

## **BAB 2**

### **LANDASAN TEORI**

Pada bab ini berisi penjelasan tentang pemahaman-pemahaman penulis yang diperoleh dari tinjauan buku-buku referensi, makalah-makalah di jurnal-jurnal yang berhubungan dengan topik tugas akhir, khususnya deteksi wajah.

#### **2.1 Wajah**

Wajah merupakan bagian dari tubuh manusia yang menjadi fokus perhatian di dalam interaksi sosial. Wajah memainkan peranan vital dengan menunjukkan identitas seseorang. Wajah terutama digunakan untuk ekspresi wajah, penampilan, serta identitas. Tidak ada satu wajahpun yang serupa mutlak, bahkan pada manusia kembar identik sekalipun. Ada beberapa pengertian dan definisi wajah berdasarkan sudut pandang bidang keilmuan. Definisi wajah menurut kamus besar bahasa Indonesia merupakan bagian dari kepala; roman muka; muka. Wajah atau muka adalah bagian depan dari kepala pada manusia meliputi wilayah dari dahi hingga dagu. Bagian-bagian yang termasuk wilayah wajah yaitu rambut, dahi, alis, mata, hidung, pipi, mulut, bibir, gigi, kulit, dan dagu [WIK14].

Dalam ilmu kesehatan wajah berdasarkan anatominya merupakan bagian *anterior* (struktur bagian depan) dari kepala, dengan batas kedua telinga di *lateral* (struktur terjauh dari garis pertengahan tubuh), dagu di *inferior* (bagian terendah) dan garis batas tumbuhnya rambut di *superior* (bagian teratas). Wajah terbentuk dari tulang belakang dan jaringan lunak yang terletak di atasnya (jaringan otot, jaringan tulang rawan, pembuluh darah, saraf, pembuluh limfe dan kelenjar-kelenjar), yang secara bersama-sama memberikan tampilan dan fungsi dari wajah [MUH14].

Wajah merupakan bagian dari tubuh manusia yang bisa dikenali dengan teknologi biometrik. Pada teknologi biometrik ciri-ciri biologis dari manusia seperti wajah bisa memberikan informasi yang unik. Informasi unik tersebut dapat berupa karakteristik dari pola wajah sesuai dengan individu masing-masing. Karena karakteristik dari pola wajah bisa diukur dan dianalisis untuk proses deteksi maupun autentifikasi. Oleh karena itu wajah bisa digunakan sebagai organ dari tubuh manusia yang dijadikan indikasi untuk pengenalan seseorang [MUH14].

#### **2.2 Biometrik**

##### **2.2.1 Definisi Biometrik**

Biometrik (berasal dari bahasa Yunani *bios* yang artinya hidup dan *metron* yang artinya mengukur) adalah studi tentang metode otomatis untuk mengenali manusia berdasarkan satu atau lebih bagian tubuh manusia atau kelakuan dari manusia itu sendiri. Dalam dunia teknologi informasi, biometrik relevan dengan teknologi yang digunakan untuk menganalisa fisik dan kelakuan manusia untuk autentifikasi. Dalam dunia medis disebutkan bahwa ada berapa bagian tubuh kita yang sangat unik. Artinya, tidak dimiliki oleh lebih dari satu individu. Seperti sidik jari

atau retina mata. Meskipun bentuk atau warna mata bisa saja sama, namun retina mata belum tentu sama. Begitu juga dengan suara dan struktur wajah. Bagian-bagian unik inilah yang kemudian dikembangkan sebagai atribut keamanan [ADI12].

Para ahli mengungkapkan bahwa sangat mustahil untuk mengakali sebuah produk biometrik, karena karakteristik *fisiologis* tiap manusia tidak mungkin bisa sama persis. Hal tersebut bisa meningkatkan keamanan dan mengurangi terjadinya kesalahan dalam proses identifikasi. Atribut-atribut *fisiologis* seperti sidik jari, pola retina, suara, maupun pola wajah tidak diambil secara sembarangan sebagai bagian identifikasi, melainkan masing-masing atribut ini sesuai dengan sejumlah persyaratan untuk memastikan bahwa identitas ini sesuai untuk identifikasi biometrik. Ada tiga buah persyaratan penting untuk menentukan atribut-atribut *fisiologis* sebagai berikut [DAR14]:

1. *Inimitable* : atribut tidak dapat diimitasi atau diproduksi ulang.
2. *Singularity* : atribut harus tunggal dan unik untuk membedakan dari pemilik berbeda.
3. *Convenience* : atribut harus nyaman atau tidak mengganggu untuk diukur dan direkam .

### 2.2.2 Penerapan Teknologi Biometrik

Teknologi yang digunakan untuk masing-masing jenis biometrik tentunya berlainan, beberapa telah dapat dilakukan dan dapat ditemui di pasaran namun beberapa masih dalam tahap penelitian. Untuk penggunaan sebagai otentikasi, biometrik harus terlebih dahulu dimasukkan ke dalam *database* sebuah sistem atau program aplikasi. Kita dapat menggunakan biometrik untuk berbagai keperluan yang biasa, tetapi biometrik tidak dapat digunakan untuk hal-hal yang bersifat rahasia. Walaupun biometrik sangat bagus dan berguna tetapi bukanlah sebuah kunci, karena tidak dapat disembunyikan, tidak dapat dilakukan pengacakan dan tidak dapat ditingkatkan atau dihancurkan [DAR14].

Ada banyak penerapan teknologi biometrik yang sudah banyak berkembang saat ini. Berikut ini pengenalan beberapa penggunaan biometrik yang diterapkan pada berbagai alat yang digunakan konsumen saat ini, berikut dengan sisi positif dan negatifnya ketika digunakan untuk menggantikan penggunaan kata sandi [DAR14].

#### 1. Pemindaian Sidik Jari

Pemindaian sidik jari, saat ini telah menjadi salah satu bentuk penggunaan teknologi biometrik yang paling biasa digunakan, dan menurut Frost & Sullivan, dapat menjadi teknologi biometrik terdepan setidaknya untuk beberapa tahun ke depan. Pemindaian sidik jari bekerja dengan cara memindai bentuk alur-alur pada permukaan kulit jari. Salah satu realisasi pemindaian sidik jari pada perangkat elektronik adalah antara lain sensor sidik jari pada *smartphone*. Pemindaian sidik jari sangat sesuai dengan kriteria *Convenience & Singularity*, namun tidak sesuai dengan kriteria *Inimitable as Other Forms of Biometrics* karena sangat memungkinkan untuk membuat sidik jari tiruan, asalkan jejak sidik jari yang utuh dapat diambil dari permukaan layar [DAR14].

## 2. Pemindaian Iris dan Retina

Pemindaian retina dan iris bekerja dengan prinsip yang kira-kira sama, pola iris dan retina yang unik yang menjadikannya cocok untuk identifikasi biometrik. Pemindaian retina dilakukan dengan cara pemindaian bola mata ke dalam bola mata berdasarkan pola retina sedangkan pemindaian iris berdasarkan pola iris namun hanya pemindaian bola mata di permukaan yang lebih luar. Iris dan retina, kedua-duanya adalah tunggal dan tidak dapat ditiru, dengan kemungkinan salah interpretasi yang sangat kecil. Pemindaian retina tidak cukup nyaman dilakukan karena hal ini menuntut mata kita dihadapkan sangat dekat dengan perangkat pemindai. Pemindaian iris jauh lebih nyaman digunakan, karena hanya memerlukan kamera dengan sinar inframerah biasa dan dapat dilakukan dengan jarak tertentu [DAR14].

## 3. Pengenalan Wajah

Perangkat lunak pengenalan wajah mengenali wajah seseorang setelah sebelumnya dilakukan proses pendeteksian wajah berdasarkan penanda-penanda tertentu seperti jarak antara mata, lebar hidung dan bentuk dari tulang pipi. Proses deteksi wajah (*face detection*) merupakan tahapan awal dari proses pengenalan wajah (*face recognition*). Salah satu isu terbesar teknologi pengenalan wajah adalah bahwa metode ini tidak seratus persen aman. Sistem-sistem tertentu dapat diretas dengan menggunakan foto target pengguna. Namun tidak semua sistem memiliki kelemahan ini. Aspek tunggal dan kenyamanan bukanlah masalah pada metode pengenalan wajah. Namun yang lebih penting dengan metode ini adalah perangkat lunak yang digunakan. Secara spesifiknya adalah bahwa perangkat lunak yang digunakan mampu mengenali wajah pada berbagai kondisi pencahayaan dan manakala pada wajah terdapat berbagai aksesoris seperti penggunaan antara lain kacamata dan topi [DAR14].

## 4. Pengenalan Suara

Identifikasi biometrik berdasarkan suara belum digunakan secara luas pada saat ini, namun pasarnya diharapkan dapat tumbuh dalam beberapa tahun ke depan. Pengenalan suara (*voice recognition*), bedakan dengan pengenalan perkataan (*speech recognition*), bekerja dengan cara mengenali karakteristik suara dari seseorang. Metode pengenalan suara merupakan identifikasi biometrik yang sangat nyaman diterapkan dan berbiaya murah karena teknologi untuk mengumpulkan data suara tersedia pada berbagai perangkat yang kita gunakan sehari-hari seperti *smartphone*. Di samping itu, suara benar-benar tidak dapat ditiru dan cukup tunggal ketika digunakan untuk keperluan identifikasi pengguna. Ada beberapa kelemahan pada metode pengenalan suara. Pertama, suara latar dapat mempengaruhi keakuratan analisis. Di samping itu, penyakit seperti iritasi tenggorokan dapat menyebabkan perubahan suara yang menyebabkan keakuratan pengenalan menurun [DAR14].

## 5. Biometrik Denyut Jantung

Penggunaan tanda-tanda vital seperti denyut jantung untuk metode identifikasi bukanlah hal yang sama sekali baru. Denyut jantung seseorang, atau gelombang *elektrokardiogram* (ECG), adalah unik dan konstan bahkan ketika frekuensi denyut meningkat karena emosi senang atau latihan. Denyut jantung mungkin dapat menjadi salah satu atribut fisiologi yang paling aman yang dapat digunakan untuk identifikasi biometrik. Gelombang ECG seseorang adalah unik, sulit ditiru dan dibuat. Pengindaian ECG juga cukup nyaman dilakukan karena tidak memerlukan prosedur pengindaian invasif atau perangkat-perangkat khusus yang rumit [DAR14].

### 2.3 Deteksi Wajah

Deteksi wajah dapat dipandang sebagai masalah klasifikasi pola dimana inputnya adalah citra masukan dan akan ditentukan *output* yang berupa label kelas dari citra tersebut. Dalam hal ini terdapat dua label kelas, yaitu wajah dan non-wajah [KAH98].

Teknik-teknik pengenalan wajah yang dilakukan selama ini banyak yang menggunakan asumsi bahwa data wajah yang tersedia memiliki ukuran yang sama dan latar belakang yang seragam. Di dunia nyata, asumsi ini tidak selalu berlaku karena wajah dapat muncul dengan berbagai ukuran dan posisi di dalam citra dan dengan latar belakang yang bervariasi [DWI12].

Pendeteksian wajah (*face detection*) adalah salah satu tahap awal (*preprocessing*) yang sangat penting sebelum dilakukan proses pengenalan wajah (*face recognition*). Bidang-bidang penelitian yang juga berkaitan dengan pemrosesan wajah (*face processing*) adalah autentikasi wajah (*face authentication*), lokalisasi wajah (*face localization*), penjejakan wajah (*face tracking*), dan pengenalan ekspresi wajah (*facial expression recognition*) [YAN02].

- a. Pengenalan wajah (*face recognition*) yaitu membandingkan citra wajah masukan dengan suatu database wajah dan menemukan wajah yang paling cocok dengan citra masukan tersebut.
- b. Autentikasi wajah (*face authentication*) yaitu menguji keaslian/kesamaan suatu wajah dengan data wajah yang telah diinputkan sebelumnya.
- c. Lokalisasi wajah (*face localization*) yaitu pendeteksian wajah namun asumsi hanya ada satu wajah di dalam citra
- d. Penjejakan wajah (*face tracking*) yaitu memperkirakan lokasi suatu wajah di dalam video secara real time.
- e. Pengenalan ekspresi wajah (*facial expression recognition*) untuk mengenali kondisi emosi manusia.

Pada sebuah penelitian dijelaskan bahwa ada beberapa tantangan yang dihadapi pada masalah deteksi wajah disebabkan oleh adanya faktor-faktor berikut [YAN02] :

- a. Terhalang objek lain. Citra wajah dapat terhalangi sebagian oleh objek atau wajah lain, misalnya pada citra berisi sekelompok orang.

- b. Kondisi pengambilan citra. Citra yang diperoleh sangat dipengaruhi oleh faktor-faktor seperti intensitas cahaya ruangan, arah sumber cahaya, dan karakteristik sensor dan lensa kamera.
- c. Posisi wajah. Posisi wajah di dalam citra dapat bervariasi karena posisinya bisa tegak, miring, menoleh, atau dilihat dari samping.
- d. Komponen-komponen pada wajah yang bisa ada atau tidak ada, misalnya kumis, jenggot, dan kacamata.
- e. Ekspresi wajah. Penampilan wajah sangat dipengaruhi oleh ekspresi wajah seseorang, misalnya tersenyum, tertawa, sedih, berbicara, dan sebagainya.

## 2.4 Algoritma Viola-Jones

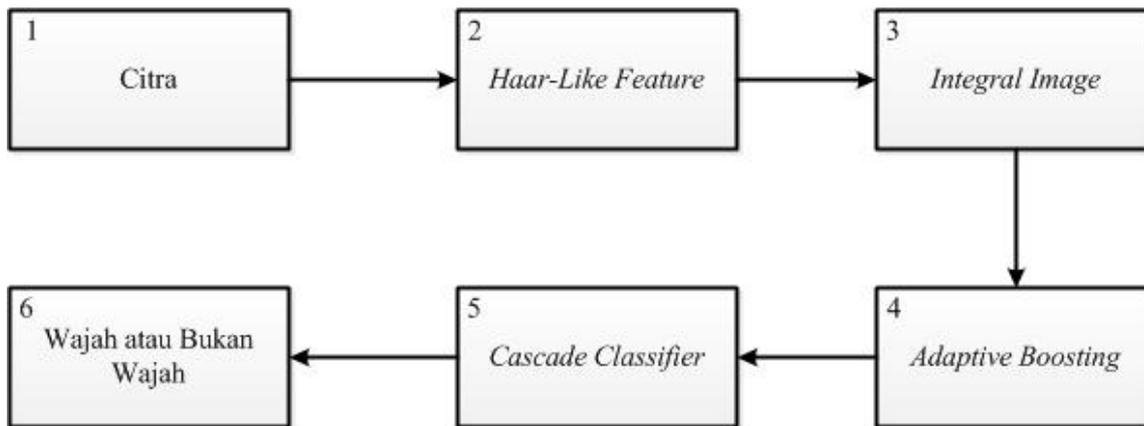
Saat ini telah banyak berkembang aplikasi-aplikasi yang menggunakan fitur deteksi wajah. Deteksi wajah sendiri dapat dilakukan dengan berbagai cara, salah satunya menggunakan algoritma *Viola-Jones*. Algoritma *Viola-Jones* merupakan algoritma yang paling banyak digunakan untuk mendeteksi wajah. *Viola-Jones* telah memperkenalkan sebuah *framework* deteksi wajah yang mampu memproses gambar dengan sangat cepat dengan tingkat deteksi yang tinggi. *Viola-Jones* menerapkan algoritma *Adaboost* (*adaptive boosting*) algoritma yang dapat meningkatkan kinerja pendeteksian. Dalam prosesnya algoritma *Adaboost* memainkan peran dalam memilih fitur yang cocok untuk mendeteksi objek yang menarik [WUL14].

Dalam *framework Viola-Jones* disediakan banyak *library* untuk melakukan proses seleksi fitur, fitur yang merupakan fungsi dasar untuk meningkatkan proses seleksi dikenal sebagai *Haar-Like feature*. Didorong oleh hasil karya Tieu dan Viola, *Viola-Jones* terhambat oleh seleksi fitur yang harus berdasarkan pada setiap *classifier* yang lemah sehingga tergantung pada satu fitur tunggal. *Viola-Jones* memperkenalkan sebuah representasi *image* baru yang dikenal sebagai *Integral Image* yang tidak terpisahkan. Dengan metode ini *Haar-Like features* dapat dihitung pada setiap skala atau lokasi dalam waktu yang konstan [VIO04].

Pada sebuah jurnal *EECCIS* kedelapan disebutkan bahwa metode *Viola-Jones* merupakan metode pendeteksian obyek yang memiliki tingkat keakuratan yang cukup tinggi yaitu sekitar 93,7 % dengan kecepatan 15 kali lebih cepat dari pada detektor *Rowley Baluja-Kanade* dan kurang lebih 600 kali lebih cepat dari pada detektor *Schneiderman-Kanade*. Metode *Viola-Jones* menggabungkan empat kunci utama yaitu *Haar-Like Feature*, *Integral Image*, *Adaboost* dan *Cascade classifier* yang penjelasannya bisa dilihat di bawah ini [AND14].

### 2.4.1 Cara Kerja Deteksi Wajah Viola-Jones

Pada gambar 2.1 berikut ini merupakan skema cara kerja deteksi wajah dengan menggunakan algoritma *Viola-Jones* mulai dari awal proses pendeteksian sampai dengan hasil dari proses pendeteksian.



Gambar 2.1 Skema Proses deteksi wajah dengan metode Viola-Jones [DWI12]

Berikut ini merupakan penjelasan dari skema proses yang dilakukan dalam deteksi wajah dengan metode *Viola-Jones* [DWI12] :

1. Proses pertama yaitu membaca sampel citra wajah dari sebuah gambar ataupun membaca sampel citra wajah yang sedang menghadap ke kamera.
2. Dari citra yang sudah ditangkap maka akan dilakukan pembacaan fitur *Haar* dengan memproses citra tersebut menjadi kotak-kotak untuk mendapatkan perbedaan nilai *threshold* dari daerah gelap dan daerah terang dari citra tersebut. Jika nilai perbedaann antara daerah gelap dan daerah terang di atas nilai ambang atau *threshold*, maka dapat dikatakan bahwa fitur tersebut ada.
3. Selanjutnya untuk menentukan ada atau tidaknya dari ratusan fitur *Haar* pada sebuah citra dan pada skala yang berbeda secara efisien digunakan *Integral Image*. Pada umumnya, pengintegrasian tersebut menambahkan unit-unit kecil secara bersamaan. Dalam hal ini unit-unit kecil tersebut adalah nilai-nilai piksel. Nilai integral untuk masing-masing piksel adalah jumlah dari semua piksel-piksel dari atas sampai bawah. Dimulai dari kiri atas sampai kanan bawah, keseluruhan citra itu dapat dijumlahkan dengan beberapa operasi bilangan bulat per piksel.
4. Kemudian untuk memilih fitur *Haar* yang spesifik yang akan digunakan dan untuk mengatur nilai ambangnya (*threshold*) digunakan sebuah metode *machine learning* yang disebut *AdaBoost*. *AdaBoost* menggabungkan banyak *classifier* lemah untuk membuat sebuah *classifier* kuat. Dengan menggabungkan beberapa *AdaBoost classifier* sebagai rangkaian filter yang cukup efisien untuk menggolongkan daerah *image*. Masing-masing filter adalah satu *AdaBoost classifier* terpisah yang terdiri *classifier* lemah atau satu filter *Haar*. Selama proses filter, bila ada salah satu filter gagal untuk melewati sebuah daerah citra, maka daerah itu langsung digolongkan sebagai bukan wajah. Namun ketika filter melewati sebuah daerah citra dan sampai melewati semua proses filter yang ada dalam rangkaian filter, maka daerah citra tersebut digolongkan sebagai wajah.

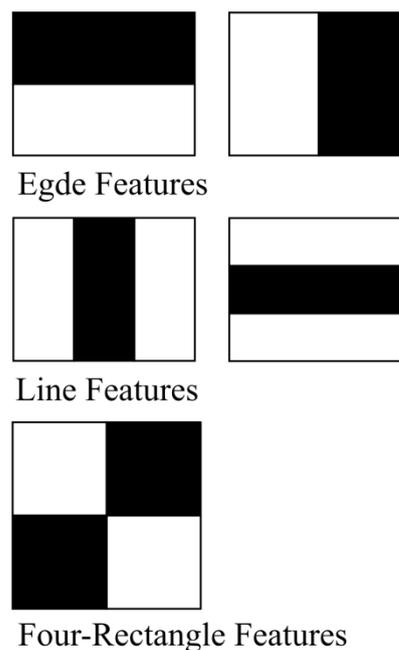
5. Tahap selanjutnya yaitu *cascade classifier*. Urutan filter pada *cascade* ditentukan oleh bobot yang diberikan *AdaBoost*. Filter dengan bobot paling besar diletakkan pada proses pertama kali, bertujuan untuk menghapus daerah citra bukan wajah secepat mungkin.
6. Tahapan yang terakhir adalah menampilkan objek sampel citra yang telah terdeteksi wajah ataupun bukan wajah.

Di bawah ini merupakan penjelasan dari tiap-tiap proses utama dalam pendeteksian wajah *Viola-Jones* seperti *Haar-Like Feature*, *Integral Image*, *Adaptive Boosting* dan *Cascade Classifier*.

#### 2.4.2 Haar-Like Feature

Secara umum *Haar-Like Feature* digunakan dalam mendeteksi objek pada *image* digital. Nama *Haar* merujuk pada suatu fungsi matematika (*Haar Wavelet*) yang berbentuk kotak, prinsipnya sama seperti pada fungsi *Fourier*. Awalnya pengolahan gambar hanya dengan melihat dari nilai RGB setiap *pixel*, namun metoda ini ternyata tidaklah efektif [GOF12].

Viola dan Jones kemudian mengembangkannya sehingga terbentuk *Haar-Like feature*. *Haar-Like feature* yang digunakan dalam *Viola-Jones* ada tiga jenis yaitu *edge features*, *line features*, *four-rectangle features* dapat dilihat pada gambar 2.2 [GOF12].



Gambar 2.2 Jenis *Haar-Like Feature* [ENO14]

*Haar-like feature* memproses gambar dalam kotak-kotak, dimana dalam satu kotak terdapat beberapa *pixel*. Per kotak itu pun kemudian diproses dan didapatkan perbedaan nilai (*threshold*) yang menandakan daerah gelap dan terang. Nilai-nilai inilah yang nantinya dijadikan dasar dalam *image processing* [GOF12].

Nilai *Haar-Like Feature* diperoleh dari selisih jumlah nilai piksel daerah gelap dengan jumlah nilai piksel daerah terang :

$$F(\text{Haar}) = \sum F \text{ White} - \sum F \text{ Black}$$

(1.2)

$F(\text{Haar})$  = Nilai fitur total

$\sum F \text{ White}$  = Nilai fitur pada daerah terang

$\sum F \text{ Black}$  = Nilai fitur pada daerah gelap

Kotak rectangular *Haar-Like Feature* dapat dihitung secara cepat dengan menggunakan *Integral Image*. Lalu untuk gambar bergerak (*video*), perhitungan dan penjumlahan *pixel* terjadi secara terus menerus dan membutuhkan waktu yang lama. Oleh karena itu, penjumlahan diganti dengan integral sehingga didapatkan hasil lebih cepat. Hasil deteksi dari *Haar-Like* kurang akurat jika hanya menggunakan satu fungsi saja sehingga biasanya digunakan beberapa fungsi sekaligus. Semakin banyak fungsi yang digunakan maka hasilnya akan semakin akurat. Pemrosesan *Haar-Like feature* yang banyak tersebut diorganisir atau diatur di dalam *classifier cascade* [GOF12].

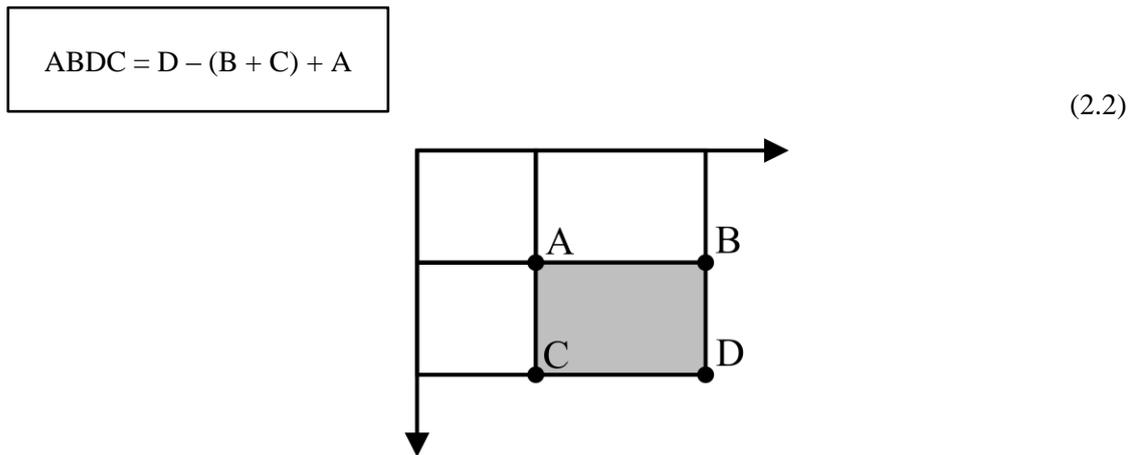
### 2.4.3 Integral Image

*Integral Image* adalah sebuah citra yang nilai tiap pikselnya merupakan penjumlahan dari nilai piksel kiri atas hingga kanan bawah. Untuk memudahkan proses perhitungan nilai dari setiap fitur *Haar* pada setiap lokasi gambar digunakan teknik yang disebut citra integral. Secara umum integral mempunyai makna menambahkan bobot, bobot merupakan nilai-nilai piksel yang akan ditambahkan ke dalam gambar asli. Nilai integral dari setiap piksel merupakan jumlah dari semua piksel sebelah atasnya dan di sebelah kirinya. Keseluruhan gambar dapat diintegrasikan dengan operasi bilangan bulat per piksel. Contoh dari perhitungan *integral image* dapat dilihat pada gambar 2.3 [ILH10].

1	2	3	1	3	6
4	5	6	5	12	21
7	8	9	12	27	45
Citra Masukan			Citra Integral		

Gambar 2.3 Contoh Citra Integral [ILH10]

Dengan mendapatkan nilai dari citra integral maka jumlah dari seluruh piksel yang ada dalam setiap persegi panjang dapat dihitung dengan empat nilai. Nilai-nilai ini merupakan piksel pada citra integral yang bertepatan dengan sudut-sudut persegi panjang yang ada pada citra masukan. Rumus dari perhitungan tersebut adalah [ILH10] :



Gambar 2.4 Penjumlahan Nilai Per Pixel dari Setiap Kotak [ILH10]

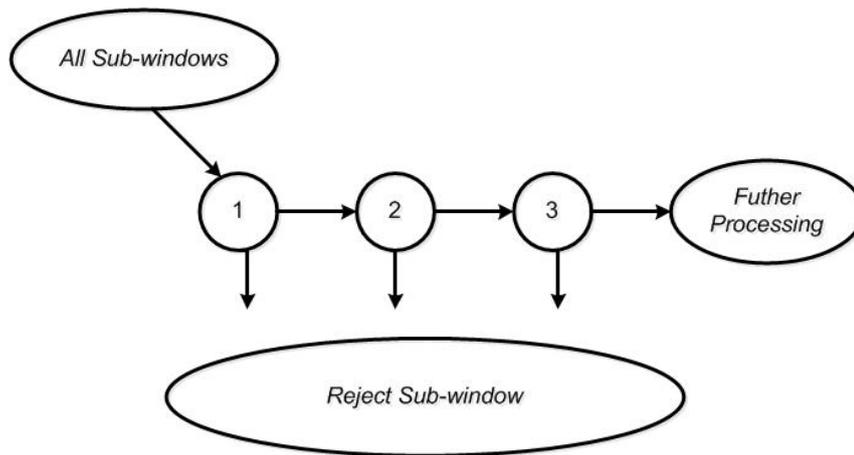
Untuk memilih fitur *Haar* yang khusus untuk digunakan dalam proses pendeteksian wajah dan untuk menetapkan ambang batas maka digunakan teknik pembelajaran yang disebut sebagai algoritma *AdaBoost* (*Adaptive Boosting*) [ILH10].

#### 2.4.4 Adaptive Boosting

Untuk memilih fitur yang spesifik yang akan digunakan dan untuk mengatur nilai ambangnya (*threshold*), Viola dan Jones menggunakan sebuah metode *machine learning* yang disebut *AdaBoost*. *AdaBoost* menggabungkan banyak *classifier* lemah untuk membuat sebuah *classifier* kuat. Lemah disini berarti urutan filter pada *classifier* hanya mendapatkan jawaban benar lebih sedikit. Jika keseluruhan *classifier* lemah digabungkan maka akan menjadi *classifier* yang lebih kuat. *AdaBoost* memilih sejumlah *classifier* lemah untuk disatukan dan menambahkan bobot pada setiap *classifier*, sehingga akan menjadi *classifier* yang kuat. Viola-Jones menggabungkan beberapa *AdaBoost classifier* sebagai rangkaian filter yang cukup efisien untuk menggolongkan daerah *image*. Masing-masing filter adalah satu *AdaBoost classifier* terpisah yang terdiri *classifier* lemah atau satu filter fitur [DWI12].

#### 2.4.5 Cascade Classifier

Karakteristik dari algoritma Viola-Jones adalah adanya klasifikasi bertingkat. Klasifikasi pada algoritma ini terdiri dari tiga tingkatan dimana tiap tingkatan mengeluarkan subcitra yang diyakini bukan wajah. Hal ini dilakukan karena lebih mudah untuk menilai subcitra tersebut bukan wajah ketimbang menilai apakah subcitra tersebut berisi wajah [LAT12].



Gambar 2.5 *Cascade Classifier* [LAT12]

Pada klasifikasi tingkat pertama, tiap subcitra akan diklasifikasi menggunakan satu fitur. Hasil dari klasifikasi pertama ini berupa T (*True*) untuk gambar yang memenuhi fitur *Haar* tertentu dan F (*False*) bila tidak. Klasifikasi ini kira-kira akan menyisakan 50% subcitra untuk diklasifikasi di tahap kedua. Hasil dari klasifikasi kedua berupa T (*True*) untuk gambar yang memenuhi proses integral image dan F (*False*) bila tidak. Seiring dengan bertambahnya tingkatan klasifikasi, maka diperlukan syarat yang lebih spesifik sehingga fitur yang digunakan menjadi lebih banyak. Jumlah subcitra yang lolos klasifikasi pun akan berkurang hingga mencapai jumlah sekitar 2%. Hasil dari klasifikasi terakhir berupa T (*True*) untuk gambar yang memenuhi proses *AdaBoost* dan F (*False*) bila tidak [DWI12].

## 2.5 MATLAB

### 2.5.1 Sejarah MATLAB

MATLAB yang berarti "*matrix laboratory*" diciptakan pada akhir tahun 1970-an oleh Cleve Moler. Ia merencangkannya untuk memberikan akses bagi mahasiswa dalam memakai LINPACK dan EISPACK tanpa harus mempelajari Fortran. Jack Little, seorang insinyur, dipertemukan dengan karyanya tersebut selama kunjungan Moler ke Universitas Stanford pada tahun 1983. Menyadari potensi komersialnya, ia bergabung dengan Moler dan Steve Bangert. Mereka menulis ulang MATLAB dalam bahasa pemrograman C, kemudian mendirikan *The MathWorks* pada tahun 1984 untuk melanjutkan pengembangannya. Pustaka yang ditulis ulang tadi kini dikenal dengan nama JACKPAC. Pada tahun 2000, MATLAB ditulis ulang dengan pemakaian sekumpulan pustaka baru untuk manipulasi matriks, LAPACK [AMA12].

MATLAB pertama kali diadopsi oleh insinyur rancangan kontrol yang juga spesialisasi Little, tapi lalu menyebar secara cepat ke berbagai bidang lain. Kini juga digunakan di bidang pendidikan, khususnya dalam pengajaran aljabar linear dan analisis numerik, serta populer di kalangan ilmuwan yang menekuni bidang pengolahan citra. Dalam perkembangannya, MATLAB mampu mengintegrasikan beberapa software matriks sebelumnya dalam satu *software* untuk komputasi

matriks. Tidak hanya itu, MATLAB juga mampu melakukan komputasi simbolik yang biasa dilakukan oleh MAPLE. Berikut ini beberapa fungsi MATLAB digunakan untuk [AMA12] :

1. Matematika dan komputasi
2. Pengembangan dan algoritma
3. Pemodelan, simulasi dan pembuatan *prototype*
4. Analisa data, eksplorasi dan visualisasi
5. Analisis numerik dan statistic
6. Pembuatan aplikasi termasuk pembuatan *graphical user interface*

Sistem MATLAB terdiri atas lima bagian utama yaitu [AMA12] :

1. *Development Environment*. Ini adalah kumpulan semua alat-alat dan fasilitas untuk membantu kita dalam menggunakan fungsi dan *file* MATLAB. Bagian ini memuat *desktop*, *Command window*, *command history*, *editor and debugger*, dan *browser* untuk melihat *help*, *workspace*, *files*.
2. *The MATLAB Mathematical Function Library*. Bagian ini adalah koleksi semua algoritma komputasi, mulai dari fungsi sederhana seperti *sum*, *sine*, *cosine* sampai fungsi lebih rumit seperti *invers matriks*, nilai *eigen*, fungsi *Bessel* dan *fast Fourier transform*.
3. *The MATLAB language*. Ini adalah bahasa *matriks/array level* tinggi dengan *control flow*, fungsi, struktur data, *input/output*, dan fitur objek *programming* lainnya.
4. *Graphics*. MATLAB mempunyai fasilitas untuk menampilkan *vector* dan *matriks* sebagai grafik. Fasilitas ini mencakup visualisasi data dua/tiga dimensi, pemrosesan citra (*image*), animasi, dan grafik animasi.
5. *The MATLAB Application Program Interface (API)*. Paket ini memungkinkan kita menulis bahasa C dan Fortran yang berinteraksi dengan MATLAB. Ia memuat fasilitas untuk pemanggilan kode-kode dari MATLAB (*dynamic linking*), yang disebut MATLAB sebagai mesin penghitung, dan untuk membaca dan menulis *MAT-files*.

### 2.5.2 Perkembangan MATLAB

MATLAB (*Matrix Laboratory*) adalah sebuah program untuk analisis dan komputasi numerik dan merupakan suatu bahasa pemrograman matematika lanjutan yang dibentuk dengan dasar pemikiran menggunakan sifat dan bentuk matriks. Pada awalnya, program ini merupakan *interface* untuk koleksi rutin-rutin numerik dari proyek LINPACK dan EISPACK, dan dikembangkan menggunakan bahasa FORTRAN namun sekarang merupakan produk komersial dari perusahaan *Mathworks, Inc.* yang dalam perkembangan selanjutnya dikembangkan menggunakan bahasa C++ dan *assembler* (utamanya untuk fungsi-fungsi dasar MATLAB) [ARI14].

MATLAB telah berkembang menjadi sebuah environment pemrograman yang canggih yang berisi fungsi-fungsi *built-in* untuk melakukan tugas pengolahan sinyal, aljabar linier, dan kalkulasi matematis lainnya. MATLAB juga berisi *toolbox* yang berisi fungsi-fungsi tambahan untuk aplikasi khusus. MATLAB bersifat *extensible*, dalam artian bahwa seorang pengguna ketika

fungsi fungsi *built-in* yang tersedia tidak dapat melakukan tugas tertentu. Kemampuan pemrograman yang dibutuhkan tidak terlalu sulit bila telah memiliki pengalaman dalam pemrograman bahasa lain seperti C, PASCAL, atau FORTRAN [ARI14].

MATLAB merupakan merk *software* yang dikembangkan oleh *Mathworks.Inc*. MATLAB merupakan *software* yang paling efisien untuk perhitungan numeric berbasis matriks. Dengan demikian jika di dalam perhitungan kita dapat menformulasikan masalah ke dalam format matriks maka MATLAB merupakan *software* terbaik untuk penyelesaian numeriknya. Dikemudian hari MATLAB membelilisensi dari MAPLE sehingga MATLAB juga didukung komputasi simbolik [ARI14].

MATLAB dapat digunakan untuk melakukan bermacam-macam komputasi, mulai dari aritmatika sederhana sampai perhitungan matriks yang sangat rumit. MATLAB juga bisa membuat grafik dua dimensi dan tiga dimensi. Yang unik di dalam bahasa MATLAB adalah MATLAB hanya mengenal satu struktur data, yaitu matriks. Vektor adalah matriks kolom dan skalar adalah matriks baris. Karena hanya mengenal satu macam struktur data, maka variabel di dalam bahasa MATLAB tidak perlu dideklarasikan dulu dan bisa langsung diberi nilai [ARI14].

### 2.5.3 OpenCV Library

Seperti pada aplikasi pemrograman lainnya, MATLAB juga menyediakan *library-library* yang dapat digunakan sesuai dengan kebutuhan pemrograman. Pada aplikasi pengolahan citra misalnya, dalam pembuatan aplikasi pendeteksi wajah sesuai dengan penelitian yang sedang dilakukan, *library OpenCV* ini bisa digunakan. *OpenCV (Open Computer Vision)* merupakan *library* yang digunakan dalam membuat aplikasi yang berkaitan dengan *computer vision*. *OpenCV* juga menyediakan banyak fungsionalitas, yang tentunya dapat mempersingkat waktu dalam pembuatan aplikasi yang lumayan rumit. *Library OpenCV* bisa diintegrasikan dengan pemrograman MATLAB dengan menggunakan *MEX-Interface*, untuk mengembangkan sebuah agloritma, analisis data, dan komputasi numerik pada MATLAB. MATLAB menyediakan sebuah *MEX-environment* yang bisa digunakan untuk menulis fungsi bahasa pemrograman C. *MEX (Matlab-executable)* merupakan sebuah *file* yang dinamis yang terhubung dengan C/C++ atau FORTRAN yang ketika dikompilasi bisa dijalankan pada MATLAB seperti *M-File*. Oleh karena itu *MEX-environment* menawarkan cara untuk memanggil custom C/C++ seolah-olah menggunakan fungsi pada MATLAB sendiri [GEO14].

*OpenCV* merupakan singkatan dari *Intel Open Source Computer Vision Library* yang sekurang-kurangnya terdiri dari tiga ratus fungsi-fungsi C, bahkan bisa lebih. *OpenCV* dapat beroperasi pada komputer berbasis *windows* ataupun *linux*. Beberapa contoh aplikasi dari *OpenCV* adalah pada *Human Computer Interaction* (interaksi manusia komputer) yaitu [KUR10] :

1. *Object Indentification* (identifikasi objek)
2. *Segmentation* (segmentasi) dan *Recognition* (pengenalan)
3. *Face Recognition* (pengenalan wajah)
4. *Gesture Recognition* (pengenalan gerak isyarat)
5. *Motion Tracking* (penjajakan gerakan)
6. *Ego Motion* (gerakan ego) dan *Motion Understanding* (pemahaman gerakan)
7. *Structure From Motion* (gerakan dari struktur)
8. *Mobile Robotics* (robot-robot yang bergerak)

*OpenCV* menggunakan sebuah tipe *face detector* yang disebut *Haar-cascade classifier*. Jika ada sebuah *image* yang berasal dari *file image* atau *live video*, *face detector* akan menguji tiap lokasi *image* dan mengklasifikasinya sebagai wajah atau bukan wajah. Klasifikasi wajah ini menggunakan sebuah pemisalan skala yang tetap, misalnya  $50 \times 50$  *pixel*. Jika wajah pada *image* lebih besar atau lebih kecil dari *pixel* tersebut, *classifier* terus menerus jalan beberapa kali, untuk mencari wajah pada gambar tersebut [KUR10].

*Classifier* menggunakan data yang disimpan pada *file XML* untuk memutuskan bagaimana mengklasifikasi tiap lokasi *image*. *OpenCV* menggunakan 4 data XML untuk deteksi wajah depan, dan satu untuk wajah *profile*. Termasuk juga tiga *file XML* untuk bukan wajah satu untuk mendeteksi badan secara penuh, satu untuk badan bagian atas, dan satu untuk badan bagian bawah. Kita harus memberitahukan (mendeklarasikan) letak dari *classifier* yang digunakan. Salah satunya bernama *haarcascade\_frontalface\_default.xml* [KUR10].