

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Berikut ini adalah beberapa penelitian terdahulu yang berkaitan dengan topik peneliti, dan menjadi referensi yang digunakan untuk menentukan metode yang akan digunakan nantinya adalah sebagai berikut :

1. Penelitian yang dilakukan oleh M. Azlansyah (2019) dengan judul “Penyisipan Pesan Pada Citra Digital Menggunakan *Metode Least Significant Bit*” bertujuan untuk melakukan penyisipan pesan pada citra digital yang memiliki tipe *bmp*, *jpeg*, *png*, dan *tiff* dengan metode LSB. Hasil yang diperoleh pada penelitian ini menunjukkan bahwa citra yang telah disisipkan pesan tidak berubah secara kasat mata dan pesan yang disisipkan dapat diekstraksi dan dikembalikan ke keadaan semula [1].
2. Penelitian yang dilakukan oleh G.H. Irawan (2019) dengan judul “Steganografi Foto dalam Citra dengan Metode LSB dan Kompresi Metode *Huffman*” bertujuan untuk melakukan perbandingan pada citra yang telah disisipkan pesan menggunakan metode LSB tanpa dilakukan kompresi dengan citra yang telah disisipkan pesan dan telah dilakukan kompresi [3].
3. Penelitian yang dilakukan oleh Anshori (2019) dengan judul “Aplikasi Steganografi pada Media Citra Digital Menggunakan Metode *Least Significant Bit (LSB)*” bertujuan untuk melakukan penyisipan pesan gambar, pesan teks, dan dokumen (*rar*) ke dalam citra *bitmap* 24 bit [2]. Penelitian ini menghasilkan aplikasi yang dapat melakukan penyembunyian pesan yang dapat diekstrak kembali tanpa mengurangi isi pesan.
4. Penelitian yang dilakukan oleh J.I. Sari (2017) dengan judul “Implementasi Penyembunyian Pesan pada Citra Digital dengan Menggabungkan Algoritma *Hill Cipher* dan Metode *Least Significant Bit (LSB)*” bertujuan untuk membuat aplikasi yang dapat menjaga serta memberikan pertahanan keamanan yang berlapis dengan tidak mengurangi atau merusak pesan teks pada citra digital yang akan dihasilkan [8].

5. Penelitian yang dilakukan oleh Widagda (2017) dengan judul “Pendeteksian Target Wajah Manusia dengan Metode *Viola-Jones*” bertujuan untuk mendeteksi target wajah dan menentukan area wajah dari objek yang ada pada sebuah citra.

Tabel 2.1 Tinjauan Studi

No	Nama Peneliti	Tahun	Metode yang digunakan	Hasil
1.	M. Azlansyah, Budi Setiyono	2019	LSB	Citra yang disisipkan pesan tidak berubah secara signifikan serta pesan yang disisipkan dapat diekstraksi kembali secara utuh tanpa perubahan apapun.
2.	G.H. Irawan, Nabil Ramdhani	2019	LSB dan Kompresi Metode <i>Huffman</i>	Penggunaan kompresi pada citra yang akan disisipkan dapat menurunkan dengan cukup signifikan ukuran citra hasil (citra stego) dan juga citra hasil ekstraksi
3.	Yusuf Anshori, A.Y. Erwin Dodu, Megawati Purwaningsih	2019	LSB	Metode LSB berhasil menyembunyikan pesan berupa teks, citra, atau dokumen (<i>rar</i>) ke dalam citra <i>bitmap</i> .
4.	J.I. Sari, Sulindawaty, H.T. Sihotang	2017	LSB dan Algoritma Hill <i>Cipher</i>	Metode LSB berhasil digunakan untuk menyisipkan pesan

No	Nama Peneliti	Tahun	Metode yang digunakan	Hasil
				pada citra digital. Penggunaan algoritma <i>Hill Cipher</i> berhasil untuk pengujian dan menghasilkan <i>cipher teks</i> berupa huruf abjad teracak.
5.	I G.A. Widagda, I Ketut Sukarasa	2017	Metode <i>Viola Jones</i>	Metode <i>Viola-Jones</i> berhasil mendeteksi target wajah yang memiliki posisi pandangan lurus ke depan, namun program gagal mendeteksi target wajah dengan posisi lain seperti wajah menghadap sekitar 45^0 hingga 90^0 kesamping.

2.2 Landasan Teori

Pada sub bab ini menjelaskan tentang konsep yang terkait dengan penelitian.

2.2.1 Steganografi

Steganografi (*steganography*) berasal dari bahasa Yunani yaitu *Steganos* yang berarti tertutup/tersembunyi, dan *graphien* yang berarti menulis. Steganografi adalah seni untuk menyembunyikan keberadaan pesan dengan cara menanamkan (*embedding*) pesan ke dalam sebuah media digital agar orang lain dapat mengabaikan keberadaan pesan di dalam media tersebut [8]. Steganografi pada dasarnya memiliki fungsi yang sama dengan kriptografi (*cryptographic*) yaitu berperan untuk mengamankan suatu data atau pesan, tetapi keduanya memiliki maksud dan proses penyembunyian pesan yang berbeda. Kriptografi memiliki maksud untuk merahasiakan makna pesan atau informasi dengan mengacak data asli selama penyandian sehingga menghasilkan data terenkripsi yang sepenuhnya asli acak sedangkan steganografi menutupi keberadaan pesan dengan menyembunyikan pesan tersebut ke dalam media penampung dengan hasil sebelum dan setelah proses penyembunyiannya hampir sama.

Proses penyembunyian pesan dengan steganografi membutuhkan dua properti untuk prosesnya yaitu wadah penampung dan pesan yang akan disembunyikan. Properti penampung disebut dengan *cover-object* yang dapat berupa citra, suara (*audio*), teks, atau citra bergerak (*video*). Sedangkan pesan yang akan disembunyikan disebut *embedded message* juga dapat berupa citra, suara, teks atau video. Proses steganografi yang dilakukan dengan media gambar sebagai *cover object* nantinya akan menghasilkan citra stego yaitu gambar yang telah disisipkan pesan. Penggunaan steganografi dalam penyembunyian pesan memungkinkan pengiriman pesan tanpa diketahui karena media yang telah disisipkan pesan terlihat sama sebelum dan sesudah pemrosesan. Hal ini menjadi keuntungan karena pihak ketiga tidak menyadari keberadaan pesan [9].

Dalam menyembunyikan pesan dengan steganografi ini memiliki kriteria yang harus dipenuhi, yaitu [1], [10] :

1. *Fidelity*; mutu dari *cover-object* tidak berubah banyak akibat penyisipan *embedded message*.
2. *Imperceptibility*; keberadaan *embedded message* tidak dapat dipersepsi oleh indra.
3. *Recovery*; pesan yang disembunyikan harus dapat diungkap kembali (*reveal*).

2.2.1.1 Metode-Metode Steganografi

Ada beberapa teknik/metode steganografi yang dapat digunakan dalam penyembunyian pesan, diantaranya adalah [11] :

1. *Least Significant Bit Insertion (LSB)*

LSB adalah salah satu metode yang dapat digunakan dalam menyembunyikan pesan pada steganografi. LSB menjadi metode ini sering digunakan oleh para peneliti pada proses penggunaan steganografi karena metode steganografi yang cukup sederhana. LSB menyisipkan sebuah pesan rahasia dengan menambahkan bit data pesan pada bagian dari data biner paling cocok atau yang mempunyai nilai kurang berarti yang dapat dilakukan secara sekuensial maupun secara acak [1], [9].

2. *Masking and filtering*

Metode ini umumnya terbatas pada citra warna 24-bit dan citra skala abu-abu atau *grayscale*. Untuk menyembunyikan pesan rahasia suatu citra diberi tanda (*marking*). Metode ini memiliki batasan kapasitas tertentu untuk pesan yang akan digunakan dalam proses penyembunyian.

3. *Algorithm and Transformation*

Metode ini lebih kompleks dari metode sebelumnya sehingga lebih sulit dalam penerapan untuk menyembunyikan pesan lebih tinggi dari metode yang lain, karena metode ini lebih kompleks dari metode-metode sebelumnya. Metode ini bekerja pada domain frekuensi file digital

menggunakan *Discrete Cosine Transformation* (DCT) dan *Wavelet Compression*.

4. *Spread Spectrum Methode*

Pada metode ini, proses menyembunyikan suatu pesan dilakukan dengan mengkodekan menyebarkan pesan rahasia yang telah dienkripsi di semua spektrum frekuensi yang memungkinkan. Namun, kelemahan dari metode ini yaitu masih mudah diserang dengan cara penghancuran atau pengrusakan gambar dari proses maupun kompresi citra.

2.2.2 Konsep Kerja LSB

Least Significant Bit adalah bagian dari barisan data biner yang letaknya paling kanan dari barisan bit dan mempunyai nilai paling tidak berarti/paling kecil. Sebaliknya, *Most Significant Bit* adalah bagian dari barisan data biner yang letaknya paling kiri dan mempunyai nilai paling berarti/paling besar [9].

0	1	0	0	0	0	0	1
---	---	---	---	---	---	---	---

Keterangan :

	<i>Most Significant Bit</i>
	<i>Least Significant Bit</i>

Pada metode LSB modifikasi dilakukan pada bit-bit LSB pada biner piksel citra dengan mengubah bit-bit tersebut dengan bit-bit dari pesan yang akan disembunyikan. Bit LSB adalah bit yang cocok digantikan dengan bit-bit pesan karena perubahan pada bit ini hanya mengubah satu lebih tinggi atau rendah dari nilai biner piksel sebelumnya. Proses penyembunyian pesan ini disebut dengan proses *embedding/encoding* yang akan menghasilkan citra stego. Apabila ingin mendapatkan kembali informasi yang telah disembunyikan maka bit-bit LSB pada citra stego akan di ekstrak dengan diambil satu per satu lalu disatukan kembali sehingga menghasilkan pesan yang tersembunyi, proses ini disebut dengan *decoding/retrieving*.

Penyisipan pesan disisipkan secara berurutan dari data titik pertama piksel citra yang ditentukan dari kiri ke kanan baris per baris. Berikut adalah contoh penggantian bit-bit LSB pada steganografi dengan cara sekuensial. Misalkan pesan akan disembunyikan pada citra warna RGB yang memiliki 3 layer warna, yaitu *Red*, *Green*, dan *Blue*. Pesan akan disembunyikan pada gambar yang memiliki ukuran 3 piksel :

11001011	10011001	01100101	Layer <i>Red</i>
11110110	11101101	11010100	Layer <i>Green</i>
11111111	11111110	11111111	Layer <i>Blue</i>
Piksel 1	Piksel 2	Piksel 3	

Pesan yang akan disisipkan adalah karakter 'Z' yang memiliki biner 01011010, maka citra stego yang dihasilkan akan memiliki biner piksel sebagai berikut :

1100101 0	1001100 1	0110010 0
1111011 1	1110110 1	1101010 1
1111111 0	1111111 0	1111111 1

Untuk menghitung ukuran pesan yang dapat disisipkan bergantung pada ukuran citra *cover*. Namun pada penelitian ini ukuran pesan yang dapat disisipkan bergantung pada ukuran citra *cover* baru yang didapat dari area wajah yang telah terdeteksi. Pada citra berwarna 24-bit dengan ukuran area wajah yang terdeteksi 256 x 256 piksel ukuran pesan yang dapat disisipkan dapat dihitung dengan :

- Jumlah piksel $256 \times 256 = 65536$
- Setiap piksel = 3 *byte* (karena terdapat 3 layer warna yaitu *red*, *green*, *blue*), berarti ada $65535 \times 3 = 196608$ *byte*.
- Setiap *byte* dapat menyembunyikan 1 bit pesan
- Jadi, ukuran maksimal pesan = 196608 bit = 24576 *byte*

2.2.3 Citra Digital

Citra digital adalah file yang dapat diolah oleh komputer yang dibuat dari sebuah media digital seperti kamera atau perangkat elektronik lainnya. Sebuah citra digital merupakan suatu matriks yang terdiri dari kumpulan piksel dimana indeks dari baris dan kolomnya menyatakan titik tertentu pada citra dan elemen matriksnya (piksel) menyatakan tingkat keabuan dari titik tersebut [1], [3]. Citra digital dapat dibagi menjadi dua kategori : *raster* dan *vektor*. Pada umumnya, yang biasa disebut dengan citra digital adalah citra digital berformat raster yang biasa disebut dengan citra *bitmap* [9]. Manipulasi yang dilakukan terhadap file *bitmap* tidak dapat diubah atau dikonvert ke dalam bentuk atau format yang lain karena data yang tersembunyi dalam file *bitmap* akan hilang [8].

2.2.3.1 Tipe Citra Digital

Ada beberapa tipe citra digital yang sering digunakan untuk proses steganografi. Pada penelitian ini akan digunakan tipe citra digital berikut [3]:

1. *Bitmap* (BMP)

Citra dengan format file *bitmap* (*.bmp) biasa digunakan untuk proses steganografi karena file ini dibentuk oleh bit-bit penyusun yang bersifat independen dapat diubah serta dimodifikasi dan memiliki ukuran lebih besar dibandingkan lainnya. Citra dengan tipe *bitmap* ini juga dapat dibuka dan digunakan pada hampir semua aplikasi yang biasa digunakan untuk melakukan pengolahan citra. *Bitmap* merupakan representasi atau gambaran dari citra yang terdiri dari baris dan kolom pada titik citra grafis di komputer. Struktur *bitmap* berbeda dari struktur citra dengan format lain, yaitu terdiri dari *header*, *info header*, dan *color tabel*.

Header bitmap merupakan bagian dari file *bitmap* yang berisi informasi *header* untuk file citra. *Header* ini memiliki ukuran 14 *bytes* yang terdiri dari 2 *signature bytes* "BM" yang menjadi tanda citra memiliki format bmp, 4 *byte filesize* (ukuran gambar dalam satuan *bytes*), 4 *bytes reserved* yang tidak digunakan bernilai 0, dan 4 *bytes data offset* untuk raster data.

Info Header adalah bagian dari *header* yang berisi data file citra *bitmap*. Letaknya berada dibelakang atau setelah *header*. *Info header* berukuran 40 *byte* dengan 4 *bytes* tinggi citra, 4 *bytes* lebar citra masing-masing dalam satuan piksel, 2 *bytes* planes jumlah warna yang isinya selalu 1, 2 *bytes* *BitCount* jika bernilai 8 maka banyak warna adalah 256 dengan 8 bit *pallette*, 4 *bytes* *compression*, 4 *bytes* *ImageSize* yang didapat dari perkalian panjang dan lebar ukuran citra dalam *bytes*, 4 *bytes* resolusi horizontal, 4 *bytes* resolusi vertikal masing-masing dalam satuan piksel, 4 *byte* *colorused*, 4 *bytes* *colorImportant* yang merupakan banyaknya warna utama dalam color tabel.

Color table berisi warna-warna yang terdapat pada citra *bitmap*. Ukurannya 4 kali lipat dari warnanya. *Color table* berisi *red-green-blue (RGB)*. Strukturnya terdiri dari masing-masing 1 *bytes* untuk intensitas warna *rgbblue*, *rgbgreen*, dan *rgbred*, serta 1 *bytes* yang diset sama dengan 0 yaitu *rgbReserved*.

2. *Joint Photographic Experts Group (JPG/JPEG)*

Citra dengan format file ini bersifat kompresi dan *lossy* (menghilangkan) yang artinya bahwa jika menyimpan citra dari tipe lain menjadi tipe JPG/JPEG ada kemungkinan terjadi penghilangan beberapa bit dari citra semula yang akan berakibat kualitas maupun ukuran gambarnya dapat mengecil dan tidak dapat dikembalikan.

3. *Portable Network Graphics (PNG)*

Citra dengan format file PNG dapat mengandung data warna 24 bit RGB + alpha. Nilai alpha menentukan tingkat ketransparanan warna, nilai alpha 0 akan membuat piksel warna transparan dan nilai alpha 1 akan membuat piksel warna pekat. Tipe ini merupakan *solution* kompresi dimana kompresi yang digunakan pada tipe ini tidak menghilangkan data.

2.2.3.3 Jenis Citra Digital

Jenis citra digital ini berkaitan erat dengan warna karena setiap nilai data digital mewakili warna citra tersebut. Beberapa format citra digital yang termasuk umum untuk digunakan diantaranya adalah citra biner (*monochrome*), citra skala keabuan (*grayscale*) dan citra warna (*true color*).

1. Citra Biner (*monochrome*)

Citra biner, juga disebut *binary image* merupakan citra dengan 2 buah warna, yaitu hitam dan putih. Setiap piksel dalam citra biner hanya memiliki dua kemungkinan tingkat keabuan, Hitam direpresentasikan dengan nilai 0 dan putih direpresentasikan dengan nilai 1. Setiap piksel yang akan disimpan membutuhkan 1 bit pada media penyimpanannya [12].

2. Citra Skala Keabuan (*grayscale*)

Citra skala keabuan menawarkan lebih banyak kemungkinan warna daripada citra biner. Karena memiliki potensi hitam minimum dan maksimum putih, sehingga ada warna abu-abu diantara kemungkinan warna tersebut. Ukuran bit media penyimpanan akan mempengaruhi banyaknya warna yang akan disimpan sehingga semakin banyak jumlah bit yang disediakan maka gradasi warna yang terbentuk akan semakin halus [12].

3. Citra Warna (*true colour*)

Citra warna juga dikenal sebagai citra RGB (*Red Green Blue*). Hal ini karena setiap piksel dalam citra berwarna merupakan kombinasi dari tiga warna primer yang dapat diterima oleh mata manusia yaitu warna *Red* (Merah), *Green* (Hijau), dan *Blue* (Biru). Setiap titik atau piksel pada citra RGB membutuhkan media penyimpanan sebesar 3 *byte*. Setiap komponen warna memiliki intensitas sendiri dari nilai minimal nol (0) hingga nilai maksimal 255 (8 bit) [1].

2.2.5 Mean Square Error (MSE)

MSE merupakan alat untuk mengukur keberhasilan kinerja peningkatan pada citra. Semakin kecil *MSE*, semakin baik kinerja reduksi citra yang dapat diukur secara kuantitatif [13].

$$MSE = \frac{1}{M \times N} \sum_{x=1}^M \sum_{y=1}^N (S_{xy} - C_{xy})^2 \quad (1)$$

Keterangan :

M = Panjang citra

N = Lebar citra

S_{xy} = Citra stego (citra yang telah disisipi)

C_{xy} = Citra *cover*

x dan y = titik koordinat pada citra

2.2.6 Peak Signal to Noise Ratio (PSNR)

Peak Signal to Noise Ratio adalah perbandingan antara nilai maksimum dari sinyal yang diukur dalam satuan desibel (dB) dengan besarnya derau yang mempengaruhi sinyal tersebut. *PSNR* adalah penilaian kualitas yang paling umum digunakan untuk mengukur kualitas rekonstruksi *lossy image compression codecs*[14].

PSNR dinyatakan sebagai berikut :

$$PSNR = 10 \log \frac{MAX^2}{MSE} = 20 \log \frac{MAX}{\sqrt{MSE}} \quad (2)$$

Nilai MAX adalah *peak value* yang merupakan maksimal dalam data gambar. MAX bernilai 255 jika itu adalah 8-bit tipe data bilangan bulat yang tidak ditandai. Kualitas suatu citra dinyatakan baik apabila nilai *PSNR* semakin besar dan nilai *MSE* semakin kecil. Nilai *PSNR* bervariasi dari 30 hingga 50 dB untuk representasi data 8-bit dan dari 60 hingga 80 dB untuk 16-bit data dan seterusnya[14].

2.2.7 Mean Opinion Score (MOS)

MOS (*Mean Opinion Score*) yang merupakan metode yang dapat digunakan untuk mengukur kinerja dan kualitas dari sistem yang dibangun. Pada penelitian ini MOS digunakan sebagai parameter subjektif untuk membandingkan objek sebelum dan sesudah disisipin informasi. Parameter MOS dapat dilihat pada tabel di bawah ini.

Tabel 2.2 Parameter Penilaian MOS

Nilai MOS	Kualitas Gambar	Keterangan
5	Sangat Baik	Kemiripan Gambar (90% - 100%)
4	Baik	Kemiripan Gambar (80% - 90%)
3	Sedang	Kemiripan Gambar (60% - 70%)
2	Buruk	Kemiripan Gambar (40% - 69%)
1	Sangat Buruk	Kemiripan Gambar (<40%)

Pada **Tabel 2.2** parameter penilaian MOS memiliki skala dari 1-5 dimana nilai 5 sangat baik dan nilai 1 sangat buruk. Untuk menentukan nilai MOS dari hasil sampel kuisisioner dapat dihitung sebagai berikut [15] :

$$MOS = \frac{\sum(\text{Nilai MOS} + \text{Total Responden})}{\sum \text{Nilai MOS}} \quad (3)$$

2.2.8 Pendeteksian Area Wajah

Proses pendeteksian wajah merupakan tahapan dasar yang harus dilakukan sebelum melakukan proses pengenalan (*recognition*) sebuah wajah pada gambar. Proses deteksi wajah melakukan klasifikasi pola pada sebuah citra yang akan menghasilkan *output* berupa label kelas dari citra yang telah diinputkan yaitu label wajah dan non-wajah.

Dalam proses pendeteksian wajah biasanya ditemui masalah yang disebabkan oleh faktor-faktor berikut ini [16]:

1. Terhalang objek lain. Biasanya pada sebuah citra yang terdapat banyak objek baik itu objek lain maupun sekelompok orang, area wajah dapat terhalangi oleh objek-objek lain yang ada pada citra.

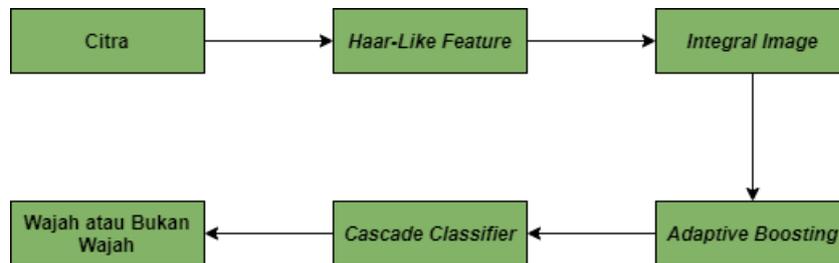
2. Posisi wajah. Di dalam sebuah citra dapat menampilkan posisi wajah yang bervariasi tergantung dengan posisi pengambilan dan pose wajah tersebut. Posisi wajah yang terdapat pada sebuah citra dapat tegak, menoleh, miring, ataupun tampak samping.
3. Ekspresi wajah. Ekspresi wajah dapat mempengaruhi penampilan wajah pada citra misalnya tersenyum, sedih, berbicara, tertawa, dan sebagainya.
4. Komponen-komponen yang ada pada wajah misalnya kacamata, kumis, dan jenggot.
5. Kondisi pengambilan citra. Kondisi pencahayaan dapat menjadi pengaruh pada citra seperti arah sumber cahaya, intensitas cahaya, serta karakteristik dan lensa kamera yang digunakan.

Metode deteksi wajah digunakan untuk mencari lokasi serta ukuran objek pada citra.

2.2.9 Metode Viola-Jones

Metode pengenalan area wajah yang akan digunakan dalam penelitian ini adalah *Viola-Jones*. Pendeteksian wajah pada proses pengenalan wajahnya menggunakan metode *Viola-Jones* yaitu dengan melakukan mengklasifikasikan objek dalam suatu citra menurut nilai fitur atau karakteristik sederhana. Metode *Viola-Jones* dipilih karena dapat melakukan pendeteksian yang mampu memberikan hasil dengan tingkat keakuratan dan kecepatan yang tinggi [7]. Dalam proses penggunaan algoritma *Viola-Jones* untuk pendeteksian wajah, beberapa proses perlu dilakukan untuk akhirnya menghasilkan keluaran wajah yang terdeteksi pada citra. Pada deteksi wajah *Viola-Jones*, proses-proses tersebut adalah *Haar-Like Feature*, *Integral Image*, *Adaboost (Adaptive Boosting)*, dan *Cascade Classifier*.

Proses setiap tahap yang dilalui citra untuk mendapatkan hasil pendeteksian wajah ditunjukkan pada **Gambar 2.1**.

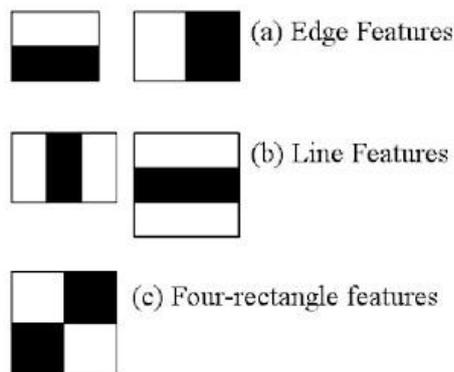


Gambar 2.1 Skema Deteksi *Viola-Jones*

Detail setiap tahap yang dilalui oleh citra pada proses pendeteksian wajah pada **Gambar 2.1** diatas adalah sebagai berikut :

2.2.5.1 *Haar-Like Feature*

Haar-Like Feature adalah fungsi fitur persegi yang memberikan indikasi sebuah citra atau *image* secara spesifik. Dalam metode *Viola-Jones*, terdapat tiga jenis fitur yaitu *edge features*, *line features*, *four-rectangle features* seperti pada **Gambar 2.2**.



Gambar 2.2 Jenis *Haar-Like Feature*

Untuk dapat mendeteksi ada tidaknya fitur wajah pada sebuah citra, proses yang dilakukan oleh metode *Viola-Jones* pertama adalah mengubah citra rgb menjadi grayscale. Contoh *Haar-Like Feature* dapat dilihat pada **Gambar 2.3**.



Gambar 2.3 Contoh Perubahan Citra RGB ke *Grayscale Haar-Like Feature*

Setelah mengubah citra rgb menjadi citra *grayscale*, langkah selanjutnya adalah menentukan fitur haar yang ada pada citra. Teknik yang digunakan adalah dengan mengkotak-kotakkan setiap area pada citra dari sudut kiri atas ke sudut kanan bawah untuk mengetahui apakah terdapat fitur wajah pada area tersebut. Dalam proses seleksi fitur haar, fitur-fitur tersebut digunakan untuk menemukan fitur wajah seperti mata, hidung, dan mulut. Tiap-tiap kotak fitur kemudian diproses dan didapatkan perbedaan atau selisih nilai ambang (*threshold*) yang menyatakan antara wilayah gelap (*black*) dan terang (*white*). Perhitungan nilai *Haar-Like Feature* didapatkan dari selisih antara jumlah nilai piksel pada daerah gelap (*black*) dengan jumlah piksel pada daerah terang (*white*) :

$$F(Haar) = \sum F White - \sum F Black \quad (4)$$

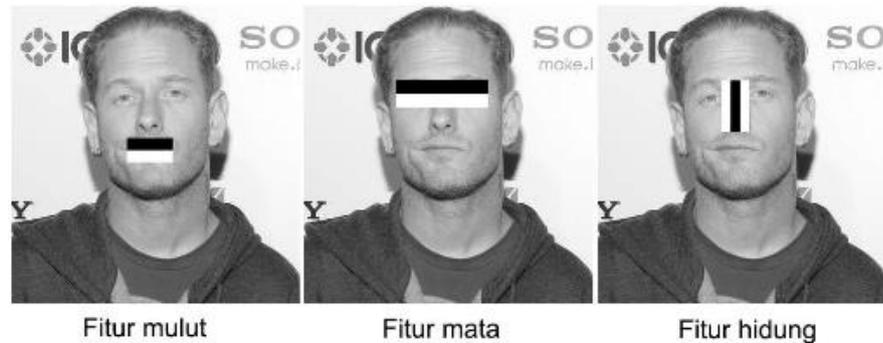
Keterangan :

$F(Haar)$ = Nilai fitur total

$\sum F White$ = Nilai fitur pada daerah terang

$\sum F Black$ = Nilai fitur pada daerah gelap

Untuk memilih fitur mata, hidung, dan mulut digunakan kotak-kotak fitur seperti yang ditunjukkan pada **Gambar 2.4** di bawah ini.

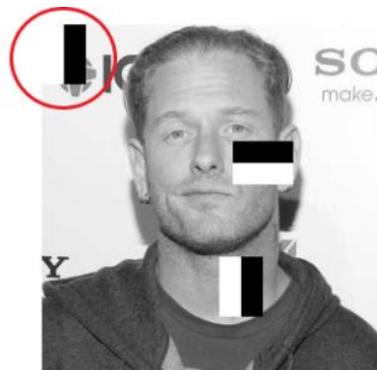


Gambar 2.4 Haar-Like Feature untuk Pemilihan Fitur Mata, Hidung, Mulut

Secara umum, citra pada area mata, hidung, mulut cenderung lebih gelap dari pada area pipi, dagu dan kening. Untuk mempermudah dan mempercepat proses penghitungan nilai haar suatu citra, maka algoritma *Viola-Jones* menggunakan perhitungan yang disebut *integral image*.

2.2.5.2 Citra Integral (*Integral Image*)

Citra Integral (*Integral Image*) adalah citra dimana setiap nilai pikselnya adalah berupa hasil penjumlahan nilai piksel dari kiri atas sampai piksel kiri bawah. Citra integral dapat digunakan supaya dapat memudahkan dalam proses penghitungan nilai fitur Haar pada setiap lokasi citra. Nilai integral dari setiap piksel adalah jumlah dari semua piksel sebelah atasnya dan di sebelah kirinya.



Gambar 2.5 Nilai Piksel Pada Sebuah Fitur

Nilai piksel-piksel fitur lingkaran merah pada **Gambar 2.5** adalah sebagai berikut:

1	2	3
4	5	6
7	8	9

Citra Masukan

1	3	6
5	12	21
12	27	45

Citra Integral

Sebagai contoh untuk menghitung nilai 12 maka perlu menghitung piksel pada bagian kiri dan atas citra masukan yaitu $1+2+4+5$. Begitupun pada kotak yang lainnya. Dengan pemakaian integral image maka kita dapat menentukan nilai piksel dari beberapa kotak atau segiempat (rectangular) yang lain. Hasil nilai integral dari citra awal dan nilai jumlah piksel pada area tertentu akan dibandingkan antara nilai piksel pada area terang dan gelap. Jika selisih antara nilai piksel di area terang dan nilai piksel di area gelap lebih besar dari ambang batas (*threshold*), maka area tersebut dinyatakan memiliki fitur.

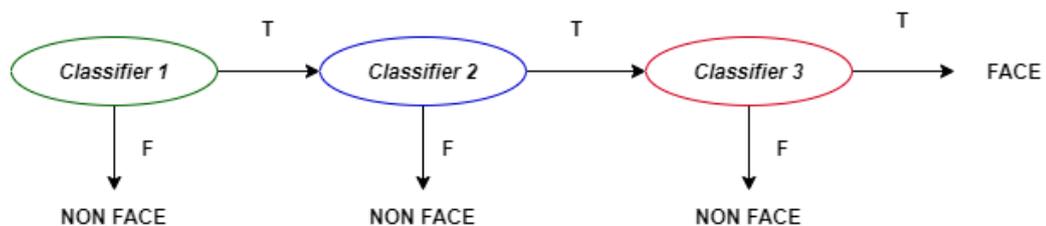
2.2.5.3 Adaptive Boosting

Pada metode *AdaBoost* beberapa *classifier* lemah akan digabungkan menjadi sebuah *classifier* kuat. Dalam hal ini lemah mengandung arti bahwa filter pada *classifier* menghasilkan jawaban benar lebih sedikit. Andaikan semua *classifier* lemah digabung maka hasilnya adalah *classifier* yang lebih kuat. Metode *AdaBoost* akan memilih beberapa *classifier* lemah untuk digabungkan dan pada tiap-tiap *classifier* tersebut ditambahkan bobot, dan hasilnya berupa *classifier* kuat. Beberapa *AdaBoost classifier* digabung menjadi sebuah rangkaian filter yang dapat mengelompokkan daerah image dengan efisien.

2.2.5.4 Cascade Classifier

Proses pengklasifikasian pada metode *Viola-Jones* berlangsung secara bertingkat. Proses pengklasifikasian pada algoritma *Viola-Jones* ini terdiri dari n tingkatan. Dimana pada setiap tingkatan mengeluarkan sub citra yang diyakini bukan objek (wajah). Hal ini dilakukan oleh karena akan lebih mudah menentukan sub citra tersebut bukan objek (wajah) daripada menentukan sub citra tersebut berupa objek

(wajah). *Cascade classifier* melakukan proses klasifikasi multi-level atau bertingkat. Ada tiga klasifikasi yang menentukan apakah subjek yang dipilih memiliki karakteristik wajah. Pada filter *classifier* pertama, setiap subcitra diklasifikasikan berdasarkan satu fitur. Apabila hasil dari nilai fitur dari filter belum memenuhi kriteria yang dipersyaratkan, maka hasilnya tidak diterima. Lalu selanjutnya melanjutkan bergerak ke *sub-window* selanjutnya, menghitung ulang nilai fitur, dan ketika hasilnya sesuai dengan ambang batas yang diinginkan, melanjutkan ke tahap filter berikutnya hingga jumlah *sub-window* yang lolos melewati *classifier* berkurang hingga mendekati citra yang dideteksi. Proses rangkaian filter yang dilewati oleh masing-masing *classifier* dapat dilihat pada **Gambar 2.6** di bawah ini.



Gambar 2.6 *Cascade Classifier*

Proses diatas dilakukan untuk memperoleh hasil deteksi setelah melakukan serangkaian proses seperti pemilihan fitur dan klasifikasi bertingkat. Hasil pendeteksian bisa berupa wajah ataupun bukan wajah.