BAB 2
TINJAUAN PUSTAKA

2.1. Profil Sekolah SMPN 3 Rangkasbitung

Pada tanggal 5 november 1977 SMPN 3 Rangkasbitung mulai berdiri dengan SK (Surat Keputusan) Nomor 2433/E.15/II/1977 (SK Kepala Kantor wilayah Depdikbud Provinsi Jawa Barat), dengan menggunakan bangunan yang masih menginduk ke SMPN 1 Rangkasbitung dari tahun 1979 sampai dengan 1980. Yang mampu menampung 123 orang siswa untuk 4 kelas. Pada tahun 1980-1981 gedung SMPN 3 Rangkasbitung mulai dibangun.

Tepatnya tanggal 2 mei 1982 gedung SMPN 3 Rangkasbitung mulai resmi digunakan sampai sekarang dengan alamat di Jl. Jendral sudirman No.47, telp (0252) 201851, email : smpn3rksbitung@gmail.com.

Setelah melakukan wawancara dan observasi langsung ke SMPN 3 Rangkasbitung, maka didapat data sekolah saat ini yang dapat membantu terlaksanya penelitian ini, diantaranya adalah jumlah siswa kelas VII 260 siswa, kelas VIII 274 siswa dan kelas IX 241 siswa. Dalam hal fasilitas sekolah saat ini SMPN 3 Rangkasbitung memiliki 19 ruang kelas/belajar, 3 buah laboratorium yaitu laboratorium fisika, laboratorium biologi, dan laboratorium komputer. Selain itu satu unit ruang workshop/untuk praktek keterampilan dan juga digunakan sebagai ruang serba guna, satu unit masjid/laboratorium imtaq, satu ruang BP/BK, satu ruang guru, satu ruang kepala sekolah, satu ruang tata usaha, satu unit perpustakaan, dua unit kantin sehat sekolah, satu unit kantin hasil kerja sama dengan PT. Sosro, empat unit kantin yang dikelola oleh masyarakat, satu ruang osis, satu unit rumah penjaga sekolah, satu ruang UKS, dua unit WC guru, tiga belas ruang WC siswa, dua ruang gudang, satu ruang studio radio komunitas, satu lapangan basket, dua lapangan volley, satu lapangan badminton, satu halaman lapangan upacara, satu bangsal parkir motor guru, tiga unit meteran sumber listrik kapasitasnya 2500watt dan dua unit 1300watt, tiga unit sumber air (satu sumber air dari pompa jet pump, satu unit sumber air dari sumur gali, dan dua buah

meteran sumber air dari PDAM), satu ruang ganti anak putri ketika berolah raga, serta dia gudang penyimpanan dokumen.

2.1.1. Logo SMPN 3 Rangkasbitung

Sebuah lembaga pendidikan mempunyai sebuah tanda pengenal berupa logo. Berikut merupakan logo dari sekolah SMPN 3 Rangkasbitung seperti yang terdapat pada Gambar 2.1.



Gambar 2.1 Logo SMPN 3 Rangkasbitung

2.1.2. Visi dan Misi

Visi dari Sekolah SMPN 3 Rangkasbitung adalah Religius, Unggul, Visioner dan Berwawasan Lingkungan.

Adapun misi dari Sekolah SMPN 3 Rangkasbitung yaitu :

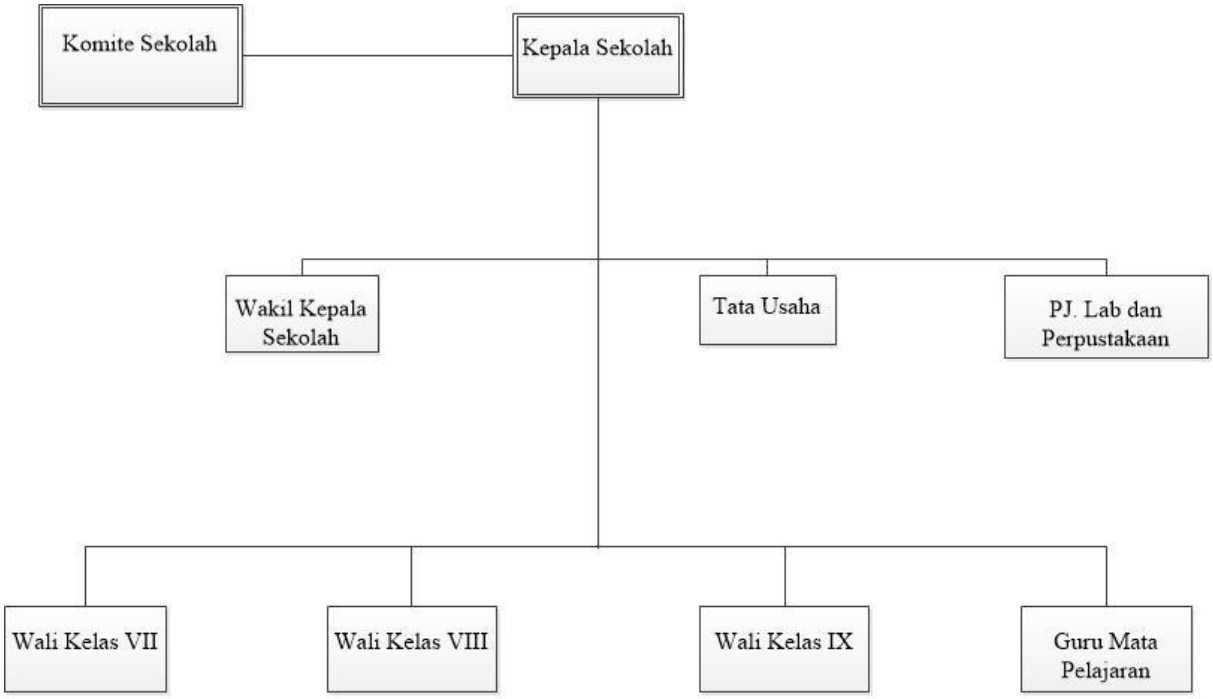
1. Membentuk generasi yang beriman, bertaqwa dan berakhlak mulia
2. Membentuk generasi yang kreatif, mandiri, terampil, dan berjiwa kompetitif
3. Membudayakan hidup sehat dan peduli lingkungan
4. Membentuk generasi yang inovatif dam berwawasan global
5. Memberikan keteladanan, meningkatkan etos kerja, dan kemampuan professional
6. Mengoptimalkan manajemen berbasis sekolah
7. Menjalin mitra kerja dengan instansi terkait

2.1.3. Struktur Organisasi

Berikut merupakan Struktur Organisasi dari SMPN 3 Rangkasbitung seperti yang terdapat pada Gambar 2.2.

STRUKTUR ORGANISASI

SMPN 3 Rangkasbitung



Gambar 2.2 Struktur Organisasi SMPN 3 Rangkasbitung

Berikut adalah deskripsi tugas dari masing-masing bagian yang terdapat di sekolah SMPN 3 Rangkasbitung:

1. **Kepala Sekolah**

   Tugas Utama:

   a. Menyusun Perencanaan
   b. Mengorganisasikan Kegiatan
   c. Mengaragkan/mengendalikan kegiatan
   d. Mengkoordinasikan kegiatan
   e. Melaksanakan pengawasan
   f. Menentukan kebijaksanaan
   g. Mengadakan rapat mengambil keputusan
   h. Mengatur proses belajar mengajar
   i.

2. Wakil Kepala Sekolah

Tugas Utama:

a. Penyusunan rencana, pembuatan program kegiatan dan program pelaksanaan
b. Pengorganisasian
c. Pengarahan
d. Ketenagakerjaan
e. Pengkoordinasian
f. Pengawasan
g. Identifikasi dan pengumpulan data
h. Pengembangan keunggulan
i. Penyusunan laporan

3. Tata Usaha

Tugas Utama:

a. Penyusunan program kerja tata usaha sekolah
b. Pengelolaan keuangan sekolah
c. Pengurus administrasi ketenagaan dan siswa
d. Penyusunan administrasi perlengkapan
e. Penyusun dan penyajian data/statistik sekolah
f. Penyusunan laporan pelaksanaan kegiatan pengurusan keteta usahaan secara berkala

4. PJ. Lab dan Perpustakaan

4.1 PJ. Labolatorium

Tugas utama :

a. Perencanaan pengadaan alat dan bahan laboratorium
b. Menyusun jadwal dan tata tertib penggunaan laboratorium
c. Mengatur penyimpanan dan daftar alat-alat laboratorium
d. Menyusun laporan pelaksanaan kegiatan laboratorium

4.2 PJ. Perpustakaan

Tugas Utama:

a. Perencanaan pengadaan buku/bahan pustaka
b. Pengurusan pelayanaan perpustakaan
c. Perencanaan pengembangan perpustakaan
d. Pemeliharaan dan perbaikan buku-buku/bahan pustaka
e. Inventarisasi dan pengadiministrasian buku-buku/bahan pustaka
f. Melakukan layanan bagi siswa, guru dan tenaga kependidikan lainya
g. Penyimpanan buku dan perpustakaan
h. Menyusun tata tertib perpustakaan
i. Menyusun laporan pelaksanaan kegiatan perpustakaan secara berkala

5. Wali Kelas

Tugas Utama:

a. Pengelolaan kelas
b. Penyelenggaraan administrasi kelas meliputi : daftar pelajaran kelas, daftar piket kelas, buku absensi siswa, buku kegiatan pembelajaraan/buku kelas, tata tertib siswa, pembuatan statistic bulanan siswa
c. Pengisian daftar kumpulan nilai
d. Pembuatan catatan khusus tentang siswa
e. Pencatatan mutase siswa
f. Pengisian buku laporan penilaian hasil belajar
g. Pembagian buku laporan hasil belajar

6. Guru Mata Pelajaran

Tugas Utama:

a. Membuat perangkat pembelajaran

b. Melaksanakan kegiatan pembelajaran
c. Melaksanakan kegiatan penilaian proses belajar, ulangan harian, ulangan umum, ujian akhir
d. Melaksanakan analisis hasil ulangan harian
e. Menyusun dan melaksanakan program perbaikan dan pengayaan
f. Mengisi daftar nilai siswa
g. Melaksanakan kegiatan membimbing (pengimbasan pengetahuan) kepada guru lain dalam proses kegiatan belajar mengajar
h. Membuat alat pelajaran/alat peraga
i. Menumbuh kembangkan sikap menghargai karya seni
j. Mengikuti kegiatan pengembangan dan pemasyarakatan kurikulum
k. Melaksanakan tugas tertentu di sekolah
l. Mengadakan pengembangan program pengajaran yang menjadi tanggung jawabnya
m. Membuat catatan tentang kemajuan hasil belajar
n. Mengisi dan meneliti daftar hadir siswa sebelum memulai pelajaran
o. Mengatur keberhasilan ruang kelas dan praktikum
p. Mengumpulkan dan menghitung angka kredit untuk kenaikan perangkatnya

2.2. Landasan Teori

Landasan teori merupakan penjelasan konsep dasar dan teori-teori yang berkaitan dalam pembangunan aplikasi absensi berbasis pendeteksi wajah. Bagian teori yang terkait dalam pembanguan aplikasi adalah Bahasa pemograman C# untuk pembangunan *Desktop base*, dengan *EmguCV,* MySQL, dan UML.

2.2.1. Dasar Dasar Pengolahan Citra

Pengolahan citra berkaitan dengan manipulasi dan analisis dari citra digital. Citra-citra yang dibentuk dapat dimanipulasi sedemikian rupa hingga sesuai dengan keinginan, sebagai contoh proses *editing* foto yang biasanya dilakukan oleh di studio. Bisa juga citra yang ada dianalisis sesuai dengan tujuan yang

diharapkan. Dalam tugas akhir pengolahan citra digital yang dianalisis ada bagaimana cara mendeteksi wajah dalam prosesnya ada cakupan materi yang terkait seperti persepsi pandangan manusia, bagaimana pengambilan citra, dan kuantisasi citra.

a. Persepsi Pandangan Manusia

Persepsi pandangan manusia mempunyai peranan penting dalam pemilihan teknologi. Walaupun citra digital dibangun dari bentuk formulasi matematika dan probabilistik namun dalam prakteknya intuisi dan analisis manusia tetap diperlukan[6].

b. Pengambilan Citra

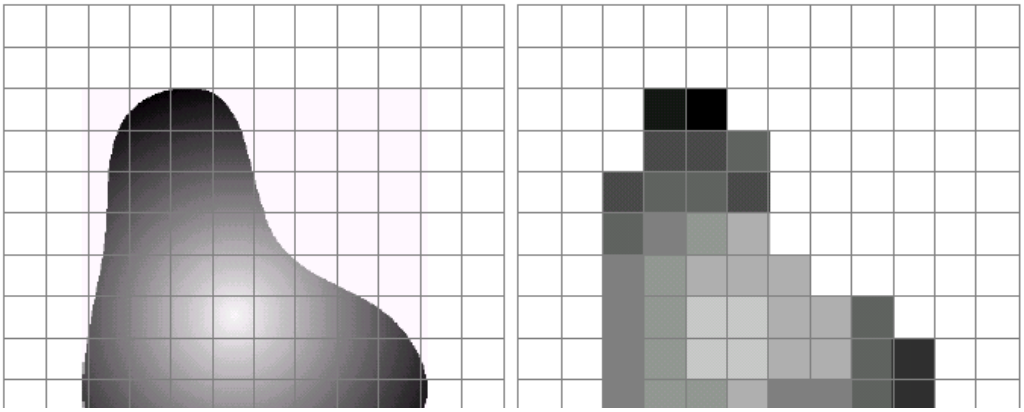
Pengambilan citra dibentuk dari kombinasi sumber iluminasi (pencahayaan) dan pencerminan atau penyerapan sumber oleh bagian citra yang diambil. Pencahayaan yang baik dapat menghasilkan citra yang bagus dengan begitu jarak pengambilan gambar juga berpengaruh terhadap citra yang dibentuk[6]

Pengambilan jarak yang baik adalah dengan mempertimbangkan estetika objek dan latar belakangnya, untuk itu diperlukan intuisi dari pengambil gambar. Terlalu jauh objek yang diambil maka semakin kabur citra tersebut, sebaliknya jika objek citra terlalu dekat maka objek pengenalan citra tidak dapat dikenali.

c. Sampling Citra dan Kuantisasi

Output dari citra digital berupa tegangan kontinu dimana amplitude dan spasial dihubungkan ke perwujudan fisik yang ditangkap. Untuk membuat citra digital diperlukan proses pengkonversian dari kontinu ke banyak digital, yaitu sampling dan kuantisasi

Sampling adalah pendigitalisasian nilai kordinat dimana kordinat *continuous image f(x,y)* di kordinat tertentu diambil untuk mempresentasikan citra. Sedangkan kuantisasi adalah pendigitalisasian nilai amplitude atau nilai dari tingkat keabuan yang dimana nilai tersebut dibuat ke bentuk diskritnya. Gabungan dari *sampling* dan kuantisasi ini disebut citra digital yang nantinya dapat diproses[6]. Hasil sampling dan kuantisasi dapat dilihat pada gambar 2.3



Gambar 2.3 (kiri) citra kontinu; (kanan) hasil sampling dan kuantisasi

2.2.2. Pendekteksi Wajah

Pendeteksi wajah adalah suatu teknologi computer untuk mendeteksi wajah manusia dengan cara menentukan letak dan ukuran wajah manusia di dalam citra digital. Teknologi ini dapat mendeteksi wajah melalui ciri/sifat wajah dan tidak mempedulikan hal-hal lainya, seperti pohon dan badan manusia itu sendiri.

Bidang-bidang penelitian yang juga berkaitan dengan pemrosesan wajah (*face processing)* adalah autentikasi wajah (*face authentication),* lokalisasi wajah (*face localization),* penjejakan wajah (*facetracking),* dan pengenalan ekspresi wajah (*face expression recognition)*[7].

Pendeteksi wajah merupakan salah satu tahap awal (*preprocessing)* yang sangat penting sebelum dilakukan proses pengenalan wajah (*face recognition).* Pendeteksi wajah dapat juga diartikan dengan deteksi benda yang spesifik. Dalam kasus ini benda yang dideteksi secara spesifik atau berupa wajah manusia yang

2.2.3. Metode Pendekteksi Wajah

Untuk menentukan dan mendeteksi wajah, diperlukan metode-metode yang secara nalar bisa dipahami. Dalam deteksi wajah ada empat istilah yang sering dipakai antara lain :

1. *True positive* = Wajah terdeteksi benar sebagai wajah
2. *True Negative* = Wajah terdeteksi benar sebagai bukan wajah
3. *False positive* = Wajah tidak terdeteksi sebagai wajah
4. *False negative* = Bukan wajah terdeteksi wajah

Metode pendeteksi wajah secara garis besar dibagi menjadi empat kategori berdasarkan pendekatan metode yang digunakan, antara lain :

a. *Knowledge-based Methods*

Pada metode ini, deteksi wajah dikembangkan berdasarkan *rules* atau aturan yang simple tentang pengetahuan repesentasi wajah yang dimiliki ilmuan. Aturan yang digunakan adalah dengan cara menggabungkan fitur-fitur yang ada pada wajah. Sebagai contoh, wajah adalah bagian citra yang memiliki mata yang simetris satu sama lain, memiliki hidung dan mulut. Hubungan antar fitur-fitur ini direpresentasikan *relative distance and positions.* Fitur wajah diinputan citra diekstrak terebih dahulu, dan kandidat wajah diidentifikasi berdasarkan peraturan yang telah dibuat. Proses verifikasi dilakukan untuk mengurangi false detection[7].

Ada satu masalah dalam deteksi wajah menggunakan metode ini, yaitu sulitnya menterjemahkan pengetahuan dengan representasi wajah kedalam sebuah aturan yang terdefinisi dengan baik. Jika aturan yang dibuat detail, memungkinkan terjadiya *true negative*, dimana wajah tidak terdeteksi oleh system[7].

b. *Feature Invariant Approaches*

Metode ini merupakan lawan dari metode *Knowledge-based* yang bertujuan untuk menemukan struktur fitur wajah meskipun pose, sudut pandang, atau pencahayaan beragam, dan dengan temuan ini dapat mengetahui letak wajah. Ilmuan berasumsi berdasarkan cara manusia dapat mendeteksi wajah dan objek. Dari sana ciri atau fitur yang sama akan di dapat dari semua kerahaman data yang ada.

Ada banyak metode yang diajukan untuk mengetahui fitur wajah dan menyimpulkan keberadaan wajah pada suatu citra. Fitur wajah seperti alis, mata, hidung, mulut, dan garis pada rambut secara umum diekstrak menggunakan deteksi tepi. Berdasarkan pengekstrakan fitur, model statistika dibangun untuk menskripsikan hubungan dan memverifikasi keberadaan wajah.

Ada masalah pada metode ini, jika fitur pada citra sangat rusak dikarenakan adanya masalah pada pencahayaan, *noise,* atau terjadi kesalahan pada pengambilan citra. Cakupan fitur pada wajah melemah, dengan itu pula bayangan dapat mempengaruhi deteksi tepi yang menyebabkan persepsi algoritma menjadi tidak efisien[7].

c. *Templeate Matching Methods*

Metode *Templeate Matching* merupakan metode dengan beberapa standar pola wajah yang tersimpan untuk mendeskripsikan wajah secara keseluruhan atau bagian fitur wajah tertentu saja. Kolerasi antara citra masukan dan pola yang tersimpan adalah perhitungan deteksi yang dilakukan. Pendekatan ini mempunyai kelebihan karena mudah untuk diimplementasikan

d. *Apperance-based Methods*

Metode ini lawan dari metode *Templeate Matching*, model matematis didapat dari kumpulan citra latih yang mempresentasikan varisasi tampilan pada wajah. Model ini digunakan sebagai deteksi. Secara umum, metode ini menggunakan pendekatan secara analisis statistika dan *machine learning* untuk menemukan fitur yang membedakan antara wajah dan bukan wajah. Salah satunya adalah metode Viola Jones.

Metode ini masih dikembangkan dalam masalah deteksi objek secara luas. Metode ini juga sering termasuk kedalam *project* yang diadakan oleh google dan Microsoft karena memiliki kecepatan yang lebih cepat dibandingkan metode lainya tanpa mengurangi secara signifikan akurasi deteksi. Oleh karena itu metode ini banyaj dipakai untuk kasus deteksi.

Berikut ini adalah tabel keakuratan deteksi wajah Viola Jones dengan metode yang di-*publish* peneliti lainya. Dimana *false detection* maksudnya adalah terdeteksi wajah oleh sistem tapi sebenarnya tidak atau disebut juga *false negative*.

Tabel 2. 1 Tabel akurasi berdasarkan *false detection*

Detector	*False Detector*							
	10	31	50	65	78	95	167	422
Viola Jones	76,1%	88,4%	91,4%	92,0%	92,1%	92,9%	93.9%	94,1%
Viola Jones (voting)	81,1%	89,7	92,1%	93,1%	93,1%	93,2%	93,7%	
Rowley-Baluja-Kanade	-	-	-	-	-	89,2%	90,1%	89,9%
Schneiderman-Kanade				94,4%	-	-	-	-
Roth-Yang-Ahuja	-	-	-	-	94,8%	-	-	-

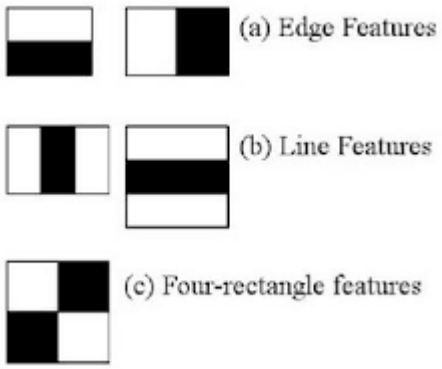
Dari tabel diatas menunjukan bahwa semakin banyak *false detector* yang digunakan maka semakin besar akurasi wajah

2.2.3.1. Viola Jones

Deteksi wajah dengan *Viola Jones* menggunakan fitur simple *haar-like* yang mengevaluasi dengan cepat representasi citra yang baru. *Viola Jones* meng-generate kumpulan fitur dengan citra integral dan *boosting algoritm* untuk mengurangi kompleksitas waktu. Sebelum dimasukan kedalam sistem terlebih dahulu citra dicari nilai keabu-abuanya (*grayscale)*. Secara umum metode Viola Jones memiliki empat dasar proses, yaitu :

a. *Haar-like Feature*

Klasifikasi citra dilakukan berdasarkan nilai dari sebuah fitur. Hal ini bertujuan untuk memisahkan citra yang tidak diperlukan. Dalam kasus ini, background tidak akan dihitung. Terdapat 3 jenis fitur berdasarkan jumlah persegi panjang (terang dan gelap) yang terdapat di dalamnya, yaitu : dua, tiga, empat persegi panjang seperti yang dapat dilihat pada gambar 2.4



Gambar 2.4 *Fitur Haar-like*

Pada gambar 2.4 dapat dilihat bahwa gambar 1 terdiri dari dua persegi, gambar 2 terdiri dari tiga persegi, gambar 3 terdiri dari dua persegi dengan salah satu persegi terletak di dalam*center* persegi lain, dan gambar 4 terdiri dari empat persegi. Haar-Like feature memproses gambar dalam wilayah kotak-kotak yang berisi beberapa pixel dari sebuah bagian gambar. Kemudian pixel-pixel dalam satu wilayah tersebut dijumlahkan dan dilakukan proses perhitungan sehingga didapatkan perbedaan dalam setiap wilayah kotak-kotak tersebut. Perbedaan inilah yang dapat dijadikan sebuah kode untuk menandai wilayah tersebut sehingga kita dapat memproses bagian gambar yang kita inginkan[8].

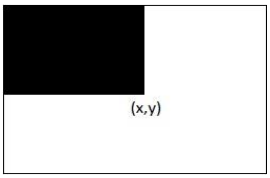
f(x)=Sumblack rectangle (pixel gray level) – Sumwhite rectangle (pixel gray level (2.1)

Dengan begitu akan dapat ditentukan tingkat *luminance* dari citra yang akan dideteksi dan dapat dibedakan mana bagian citra yang mencirikan wajah manusia. Untuk mempermudah proses perhitungan nilai fitur, metode *Viola Jones* menggunakan media berupa citra integral.

b. Citra Integral

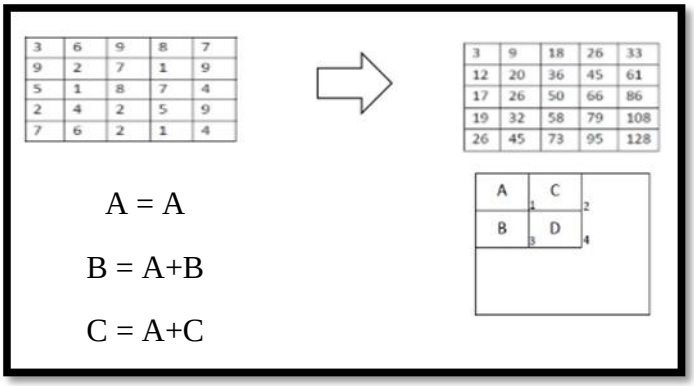
Citra integral adalah struktur data dan algoritma yang menjumlahkan nilai-nilai dalam *subset* di matriks citra, dimana nilai piksel dari suatu titik (x,y) pada citra merupakan hasil penjumlahan dari seluruh piksel yang ada disebelah kiri dan atas titik tersebut. Citra integral sangat membantu dalam perhitungan fitur *haar*.

Dengan menggunakan citra integral, perhitungan fitur haarlike dapat dilakkan dengan sangat cepat.



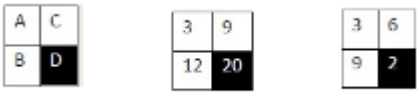
Gambar 2.5 Nilai piksel pada titik (x,y)

Misalnya, terdapat citra input berukuran 5x5 dengan nilai masing-masing piksel sebagai berikut :



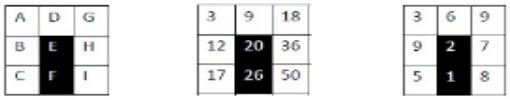
Gambar 2.6 Ilustrasi perhitungan citra integral

Pada citra integral adalah D = 4+1-(2+3). Contoh 1 (dari contoh citra integral di atas) :



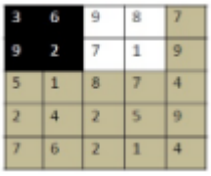
Gambar 2.7 Contoh 1

Nilai piksel pada daerah hitam = D+A-(B+C) = 20+3-(12+9) = 2. Nilai piksel pada daerah hitam adalah 2. Selanjutya contoh ke 2 dijelaskan pada gambar 2.8.



Gambar 2.8 Contoh 2

Pada gambar 2.8 jumlah nilai piksel daerah hitam = F+A-(C+D) = 26+3-(17+9) = 3. Jumlah nilai piksel pada daerah hitam 2+1 = 3. Perbandingan perhitungan nilai *fitur haar* dengan citra integral dan tanpa citra integral, sebagai berikut:



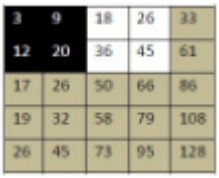
Gambar 2.9 Tanpa citra integral

f(x)=Sumblack rectangle (pixel gray level) – Sumwhite rectangle (pixel gray level)

= (3+6+9+2) - (9+8+7+1)

= 20-25

= 5



Gambar 2.10 Dengan citra integral

f(x)=Sumblack rectangle (pixel gray level) – Sumwhite rectangle (pixel gray level)

= 20+0-(0+0) – 45+0-(20+0)

= 20 – 25

= 5

Dari contoh-contoh perbandingan perhitungan fitur diatas, dapat dilihat adanya perbedaan efisiensi perhitungan dengan dan tanpa integral. Tanpa menggunakan citra integral, harus dilakukan perhitungan terhadap seluruh nilai piksel yang terdapat dalam fitur, sedangkan dengan menggunakan citra integral, hanya perlu melakukan perhitungan terhadap empat titik piksel. Perbedaan kecepataan ini akan semakin terlihat apabila fitur yang digunakan semakin besar, dan semakin banyak. Pada sebuah citra, jumlah fitur yang digunakan semakin besar, dan semakin banyak dari pada jumlah piksel pada citra tersebut, sehingga

penggunaan citra integral dalam perhitungan nilai fitur haar akan sangat membantu mempercepat proses perhitungan. Inilah alasan mengapa *Viola Jones* dapat mengevaluasi setiap fitur dengan sangat cepat.

Permasalahan yang yang terdapat dalam perhitungan fitur ini adalah *Viola jones* memiliki 160.000 jenis fitur yang berbeda. Jumlah ini terlalu besar sehingga tidak mungkin dilakukan perhitungan untuk semua fitur. Hanya fitur-fitur tertentu saja yang dipilih untuk diikutsertakan. Pemilihan fitur-fitur ini dilakukan menggunakan algoritma *ada-boost*

c. Ada-Boost

AdaBoost merupakan tahap ketiga dalam metode *Viola Jones*. Algoritma *AdaBoost* berfungsi untuk melakukan pemilihan fitur-fitur dalam jumlah banyak dengan hanya memilih fitur-fitur tertentu. Dalam prakteknya tidak satupun yang mampu melakukan pengklasifikasian dengan *error* yang kecil. Algoritma *AdaBoost* berfungsi mencari fitur-fitur yang memiliki tingkat pembeda yang tinggi. Hal ini dilakukan dengan mengevaluasi setiap fitur terhadap terbesar antara wajah dan non wajah dianggap sebagai fitur terbaik.

Inisialisasi bobot $w_{1,yi} = \frac{1}{2l}$ (2.2)

Dengan l merupakan jumlah citra positif dengan $_{yi}$ = 1 untuk gambar positif

Untuk t = 1,….,T, dimana t adalah iterasi ke t untuk gambar positif

h_t (x) merupakan nilai fitur gambar positif

jika posisi h(x) = ketentuan 1 maka citra tersebut merupakan objek

jika posisi h(x) = ketentuan 0 maka citra tersebut merupakan objek

H(x) = klasifikasi yang menyatakan objek atau bukan

$_{at}$ = tingkat pembelajaran positif

β_t = Nilai bobot setelah error rate gambar positif

1. Pilih min *error classifier* h_t *:*

   $\varepsilon t = (\sum_t^T w_{t,i}) \mid h_t (x)\text{-}y_i$ (2.3)

2. Perbaharui bobot :

   $w_t +_{1,i} = w_{t,i}\, \beta_t^{1-\varepsilon i}$ (2.4)

Dimana e_i = 0 jika x_i diklasifikasi sebagai wajah, sebaliknya e_i = 1 jika x_i diklasifikasi bukan wajah maka diatur bobot ke bawah:

$$\beta t = \frac{\varepsilon t}{1-\varepsilon t} \quad (2.5)$$

Terakhir menggabungkan secara *linier weak classifier* yang dibentuk, *final (strong) classifier* :

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} aj, hj(x) \geq \frac{1}{2} \sum_{t=1}^{T} at \\ 0 & \text{Otherwise} \end{cases} \quad (2.6)$$

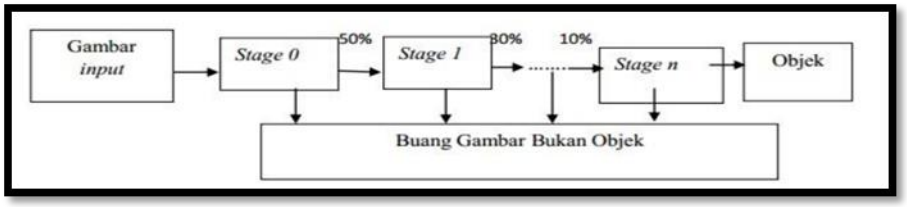
dimana $a_t = \log \frac{1}{\beta t}$, (2.7)

Boosting merupakan meta-algoritma dalam *machine learning* untuk melakukan *supervised learning*. *boosting* merupakan sekumpulan *weak learner* menciptakan satu kesatuan *strong learner*. *weak learner* adalah *classifier* yang hanya memiliki sedikit korelasi dengan klasifikasi yang sebenarnya, sementara *strong learner* adalah *classifier* yang memiliki kolerasi kuat dengan klasifikasi yang sebenarnya.

d. Cascade Classifier

Karakteristik metode *Viola Jones* adalah adanya klasifikasi bertingkat (*cascade classifier)*. Klasifikasi pada algoritma ini terdiri dari beberapa tingkatan dan setiap tingkatan mengeluarkan subcitra yang diyakini bukan wajah. Hal ini dilakukan karena lebih mudah untuk menilai subcitra yang bukan wajah daripada menilai subcitra tersebut berisi wajah[8].

Setiap *subwindows* dibandingkan dengan setiap fitur disetiap *stage*. Jika tidak mencapai target maka *subwindows* akan bergerak ke *subwindow* berikutnya dan melakukan perhitungan yang sama dengan proses sebelumnya, dapat dlihat pada gambar 2.11.



Gambar 2.11 Ilustrasi *cascade classifier*

Dari gambar 2.11 pada klasifikasi tingkat pertama, tiap subcitra pada *subwindows* diklasifikasi menggunakan beberapa fitur *haar-like*. Jika subcitra mencapai *threshold* maka proses berlanjut ke stage berikutnya. Tetapi jika tidak mencapai *threshold* maka *subwindows* ditolak dan proses berlanjut ke subcitra berikutnya. Pada proses selanjutnya didapat hasil yaitu *subwindows* yang terdeteksi sebagai wajah dan berlanjut ke subcitra berikutnya. Sampai pada akhirnya didapat kandidat kuat yang terdeteksi sebagai wajah[8].

2.2.4. Metode Pengenalan Wajah

Pengenalan wajah adalah suatu metode pengenalan yang berorentasi pada wajah. Pengenalan ini dapat dibagi menjadi dua bagian yaitu : dikenali atau tidak dikenali, setelah dilakukan perbandingan dengan pola sebelumnya disimpan di dalam *database*. Metode ini juga harus mampu mengenali objek bukan wajah. Perhitungan model pengenalan wajah memiliki beberapa masalah. Kesulitan muncul ketika wajah dipresentasikan dalam suatu pola yang berisi informasi unik yang membedakan dengan wajah yang lain[9].

Secara umum sistem pengenalan citra wajah dibagi menjadi dua jenis, yaitu sistem *feature based* dan sistem *image based*. Pada sisstem pertama digunakan fitur yang diekstraksi dari komponen citra wajah (mata, hidung, mulut, dll) yang kemudian hubungan antar fitur-fitur tersebut dimodelkan secara geometris. Sedangkan sistem kedua menggunakan informasi mentah dari piksel citra yang kemudian dipresentasikan dalam metode tertentu.

Proses pengenalan wajah yang dilakukan oleh computer tidak semudah dan secepat proses pengenalan yang dilakukan dengan cepat tanpa rasa harus berfikir. Manuskia juga tidak terpengaruh oleh orientasi wajah tersebut, misalnya wajah orang tersebut dengan keadaan agak menoleh, merunduk dan mengadah asalkan ada batas-batas yang masih bias dilihat. Sedangkan computer selain lamban dalam pengenalan juga kesulitan pada orientasi wajah yang berlainan , pencahayaan, latar belakang yang berbeda, potongan rambut, kumis atau jenggot, berkacamata atau tidak dan sebagainya

2.2.4.1 *Eigenface*

Algoritma *Eigenface* secara keseluruhan cukup sederhana. Image matriks (). Cari nilai rata-rata (Ψ) dan gunakan untuk mengekstraksi *eigenvector*(v) dan *eigenvalue*(λ) dari himpunan matriks. Gunakan nilai eigenvector untuk mendapatkan nilai *Eigenface* dari *image*. Apabila ada sebuah *image* baru atau *test face* () yang ingin dikenali, proses yang sama juga diberlakukan untuk *image* (), untuk mengektraksi *eigenvector* (v) dan *eigenvalue* (λ), kemudian cari nilai *Eigenface* dari image *test face* (). Setelah itu barulah *image* baru () memasuki tahapan pengenalan dengan menggunakan metode *euclidean distance*. Metode *euclidean distance* digunakan untuk mencari jarak dengan data fitur yang telah didapat, dan jarak terkecil adalah hasilnya.[9]

1. Langkah pertama menentukan himpunan (S) pada seluruh *training image(Γ1, Γ2, …, Γn)* dimana setiap citra wajah adalah 10 x 10 pixel :

   $$\mathbf{S = (\Gamma_1, \Gamma_2, \ldots, \Gamma_n)} \quad (2.8)$$

2. Langkah kedua adalah ambil nilai rata-rata atau *mean (Ψ)*

   $$\Psi = \frac{1}{M}\sum_{n=1}^{M} \Gamma_n \quad (2.9)$$

   Keterangan : M = jumlah baris pixel image

3. Langkah ketiga kemudian cari selisih (Φ) antara training image (Γi) dengan nilai tengah (Ψ)

   Rumus perhitungannya : $\mathbf{\Phi i = \Gamma 1 - \Psi}$ (2.10)

4. Langkah keempat adalah menghitung nilai matriks kovarian (C)

   $$\mathbf{L = A^T A} \qquad \mathbf{L = \Phi_n \Phi_m^T} \quad (2.11)$$

   Keterangan : L = Matriks Kovarian, A= Selisih Nilai (Φ)

5. Langkah kelima menghitung eigenvalue (λ) dan eigenvector (*v*) dari matriks kovarian (C).

$$C \times v_i = \lambda_i \times v_i \quad (2.12)$$

Cari nilai eigenvalue (λ) dan eigenvector (*v*)

L x v = λ x v

L x v = λ x v

(L – λi) = 0 atau (λ – L) = 0

6. Langkah keenam, setelah eigenvector (v) diperoleh, maka eigenface (μ) dpat dicari dengan :

$$\mu_i = \sum_{k=1}^{M} v_{lk} \Phi_k$$

μ i = v x Φ I (2.13)

7. Tahap pengenalan

Sebuah *image* wajah baru atau test face (Γ_{new}) akan dicoba untuk dikenali, pertama terapkan cara pada tahapan pertama perhitungan eigenface untuk mendapatkan nilai eigenface dari *image* tersebut

$$\mu_{new} = v \,.\, (\Gamma_{new} - \Psi)$$

$$\Omega = [\mu_1, \mu_2, \ldots, \mu_3] \quad (2.14)$$

2.2.5. Microsoft Visual C#

Microsoft Visual C-Sharp atau yang lebih dikenal dengan Visual C# adalah sebuah bahasa yang tidak diragukan lagi dalam proses pengembangan aplikasi bebasis *.NET Framework*, dimana C# bebas dari masalah kompabilitas dilengkapi dengan berbagai fitur yang sebagian besar merupakan fitur baru, menarik, dan tentu saja menjanjikan.

Visual C# dibuat berdasarkan pemrograman C# yang merupakan bahasa pemrograman berorientasi objek dan mempunyai banyak kesamaan dengan C++, Java, dan VB. C# pada faktanya merupakan kombinasi antara efisiensi

pemrograman C++, kesederhanaan pemrograman Java, dan penyederhanaan dari pemrograman Visual Basic.

Seperti pemrograman Java, C# juga tidak memperbolehkan multiple inheritance atau pengunaan pointer (pada safe/managed code), tetapi C# menyediakan *garbage memory colletion* pada saat runtime dan pada saat pengecekan akses memori. Meskipun bertentangan dengan pemrograman Java, C# tetap mempertahankan operasi unik yang terdapat pada bahasa pemrograman C++ seperti *overloading, enumerations, pre-processor directive, pointer (pada unmanaged/unsafe code),* dan fungsi *pointer*. Seperti halnya Visual Basic, bahasa pemrograman C# juga dilengkapi dengan *properties*.

Sebagai tambahan, bahasa pemrograman C# juga datang dengan beberapa fitur baru dan sangat menarik seperti *reflections, attributes, marshalling, remote, threads, streams data acces* dengan ADO.NET, dan masih banyak lagi.[10]

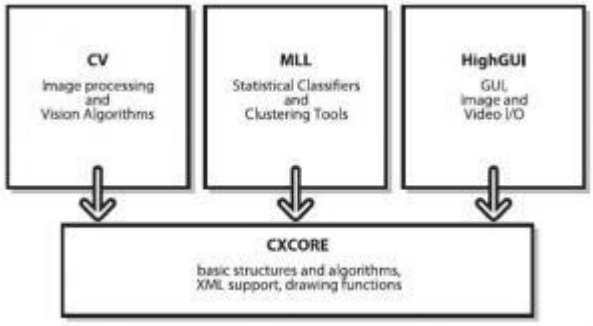
2.2.6. Open CV Library

OpenCV (*Open Computer Vision*) adalah sebuah API (*Application Programming Interface*) Library yang sudah sangat familiar pada Pengolahan Citra *Computer Vision*. *Computer Vision* itu sendiri adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (*Image Processing*) yang memungkinkan komputer dapat melihat seperti manusia. Dengan *vision* tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek. Beberapa pengimplementasian dari *Computer Vision* adalah *Face Recognition, Face Detection, Face/Object Tracking, Road Tracking, dll*. OpenCVadalah library *Open Source* untuk *Computer Vision* untuk C/C++, OpenCV didesain untuk aplikasi real-time, memiliki fungsi-fungsi akuisisi yang baik untuk *image/video*.[11]

OpenCV sendiri terdiri dari 5 library, yaitu :

1. CV : untuk algoritma *Image processing dan Vision.*
2. ML : untuk *machine learning library*
3. Highgui : untuk GUI, *Image* dan *Video* I/O.
4. CXCORE : untuk struktur data, *support XML* dan fungsi-fungsi grafis.

Struktur dan Konten OpenCV :



Gambar 2.12 Struktur dan Konten OpenCV

2.2.7. MySQL

MySQL adalah sebuah perangkat lunak system manajemen basis data, SQL (*Database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah : David Axmark, Allan Larsson dan Michael "Monty" Widenius.[12]

MySQL memiliki beberapa keistimewaan, antara lain :

1. *Portabilitas*. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga,* dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.

3. *Multi-user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*, MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level *subnetmask,* nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protocolTCP/IP, Unis Socket (UNIX), atauNamed Pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatanyang ada disertakan petunjuk online.
13. Struktur tabel MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya semacam *Postgre*SQL ataupun *Oracle*.

2.2.8. Object Oriented Analysis and Design

Analisis dan Desain Berorientasi Objek (*Object Oriented Analysis and Design*) adalah cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep. Dasar pembuatannya sendiri adalah objek yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas. Alasan mengapa harus memakai metode beorientasi objek yaitu karena perangkat lunak itu sendiri yang bersifat dinamis, di mana hal ini disebabkan karena kebutuhan pengguna berubah dengan cepat. Selain itu bertujuan untuk menghilangkan kompleksitas transisi antar tahap pada pengembangan perangkat lunak, karena pada pendekatan berorientasi objek, notasi yang digunakan pada tahap analisis perancangan dan implementasi relatif sama tidak seperti pendekatan konvensional yang dikarenakan notasi yang digunakan pada tahap analisisnya berbeda-beda hal itu menyebabkan transisi antar tahap pengembangan menjadi kompleks. Di samping itu dengan pendekatan berorientasi objek membawa pengguna kepada abstraksi atau istilah yang lebih dekat dengan dunia nyata, karena di dunia nyata itu sendiri yang sering pengguna lihat adalah objeknya bukan fungsinya. Berbeda dengan pendekatan terstruktur yang hanya mendukung abstraksi pada level fungsional. Adapun dalam pemrograman berorientasi objek menekankan berbagai konsep seperti: *Class, Object, Abstract, Encapsulation, Polymorphism, Inheritance* dan tentunya UML (*Unified Modeling Language*). UML (*Unified Modeling Language*) sendiri merupakan salah satu alat bantu yang dapat digunakan dalam bahasa pemrograman berorientasi objek. Selain itu UML merupakan *standard modeling language* yang terdiri dari kumpulan-kumpulan diagram, dikembangkan untuk membantu para pengembang sistem (*developer*) dan software agar bisa menyelesaikan tugas-tugas seperti: Spesifikasi, Visualisasi, Desain Arsitektur, Konstruksi, Simulasi dan Testing. Dapat disimpulkan bahwa UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek (*Object Oriented Programmin*[13].

Dokumentasi UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek yang 4 diantaranya adalah[14]:

1. *Activity Diagram*

   Activity Diagram adalah sebuah tahapan yang lebih fokus kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Di mana biasanya dipakai pada *business modeling* untuk memperlihatkan urutan aktifitas proses bisnis. *Activity Diagram* ini sendiri memiliki struktur diagram yang mirip *flowchart* atau *data flow diagram* pada perancangan terstruktur. *Activity Diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram*.

2. *Use Case Diagram*

   Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Di dalam *use case diagram* ini sendiri lebih ditekankan kepada apa yang diperbuat sistem dan bagaimana sebuah sistem itu bekerja. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan bentuk dari sebuah pekerjaan tertentu, misalnya *login* ke dalam sistem, *posting*, dan sebagainya, sedangkan seorang aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu[14]. Adapun komponen-komponen dalam use case diagram diantaranya:

   a. Aktor

      Aktor merupakan suatu entitas yang berkaitan dengan sistem tapi bukan dari bagian dalam sistem itu sendiri. Aktor berada diluar sistem namun berkaitan erat dengan fungsionalitas didalamnya. Aktor dapat memiliki hubungan secara langsung terhadap fungsi utama baik terhadap salah satu atau semua fungsionalitas utama. Aktor juga dapat dibagi terhadap berbagai jenis atau tingkatan dengan cara digeneralisasi atau dispesifikasi tergantung kebutuhan sistemnya. Aktor biasanya dapat berupa pengguna atau database yang secara pandang berada dalam suatu ruang lingkup sistem tersebut.

b. Relasi

Relasi (*relationship*) digambarkan sebagai bentuk garis antara dua simbol dalam *use case* diagram. Relasi antara *actor* dan *use case* disebut juga dengan asosiasi (*association*). Asosiasi ini digunakan untuk menggambarkan bagaimana hubungan antara keduanya.

c. Include

Include, pemanggilan *use case* oleh *use case* lain atau untuk menggambarkan suatu *use case* termasuk di dalam *use case* lain (diharuskan). Contohnya adalah pemanggilan sebuah fungsi program. Digambarkan dengan garis lurus berpanah dengan tulisan <<include>>.

d. Extend

Extend, digunakan ketika hendak menggambarkan variasi pada kondisi perilaku normal dan menggunakan lebih banyak kontrol *form* dan mendeklarasikan ekstension pada *use case* utama. Atau dengan kata lain adalah perluasan dari use case lain jika syarat atau kondisi terpenuhi. Digambarkan dengan garis berpanah dengan tulisan <<extend>>.

3. *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram jenis ini memberikan kejelasan sejumlah objek dan pesan-pesan yang diletakkan diantaranya di dalam sebuah *use case*. Komponen utamanya adalah objek yang digambarkan dengan kotak segi empat atau bulat, *message* yang digambarkan dengan garis penuh, dan waktu yang ditunjukkan dengan *progress vertical*. Manfaat dari *sequence diagram* adalah memberikan gambaran detail dari setiap *use case diagram* yang dibuat sebelumnya.

4. *Class Diagram*

Class Diagram adalah sebuah *class* yang menggambarkan struktur dan penjelasan *class*, paket, dan objek serta hubungan satu sama lain. *Class* diagram juga menjelaskan hubungan antar *class* secara keseluruhan di dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan.

2.2.9. Perbandingan Metode Viola Jones

Perbandingan metode bertujuan untuk menunjukan keunggulan dari metode *viola jones* dengan membandingkan dengan metode lainya. Berikut perbandinganya.

1. Viola Jones vs Robert Cross

dengan membandingkan metode *viola jones* dengan *Robert cross* maka akan dapat diketahui cara kerja dan performasi dalam kecepatan dan ketepatan dari kedua metode tersebut sehinggga metode terbaik dapat digunakan untuk perangkat lunak pengenalan wajah. Dalam perbandingan ini yang digunakan sebagai parameter perbandingan nya yaitu berupa pencahayaan redup dan terang untuk mendapatkan waktu terbaik.[16]

A. Dalam Cahaya Redup

Tabel 2. 2 tabel *Viola Jones* dalan cahaya redup

No	Bentuk Wajah	Hasil	Waktu Terdeteksi
1	Muka Lonjong	terdeteksi	120
2		terdeteksi	123
3		terdeteksi	122
4		terdeteksi	123
5		terdeteksi	122
1	Muka Oval	terdeteksi	25
2		terdeteksi	29
3		terdeteksi	26
4		terdeteksi	27
5		terdeteksi	28
1	Muka Kotak	terdeteksi	120
2		terdeteksi	122
3		terdeteksi	124
4		terdeteksi	128
5		terdeteksi	127
1	Muka Bulat	terdeteksi	77
2		terdeteksi	27
3		terdeteksi	29
4		terdeteksi	28
5		terdeteksi	26

Tabel 2. 3 tabel *Robert Cross* dalan cahaya redup

No	Bentuk Wajah	Hasil	Waktu Terdeteksi
1	Muka Lonjong	tidak terdeteksi	-
2		tidak terdeteksi	-
3		tidak terdeteksi	-
4		tidak terdeteksi	-
5		tidak terdeteksi	-
1	Muka Oval	tidak terdeteksi	-
2		tidak terdeteksi	-
3		tidak terdeteksi	-
4		tidak terdeteksi	-
5		tidak terdeteksi	-
1	Muka Kotak	tidak terdeteksi	-
2		tidak terdeteksi	-
3		tidak terdeteksi	-
4		tidak terdeteksi	-
5		tidak terdeteksi	-
1	Muka Bulat	tidak terdeteksi	-
2		tidak terdeteksi	-
3		tidak terdeteksi	-
4		tidak terdeteksi	-
5		tidak terdeteksi	-

B. Dalam Cahaya Terang

Tabel 2. 4 tabel *Viola Jones* dalan cahaya terang

No	Bentuk Wajah	Hasil	Waktu Eksekusi
1	Muka Lonjong	terdeteksi	33
2		terdeteksi	32
3		terdeteksi	40
4		terdeteksi	35
5		terdeteksi	34
1	Muka Oval	terdeteksi	25
2		terdeteksi	35
3		terdeteksi	30
4		terdeteksi	49
5		terdeteksi	38
1	Muka Kotak	terdeteksi	120
2		terdeteksi	122
3		terdeteksi	124
4		terdeteksi	128
5		terdeteksi	127
1	Muka Bulat	terdeteksi	77
2		terdeteksi	27
3		terdeteksi	29
4		terdeteksi	28
5		terdeteksi	26

Tabel 2. 5 tabel *Robert Cross* dalan cahaya terang

No	Bentuk Wajah	Hasil	Waktu Eksekusi
1	Muka Lonjong	terdeteksi	102
2		terdeteksi	90
3		terdeteksi	106
4		terdeteksi	184
5		terdeteksi	177
1	Muka Oval	terdeteksi	163
2		terdeteksi	167
3		terdeteksi	159
4		terdeteksi	186
5		terdeteksi	171
1	Muka Kotak	terdeteksi	148
2		terdeteksi	146
3		terdeteksi	158
4		terdeteksi	131
5		terdeteksi	164
1	Muka Bulat	terdeteksi	167
2		terdeteksi	106
3		terdeteksi	148
4		terdeteksi	186
5		terdeteksi	184

C. Kesimpulan

hasil penelitian yang dilakukan oleh Ilham Andrian dengan judul "Perbandingan Metode *Viola Jones* dengan Metode *Robert Cross* Pada Sistem Pengenalan Wajah" bahwa *Metode Viola Jones* lebih cocok digunakan dalam cahaya redup dan terang, sedangkan metode *Robert cross* hanya dapat digunakan dalam cahaya terang.

2. Viola Jones vs Fourier Lucas Kanade

bertujuan untuk memperbaiki kinerja *viola jones* yang memiki kendala dalam pencahayaan dan pose dngan metode *fourier lucas kanade* dengan menggunakan keselarasan gambar. Dalam perbandingan ini objek yang dibandingkan berupa kecepatan dalam membaca gerak mulut (VSR)[17].

A. Kecepatan dalam VSR

Tabel 2. 6 tabel kecepatan membaca VSR

	VSR Acuracy	
	VJ	FLK
AVICAR	46,73	49.22
CUAVE	59.50	59.07