

## BAB II TINJAUAN PUSTAKA

### 2.1 Penelitian Terdahulu

Penelitian sebelumnya yang dilakukan oleh (Prasetyo dkk, 2016) yang berjudul **“Penerapan Algoritma Viola Jones Pada Deteksi Wajah”**. Merupakan sistem deteksi wajah berbasis komputer. Penelitian ini menerapkan metode *Viola Jones* untuk pendeteksian citra wajah. Hasil penelitian ini adalah menampilkan kotak kuning dan tulisan “wajah” jika dalam gambar tersebut menampilkan gambar wajah.

Penelitian diatas tidak menggunakan webcam, maka harus menginputkan gambar secara manual untuk mendeteksi ada tidaknya wajah pada gambar. Berbeda dengan penelitian yang penulis lakukan dimana sudah memakai webcam dan bisa mendeteksi kemiripan sampel wajah dengan wajah pengguna secara otomatis.

Penelitian sebelumnya yang dilakukan oleh (Susanto dkk, 2017) yang berjudul **“Sistem Keamanan Pintu Berbasis Pengenalan Wajah Menggunakan Metode Fisherface”**. Merupakan sistem keamanan pintu menggunakan pengenalan wajah berbasis komputer. Pada penelitian ini menerapkan metode pengenalan wajah (*face recognition*) *fisherface*. Hasil penelitian ini adalah Arduino akan mengontrol solenoid.

Penelitian diatas bahasa pemrograman yang dipakai adalah Bahasa C dan juga memerlukan arduino untuk mengontrol keluarannya. Berbeda dengan penelitian yang penulis lakukan dimana bahasa pemrogramannya yakni Bahasa Python dan keluarannya langsung diatur oleh *Raspberry Pi*.


Penelitian sebelumnya yang dilakukan oleh (Wijaya dkk, 2017) yang berjudul **“Implementasi Raspberry Pi Untuk Rancang Bangun Sistem Keamanan Pintu Ruang Server Dengan Pengenalan Wajah Menggunakan Metode Triangle Face”**. Merupakan sistem keamanan ruang server menggunakan webcam berbasis raspberry pi. Pada penelitian ini menerapkan metode *triangle*

*face*. Hasil penelitian ini adalah motor servo akan membuka dan mengunci pintu ruang server yang dikendalikan oleh *Raspberry Pi*.

Penelitian diatas menggunakan metode *triangle face* dan keluarannya menggunakan motor servo. Berbeda dengan penelitian yang penulis lakukan dimana metodenya yakni *haar cascade* dan keluarannya berupa *solenoid*.

## 2.2 Raspberry Pi 3

Menurut (Amalia, 2017) Raspberry Pi adalah sebuah single board computer yang dikembangkan oleh Yayasan Raspberry Pi di Inggris (UK) dengan maksud untuk memicu pengajaran ilmu komputer dasar di sekolah-sekolah. Jadi pada dasarnya Raspberry Pi berfungsi layaknya sebuah komputer namun ukurannya yang kecil maka disebut Single Board Computer. Raspberry Pi 3 diklaim lebih cepat. Raspberry Pi 3 menggunakan prosesor ARM Cortex-A53 dari Broadcom, dengan spesifikasi 64-bit Quad-Core dan berkecepatan 1,2 Ghz. Ini merupakan peningkatan dari prosesor 32-bit 900Mhz di versi sebelumnya. Pada gambar 2.1 merupakan spesifikasi dari Raspberry Pi 3 yang dipakai. Pada gambar 2.2 merupakan pin out dari Raspberry Pi 3.

	Raspberry Pi 3 Model B	Raspberry Pi Zero	Raspberry Pi 2 Model B	Raspberry Pi Model B+
Introduction Date	2/29/2016	11/25/2015	2/2/2015	7/14/2014
SoC	BCM2837	BCM2835	BCM2836	BCM2835
CPU	Quad Cortex A53 @ 1.2GHz	ARM11 @ 1GHz	Quad Cortex A7 @ 900MHz	ARM11 @ 700MHz
Instruction set	ARMv8-A	ARMv6	ARMv7-A	ARMv6
GPU	400MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV
RAM	1GB SDRAM	512 MB SDRAM	1GB SDRAM	512MB SDRAM
Storage	micro-SD	micro-SD	micro-SD	micro-SD
Ethernet	10/100	none	10/100	10/100
Wireless	802.11n / Bluetooth 4.0	none	none	none
Video Output	HDMI / Composite	HDMI / Composite	HDMI / Composite	HDMI / Composite
Audio Output	HDMI / Headphone	HDMI	HDMI / Headphone	HDMI / Headphone
GPIO	40	40	40	40
Price	\$35	\$5	\$35	\$35

Gambar 2.1 Spesifikasi Raspberry Pi 3

## Raspberry Pi 3 GPIO Header

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I <sup>2</sup> C)		DC Power 5v	04
05	GPIO03 (SCL1 , I <sup>2</sup> C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I <sup>2</sup> C ID EEPROM)		(I <sup>2</sup> C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Gambar 2.2 Pin Out Raspberry Pi 3

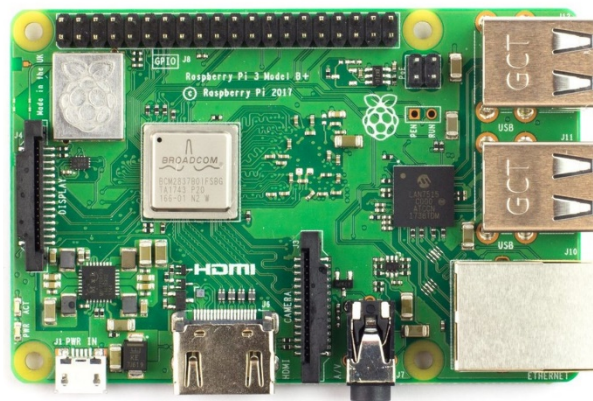
### 2.3 Raspberry Pi 3 Model B+

Menurut (Digiwrestore, 2019) Raspberry Pi 3 Model B+ adalah versi terbaru dari seri Raspberry Pi 3, Pi 3B+ memiliki bentuk dan ukuran yang identik dengan Pi 3B, Namun apabila dibandingkan dengan Raspberry Pi 3 Model B, Pi 3B+ mengalami peningkatan di beberapa bagian hardware, mulai dari prosesor 64-bit yang kini memiliki clockspeed maksimum 1.4 GHz (sebelumnya 1.2 GHz pada Pi 3B), memiliki Gigabit Ethernet (support PoE) yang tentu jauh lebih kencang dari versi sebelumnya, memiliki heatsink pada prosesor untuk distribusi panas yang lebih baik, serta mendukung dual band WLAN 5 GHz dan 2.4 GHz. Pada gambar 2.3 merupakan gambar dari Raspberry Pi 3 Model B+.

#### Spesifikasi:

- SoC: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit @ 1.4GHz
- RAM: 1 GB LPDDR2 SDRAM

- WLAN: 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN
- Bluetooth: Bluetooth 4.2, BLE
- Ethernet: Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- GPIO: 40 pin
- Display Output: HDMI
- USB: 4 USB 2.0
- Antarmuka: CSI, DSI, 3.5 mm audio jack
- Penyimpanan: Micro SD
- Catu Daya: 5V/2.5A DC



**Gambar 2.3** Raspberry Pi 3 Model B+.

## 2.4 Webcam

Menurut (Herlambang, 2016) Webcam merupakan perangkat pengcapture objek yang biasanya digunakan bersamaan dengan PC atau laptop. Webcam adalah singkatan dari web dan camera biasanya dipakai untuk keperluan konferensi video jarak jauh maupun berfungsi sebagai kamera pemantau. Webcam merupakan sebuah peripheral dalam bentuk kamera sebagai media pengambil citra atau gambar yang dikendalikan oleh sebuah komputer atau jaringan komputer. Umumnya data hasil perekaman langsung ditransfer kedalam media penyimpanan komputer sehingga tidak membutuhkan tempat penyimpanan data eksternal semacam kaset. Resolusi pada webcam dapat mempengaruhi kualitas gambar

yang ditampilkan. Webcam memiliki resolusi sekitar 352x288 piksel, 640x280 piksel, 1 Mega piksel, dan bahkan lebih besar. Pada gambar 2.4 merupakan sebuah webcam Logitech C525 yang akan digunakan untuk mendeteksi wajah.



**Gambar 2.4** Webcam Logitech C525

## 2.5 Power Supply

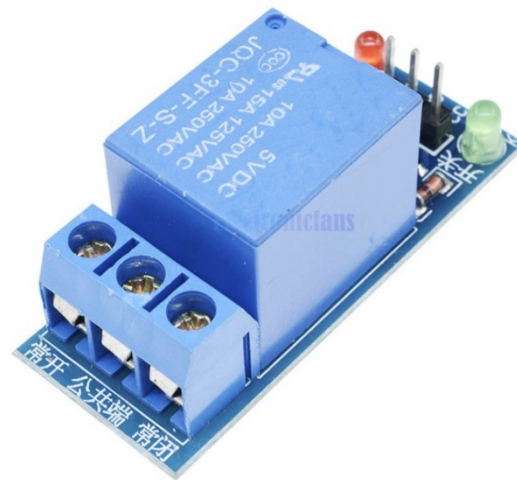
Menurut (Herlambang, 2016) Pencatu Daya (Inggris: *power supply*) adalah sebuah piranti elektronika yang berguna sebagai sumber daya untuk piranti lain, terutama daya listrik. Pada dasarnya pencatu daya bukanlah sebuah alat yang menghasilkan energi listrik saja, namun ada beberapa pencatu daya yang menghasilkan energi mekanik, dan energi yang lainnya. Pada sistem yang lebih besar, dimana tegangan dan daya yang diperlukan cukup besar, baterai sangat sulit digunakan dan sangat mahal. Oleh karena itu, diperlukan suatu peralatan yang lebih baik dan mudah digunakan sebagai sumber tegangan dan dapat digunakan sesuai dengan kebutuhan pemakaian. *Power supply* adalah salah satu bagian yang terpenting pada peralatan elektronika. Karena fungsinya sebagai sumber tegangan DC (*Direct Current*) untuk beroperasi. Pada gambar 2.5 merupakan power supply yang digunakan pada Raspberry Pi.



**Gambar 2.5** Power Supply

## 2.6 Relay

Menurut (Agustina, 2016) Relay adalah komponen elektronika berupa saklar elektronik yang digerakkan oleh arus listrik. Secara prinsip, relay merupakan tuas saklar dengan lilitan kawat pada batang besi (solenoid) di dekatnya. Ketika solenoid dialiri arus listrik, tuas akan tertarik karena adanya gaya magnet yang terjadi pada solenoid sehingga kontak saklar akan menutup. Pada saat arus dihentikan, gaya magnet akan hilang, tuas akan kembali ke posisi semula dan kontak saklar kembali terbuka. Relay biasanya digunakan untuk menggerakkan arus/tegangan yang besar (misalnya peralatan listrik 4 A/AC 220V) dengan memakai arus/tegangan yang kecil (misalnya 0.1 A/12 Volt DC). Pada gambar 2.6 merupakan relay yang digunakan pada Raspberry Pi.

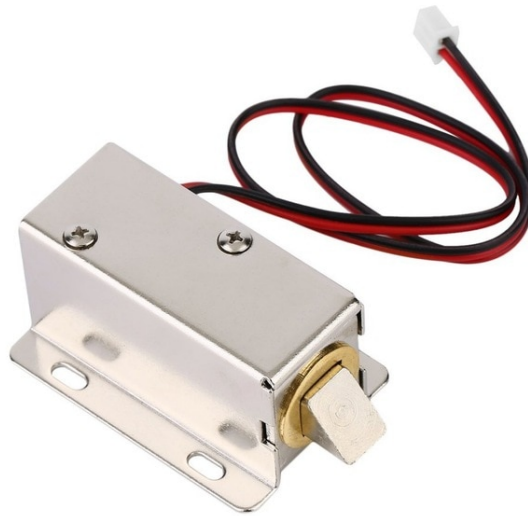


**Gambar 2.6** Relay

## 2.7 Solenoid Door Lock

Menurut (Earchaiiha, 2016) Solenoid adalah salah satu jenis kumparan terbuat dari kabel panjang yang dililitkan secara rapat dan dapat diasumsikan bahwa panjangnya jauh lebih besar daripada diameternya. Dalam kasus solenoid ideal, panjang kumparan adalah tak hingga dan dibangun dengan kabel yang saling berhimpit dalam lilitannya, dan medan magnet di dalamnya adalah seragam dan paralel terhadap sumbu solenoid. Solenoid DC beroperasi pada prinsip-prinsip dasar yang sama seperti motor DC. Perbedaan antara solenoid dan motor adalah solenoid bekerja maju atau mundur dan tidak bisa berputar. Di dalam solenoid terdapat kawat bermotor melingkar dengan dibuat khusus. Medan magnet akan tercipta ketika arus listrik mengalir melalui kawat ini. Poros pada solenoid adalah piston seperti silinder terbuat dari besi atau baja yang disebut plunger atau siput (setara dengan armature). Pada gambar 2.7 merupakan solenoid yang digunakan pada Raspberry Pi.





**Gambar 2.7** Solenoid

## 2.8 Bahasa Pemrograman Python

Menurut (Amalia, 2017) Python adalah sebuah bahasa pemrograman yang bisa digunakan pada beberapa *platform (multiplatform)*, dan berifat sumber perangkat bebas terbuka (*opensource*), pertama kali dikembangkan oleh Guido van Rossum pada tahun 1990 di CWI, Belanda. Bahasa ini dikategorikan sebagai bahasa tingkat tinggi (*very-high-level language*) dan merupakan bahasa berorientasi objek yang dinamis (*object-oriented-dynamic language*).

Hal utama yang membedakan *Python* dengan bahasa lain adalah dalam hal aturan penulisan kode program. *Python* memiliki aturan yang berbeda dengan bahasa lain, seperti indentasi, tipe data, *tuple*, dan *dictionary*. *Python* adalah bahasa pemrograman dinamis yang mendukung pemrograman berorientasi obyek. *Python* dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai sistem operasi seperti *Linux*, *Windows*, *Unix*, *Symbian* dan masih banyak lagi.



## 2.9 Citra

Menurut (Herlambang, 2016) Pengolahan citra (*image processing*) merupakan bidang yang berhubungan dengan proses transformasi citra (*image*) yang bertujuan untuk mendapatkan kualitas citra yang lebih baik.

Pengenalan pola (*pattern recognition*) merupakan bidang ilmu yang melakukan proses analisis gambar yang inputnya adalah gambar ataupun citra digital dan menghasilkan output suatu deskripsi dengan tujuan untuk mendapatkan informasi yang disampaikan oleh gambar atau citra, dengan kata lain meniru kemampuan manusia (otak manusia) dalam mengenali suatu objek atau pola tertentu.

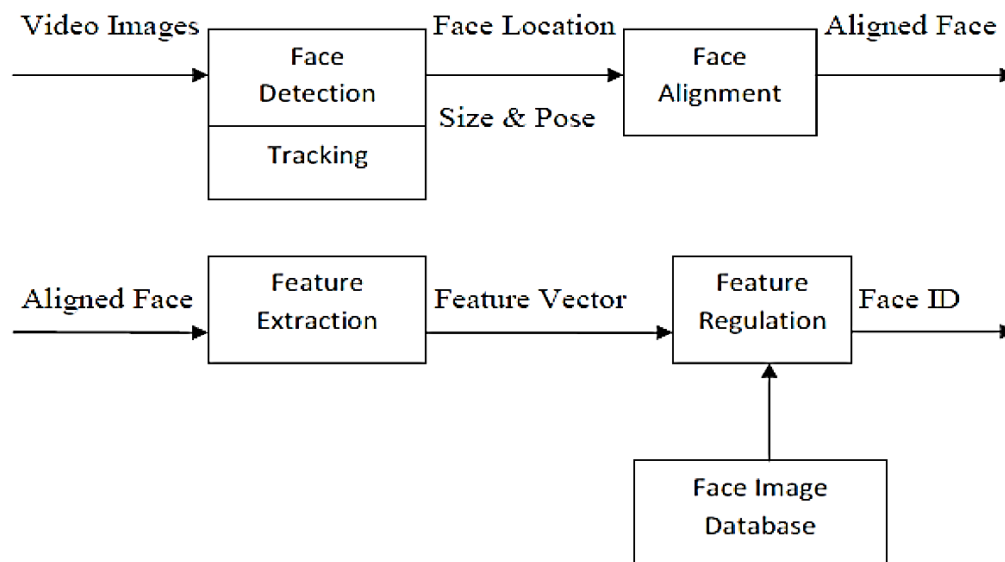
Identifikasi (pengenalan) wajah atau *face recognition* adalah sebuah tugas yang dikerjakan oleh manusia secara rutin dan mudah dalam kehidupan sehari-hari. Penelitian dan pengembangan ilmu pengenalan wajah berkembang secara otomatis atas dasar ketersediaan *desktop* kuat dan rendah biaya serta *embedded system* yang telah menciptakan minat yang sangat besar dalam pengolahan citra digital dan video. Motivasi penelitian dan pengembangan dari pengenalan wajah termasuk dalam lingkup otentikasi *biometric*, pengawasan, interaksi manusiakomputer, dan manajemen multimedia.

Pengenalan wajah adalah salah satu ilmu yang terdapat di dalam *computer vision*, di mana sebuah komputer dapat menganalisa suatu citra wajah yang terdapat di dalam sebuah gambar dan dapat menemukan identitas atau data diri dari citra wajah tersebut dengan membandingkan terhadap data-data citra wajah yang sudah disimpan sebelumnya di dalam *database*. Pada umumnya *face recognition* dilakukan dari sisi depan dengan pencahayaan yang merata ke seluruh wajah. Akan tetapi muncul beberapa permasalahan, seperti posisi wajah, skala atau jarak wajah, orientasi, umur, dan ekspresi wajah. Sistem *face recognition* pada umumnya mencakup empat modul utama, yaitu: deteksi, *alignment*, ekstraksi fitur dan pencocokan.

Proses lokalisasi dan normalisasi (deteksi wajah dan *alignment*) adalah langkah langkah sebelum proses pengenalan wajah (ekstraksi fitur wajah dan pencocokan) dilakukan. Deteksi wajah adalah langkah awal untuk melakukan

identifikasi wajah atau *face recognition*. Sebuah pendeteksi wajah yang ideal seharusnya mampu mengidentifikasi dan menemukan lokasi dan luas semua wajah yang ada di dalam sebuah gambar tanpa memperhatikan pose, skala, orientasi, umur, dan ekspresi.

Deteksi wajah melakukan segmentasi area citra wajah dengan bagian latar (*background*). Proses *alignment* bertujuan untuk memperoleh akurasi yang lebih baik dan tinggi untuk lokalisasi dan normalisasi citra wajah sebab deteksi wajah menyediakan batas lokasi dan skala dari setiap citra wajah yang dapat terdeteksi. Setelah sebuah wajah dilakukan normalisasi, ekstraksi fitur dilakukan untuk mengambil data yang efektif yang berguna untuk memisahkan antara citra citra wajah dan orang-orang yang berbeda satu sama lain dan cukup stabil untuk bermacam-macam *geometric* dan fotometrik. Pencocokan wajah dilakukan dengan cara melakukan pencocokan fitur yang telah diekstraksi dari citra wajah masukan dengan kumpulan data latihan dan uji coba citra wajah yang telah tersimpan sebagai *database* wajah. Pada gambar 2.8 merupakan blok diagram dari pengenalan wajah (*face recognition*).



**Gambar 2.8** Blok Diagram Face Recognition

Citra wajah direpresentasikan sebagai sebuah susunan *pixel* dengan dimensi tinggi. *Face recognition*, dan penelitian *computer vision* secara umum, telah mengamati pertumbuhan minat dalam teknik yang menerapkan aljabar dan fitur statistik untuk melakukan ekstraksi dan analisis kasus jenis ini. Analisa komputer untuk citra wajah dipengaruhi oleh sinyal visual (cahaya yang terpantul pada permukaan wajah) yang disimpan oleh sebuah sensor digital sebagai sebuah susunan dari nilai *pixel*. Nilai *pixel* ini menyimpan warna atau hanya intensitas cahaya. Array *pixel* dari citra wajah berukuran  $m \times n$  dapat disimpan dalam bentuk sebuah trik (contoh: vektor) di dalam sebuah ruang citra mendimensi dengan menulis nilai-nilai *pixel*-nya dalam urutan tetap. Masalah utama dari data multidimensi adalah dimensionalitasnya, jumlah koordinat yang diperlukan untuk menspesifikasikan sebuah titik data. Jumlah dimensionalitas ruang tersebut, yang dibuat dari representasi *pixel*  $m \times n$ , adalah jumlah yang sangat tinggi bahkan untuk sebuah ukuran citra wajah yang sangat sederhana. Metode pengenalan wajah untuk mengoperasikan representasi ini mengalami beberapa kesulitan.

Beberapa dari hambatan ini juga disebut *curse* dari dimensionalitas.

1. Menangani dimensionalitas yang tinggi, terutama dalam konteks pengenalan berdasarkan kecocokan memerlukan biaya yang mahal secara komputasi.
2. Untuk metode parametrik, jumlah parameter yang diperlukan untuk berkembang secara eksponensial berdasarkan dimensionalitasnya. Terkadang jumlahnya lebih banyak dari jumlah citra yang disediakan untuk latihan dan uji coba.
3. Untuk metode non-parametrik, kompleksitas dari sampel cukup tinggi.

*Face recognition* merupakan proses penganalisa karakteristik dari bentuk muka yang tidak berubah, seperti:

1. Bagian atas dari rongga mata
2. Area di sekitar tulang pipi
3. Sisi kiri dan kanan dari mulut

Kesulitan dalam pengenalan wajah sering ditemukan pada:

1. *Noise* dan *blur* yang disebabkan oleh ketidaksempurnaan kamera
2. Skala: Ukuran wajah terhadap citra
3. Perubahan bentuk: Posisi wajah, ekspresi, usia
4. Intensitas cahaya: Pencahayaan, efek pantulan sinar
5. Gangguan: kacamata, janggut, dan kumis

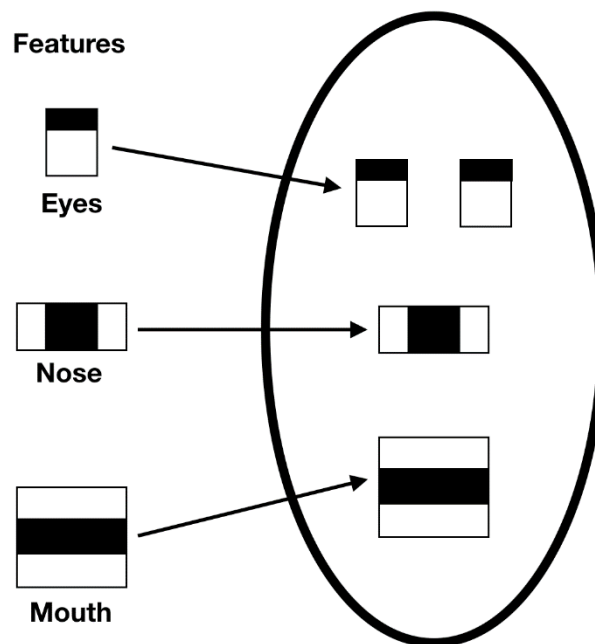
Pengenalan wajah bisa dilihat sebagai suatu cara untuk secara tepat mengenali citra dari sebuah wajah, dengan menggunakan data data dari wajah yang telah lebih dahulu dikenal. Pengenalan wajah memiliki semua hambatan dalam pengenalan yang berdasarkan pemrosesan citra. Dikarenakan citra yang digunakan dapat berubah-ubah secara drastis disebabkan beberapa faktor yang rumit dan membingungkan, seperti faktor pencahayaan, posisi kamera, pengaturan kamera, dan *noise*. Jadi hasil dari *face recognition* akan berupa informasi yang dikenal atau tidak sebagai wajah dengan sebelumnya membandingkan dengan informasi dari wajah yang diketahui. Proses *face recognition* ini memiliki permasalahan dari pencahayaan, posisi kamera, parameter kamera dan *noise* yang didapatkan pada sebuah citra.

Menurut (Limowa, 2012) OpenCV menggunakan sebuah tipe *face detector* yang disebut *Haar-cascade classifier*. Jika ada sebuah image (bisa dari file /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 50×50 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.

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 3 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*.

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. Viola dan Jones kemudian mengembangkannya sehingga terbentuk Haar-Like feature.

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



**Gambar 2.9** Fitur Wajah

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 (massal). Semakin

banyak fungsi yang digunakan maka hasilnya akan semakin akurat. Pemrosesan Haar-Like feature yang banyak tersebut diorganisir atau diatur di dalam *classifier cascade*.

OpenCV *face detector* menggunakan metode Paul-Viola dan Michael Jones yang dipublikasikan pada tahun 2001. Pendekatan ini mendeteksi objek dengan menggabungkan 4 konsep :

- Fitur rectangular sederhana yang disebut fitur Haar
- Integral image untuk deteksi fitur yang cepat
- Metode machine learning AdaBoost
- Sebuah pengklasifikasi cascade untuk mengkombinasikan banyak fitur secara efisien.

Fitur yang digunakan Viola dan Jones menggunakan bentuk gelombang Haar. Bentuk gelombang Haar ialah sebuah gelombang kotak. Pada 2 dimensi, gelombang kotak ialah pasangan persegi yang bersebelahan, 1 terang dan 1 gelap. Haar ditentukan oleh pengurangan pixel rata-rata daerah gelap dari pixel rata-rata daerah terang. Jika perbedeaan diatas threshold (diset selama learning), fitur tersebut dikatakan ada. Untuk menentukan ada atau tidaknya Haar feature di setiap lokasi image / gambar, Viola dan Jones menggunakan teknik yang disebut Integral Image. Umumnya integral menambahkan unit kecil secara bersamaan. Dalam hal ini unit kecil ini disebut dengan nilai dari pixel. Nilai dari integral / *integral value* pada masing-masing pixel merupakan penjumlahan dari semua pixel di atasnya dan di sebelah kirinya. Dimulai dari kiri atas sampai kanan bawah, *image* / gambar dapat diintegrasikan sebagai operasi matematika per pixel.

Untuk memilih fitur Haar yang digunakan dan untuk mengubah nilai threshold, Viola dan Jones menggunakan metode machine-learning yang disebut AdaBoost. AdaBoost menggabungkan banyak *classifier* untuk membuat satu *classifier*. Masing-masing *classifier* menetapkan suatu bobot, dan gabungan dari bobot inilah yang akan membentuk satu *classifier* yang kuat.

Viola dan Jones menggabungkan serangkaian AdaBoost classifier sebagai rantai filter / *filter chain*. Masing-masing filter merupakan AdaBoost classifier yang terpisah dengan jumlah *weak classifier* yang sedikit dan sama.

Filter pada masing-masing level dilatih untuk mengklasifikasikan gambar yang sebelumnya telah difilter (Training set merupakan database dari wajah). Selama penggunaannya, jika satu dari filter-filter tersebut gagal, *image region* / daerah pada gambar diklasifikasikan sebagai “Bukan Wajah”. Saat filter berhasil melewati *image region*, *image region* kemudian masuk pada filter yang selanjutnya. *Image region* yang telah melalui semua filter akan dianggap sebagai “Wajah”.

## **2.10 Flowchart**

### **2.10.1 Pengertian Flowchart**

Menurut (Husnia, 2017) Flowchart atau Diagram Alir adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. Flowchart menolong analyst dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. Flowchart Biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. Flowchart adalah bentuk gambar/diagram yang mempunyai aliran satu atau dua arah secara sekuensial. Flowchart digunakan untuk merepresentasikan maupun mendesain program. Oleh karena itu flowchart harus bisa merepresentasikan komponen-komponen dalam bahasa pemrograman.

### **2.10.2 Pedoman Menggambar**







Pada tabel 2.1 menjelaskan simbol-simbol pada flowchart dan fungsinya. Pedoman dalam menggambar suatu Flowchart atau bagan alir, analisis sistem atau pemrograman sebagai berikut;

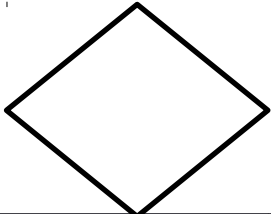
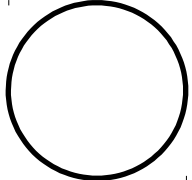
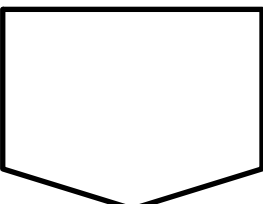
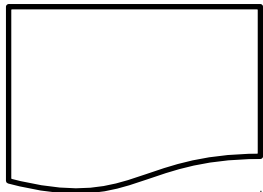

1. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan didalam bagan alir harus ditunjukkan dengan jelas.
3. Harus ditunjukkan darimana kegiatan akan dimulai dan dimana akan berakhirnya.



4. Masing-masing kegiatan didalam bagan alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan, misalnya; “persiapkan” dokumen “hitung” gaji.
5. Masing-masing kegiatan didalam bagan alir harus didalam urutan yang semestinya.
6. Kegiatan yang terpotong dan akan disambung ketempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
7. Gunakanlah simbol-simbol bagan alir yang standar.

**Tabel 2.1** Simbol-simbol Flowchart

<b>Simbol</b>	<b>Keterangan</b>
Simbol Terminal 	Permulaan atau akhir program.
Garis Alur ( <i>Flow Line</i> ) 	Arah Aliran Program.
Simbol Persiapan 	Proses instalasi atau pemberian nilai awal.
Simbol Proses 	Proses Penghitungan atau proses pengolahan data.
Input/Output 	Proses Input/Output data, parameter, informasi.
Simbol Sub Proses 	Permulaan sub program/ proses

	menjalankan sub program.
<p>Simbol Keputusan</p> 	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah sebelumnya.
<p>Connector</p> 	Penghubung bagian flowchart yang berada pada satu halaman.
<p>Simbol Penghubung</p> 	Penghubung bagian flowchart yang berada pada halaman yang berbeda.
<p>Document</p> 	Input/Output dalam format yang dicetak.
<p>Disk Storage</p> 	Input/Output yang menggunakan penyimpanan akses langsung.