*

Tahapan ini dilakukan setelah proses pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.[2]

5. Penerapan

Tahapan ini bisa dikatakan *final* dalam pembuatan sebuah sistem. Setelah melakukan analisa, *design* dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*.[2]

2.7 *Matlab* (*Matlab Laboratory*)

Matlab merupakan bahasa pemrograman yang hadir dengan fungsi dan karakteristik yang berbeda dengan bahasa pemrograman lain yang sudah ada lebih dahulu seperti Delphi, Basic, maupun C/C++. *Matlab* memiliki kemampuan mengintegrasikan komputasi, visualisasi, dan pemrograman. Oleh karenanya, *Matlab* banyak digunakan dalam bidang riset-riset yang memerlukan komputasi numerik yang kompleks. [4]

Penggunaan *Matlab* meliputi bidang-bidang :

a. Matematika dan Komputasi

b. Pembentukan Algorithma

c. Akusisi data

d. Pemodelan, Simulasi, dan Pembuatan Prototype

e. Analisa Data, Explorasi, dan Visualisasi

f. Grafik Keilmuan dan Rekayasa

Matlab merupakan kepanjangan dari *Matlab Laboratory*. Sesuai dengan namanya, struktur data yang terdapat dalam *Matlab* menggunakan matriks atau array berdimensi dua (*double*). *Matlab* membawa keistimewaan dalam fungsi-fungsi matematika, fisika, statistik, dan visualisasi. [4]

2.7.1 Kelengkapan Pada Sistem *Matlab*

Sebagai sebuah sistem, *Matlab* tersusun dari 5 bagian utama, yaitu :

1. *Development Environment*

Merupakan sekumpulan perangkat dan fasilitas yang membantu untuk menggunakan fungsi-fungsi dan file-file *Matlab.* Beberapa perangkat ini merupakan sebuah *graphical user interface* (GUI). [4]

2. *Matlab Mathematical Function Library*

Merupakan sekumpulan algoritma komputasi mulai dari fungsi-fungsi dasar seperti : sum, sin, cos, *complex arithmatic*, sampai dengan fungsi-fungsi yang lebih kompleks, seperti *matrix inverse, matrix eigencalues, Bessel Functions, dan Fast Fourier Transforms*. [4]

3. *Matlab Language*

Merupakan suatu *high-level matrix/array language* dengan *control flow statements, functions, data structures, input/output*, dan fitur-fitur *object oriented programming*.[4]

4. *Graphics*

Matlab memiliki fasilitas untuk menampilkan *vector* dan *matrices* sebagai suatu grafik. Didalamnya melibatkan *high-level functions* (fungsi-fungsi level tinggi) untuk visualisasi data dua dimensi dan data tiga dimensi, *image processing*, *animation*, dan *presentation graphics*.[4]

5. *Matlab Application Program Interface (API)*

Merupakan suatu *library* yang memungkinkan program yang telah anda tulis dalam bahasa C dan Fortran mampu berinteraksi dengan Matlab.[4]