

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Tinjauan pustaka yang dilakukan oleh penulis adalah dengan melakukan perbandingan yang sejenis dengan *literature* yang sama dan guna dijadikan bahan pertimbangan dan diharapkan dapat membantu dalam pembuatan sistem yang baru. Tinjauan pustaka ini yang dilakukan oleh penulis mengacu pada beberapa jurnal.

##### **2.1.1 Jurnal Satu**

Raden Budiarto (2017). Pada perancangannya yang berjudul “ KINERJA ALGORITMA PENGENALAN WAJAH UNTUK SISTEM PENGUNCIAN PINTU OTOMATIS MENGGUNAKAN RASPBERRY PI “. Pengenalan wajah mampu memberikan pengalaman interaksi yang paling alami, sebagaimana manusia mampu mengenali manusia lain melalui wajah. Implementasi pengenalan wajah memerlukan biaya yang rendah karena tidak memerlukan alat pengenal khusus selain kamera yang pada saat ini sudah tertanam di berbagai perangkat seperti laptop, smartphome, atau tablet. Pengenalan wajah bukan merupakan tugas mudah bagi komputer. Masalah bertambah ketika komputer diharuskan untuk melakukan klasifikasi wajah dengan berbagai situasi dan kondisi seperti pencahayaan yang gelap, dan tangkapan gambar latar belakang yang ada. Artikel ini mendeskripsikan hasil penelitian sistem pengenalan wajah menggunakan Raspberry Pi yang diterapkan untuk sebuah prototipe pengunci pintu. Metode yang digunakan adalah mengambil sampel dataset kemudian mengevaluasi dan membandingkan algoritme pembelajaran untuk dianalisis tingkat keakuratan dan

kecepatan dalam mengenali wajah. Pengujian dilakukan untuk menganalisis metode training dataset yang paling baik untuk diimplementasikan berdasarkan kriteria sensitivitas, spesifisitas, dan false rate. Terdapat 4 buah algoritme yang diuji yakni *Eigenfaces/PCA* dan *K-Nearest Neighbor* (K-NN), *PCA-LDA* dan *K-NN*, *Eigenfaces/PCA* dan *Support Vector Machine* (SVM), *PCA-LDA*, dan *SVM*. Hasil penelitian menunjukkan *algoritme hybrid Eigen-Fisherfaces* (*PCA-LDA*) dan *k-nearest neighbor* adalah yang metode yang paling akurat untuk pengenalan wajah. Akurasi mendekati 100% dapat diperoleh dengan perhitungan *machine learning* dengan 4 fold.

### 2.1.2 Jurnal Dua

Nico dan Imam Fahruzi (2015). Pada perancangannya yang berjudul “Rancang Bangun Sistem *Biometrik* Pengenalan Wajah Menggunakan *Principal Component Analysis*”. Dalam penelitian ini, penulis mengembangkan pengenalan struktur wajah menjadi sistem keamanan pintu ruangan secara *real-time*. Penelitian ini menggunakan sebuah *webcam* untuk meng-capture citra wajah pengguna dan kemudian dibandingkan dengan wajah yang tersimpan di *database*. Secara umum terdapat 2 metode yang digunakan oleh penulis yaitu metode *HaarCascade* untuk proses deteksi wajah dan metode *PCA* atau *Eigenface* untuk proses pengenalan wajah. Pengujian dilakukan dengan menggunakan 150 data training dan 150 data uji. Pada sistem ini, digunakan parameter jarak sebesar 30 cm untuk mengukur tingkat keakuratannya. Hasil pengujian menunjukkan tingkat keberhasilan pengenalan secara keseluruhan sebesar 83.33%. Sistem ini dirancang untuk bekerja secara *real-time* dengan harapan mempermudah pengguna dan dapat meminimalisir tindakan kriminal kedepannya.

### 2.1.3 Jurnal Tiga

Sinar Monika, Abdul Rakhman dan Lindawati (2017). Pada perancangannya yang berjudul “PENGAMANAN RUMAH DENGAN SISTEM *FACE RECOGNITION* SECARA *REAL TIME* MENGGUNAKAN METODE *PRINCIPAL COMPONENTS ANALYSIS*”. Pengaman rumah yang lebih canggih sangat diperlukan agar tidak terjadi tindak kejahatan yang tidak kita inginkan. Di implementasikan pada pengaman pintu dengan sistem *Face Recognition* atau pengenalan wajah, karena wajah adalah salah satu bentuk fisiologis yang paling mudah digunakan untuk membedakan identitas antar individu. Di proses dengan memasukkan citra yang sesuai dari 6 orang penghuni rumah untuk setiap satu orang memiliki 30 database citra yang diambil melalui kamera *webcam* dari sisi depan wajah tanpa menggunakan aksesoris. Dari database wajah tersebut lalu dilakukan pengujian yang berbasis *real time* objek masukannya berupa objek 3D yang diambil menggunakan kamera *webcam* karena mempunyai variabel yang sangat besar untuk melakukan transformasi dilakukan reduksi dimensi supaya menghasilkan variabel yang lebih rendah dan mudah untuk diobservasi dengan menggunakan metode *Principal Component Analysis* yang juga dikenal dengan metode *eigenfaces* yang telah dikembangkan dengan menggunakan *Software Visual Studio 2015* agar mendapatkan hasil yang sesuai dengan *database* secara *real time*. Persentase keberhasilan dalam percobaan ini mencapai nilai akurasi lebih dari 88%. Persentase keberhasilan tergantung dengan pencahayaan, serta faktor gambar yang dilakukan saat proses pengolahan citra, ekspresi dari wajah, dan sudut pengambilan gambar.

### 2.1.4 Jurnal Empat

Derian Indra Bramantio (2016). Pada perancangannya yang berjudul “PERANCANGAN IMPLEMENTASI KEAMANAN PINTU BERBASIS PENGENALAN WAJAH DENGAN METODE *EIGENFACE*”. Pada penelitian ini pengenalan wajah akan diimplementasikan untuk membuka dan mengunci pintu secara otomatis. Aplikasi yang dibuat terdiri dari *Raspberry Pi* sebagai mini PC dengan bahasa pemrograman *python*. Sistem ini diharapkan bisa mencari sendiri posisi dari wajah di dalam sebuah citra. Algoritma yang digunakan adalah

*Eigenface* atau yang biasa disebut dengan PCA ( *Principle Component Analysis* ). Penggunaan citra input diharapkan bisa menjadi lebih fleksibel, memiliki akurasi mencapai 90% sehingga hanya orang yang terdapat di database yang dapat masuk ke dalam rumah tersebut. Sedangkan *motor servo* dirancang agar dapat mengontrol posisi untuk membuka kunci.

### 2.1.5 Jurnal Lima

Mohamad Dimiyati Ayatullaha , Devit Suwardiyanto ,I WayanSuardinata (2018). Pada perancangannya yang berjudul “IMPLEMENTASI SIDIK JARI SEBAGAI OTENTIKASI PARKIR KENDARAAN MENGGUNAKAN *RASPBERRY PP*”. Otentikasi keluar masuk kendaraan pada tempat parkir merupakan salah satu permasalahan yang masih perlu dieksplorasi. Sebagian besar tempat parkir menggunakan otentifikasi mencatat nomor kendaraan pada kartu parkir. Petugas memferifikasi nomor kendaraan yang tertera dikartu parkir dengan nomor kendaraan yang tertera pada Surat Tanda Nomor Kendaraan (STNK). Metode otentifikasi tersebut membutuhkan ketelitian dari petugas parkir. Penelitian yang dilakukan memanfaatkan identifikasi sidik jari pengemudi kendaraan dan pengenalan plat nomor kendaraan melalui foto oleh petugas parkir sebagai otentikasi. Perekaman sidik jari dan foto kendaraan dilakukan pada saat kendaraan masuk. Identifikasi ini digunakan pada saat kendaraan akan keluar. Sistem dibangun dari beberapa perangkat, yaitu sensor *ultrasonic* untuk mendeteksi keberadaan kendaraan. Sensor *fingerprint* untuk mendeteksi sidik jari yang dilengkapi dengan *storage* untuk menyimpan template sidik jari. *Webcam* untuk pengambilan foto kendaraan sebagai verifikasi pada saat kendaraan keluar tempat parkir. Sedangkan pintu portal digerakan oleh *motor servo*. Semua sistem dikendalikan menggunakan *Raspberry Pi 3* menggunakan bahasa pemrograman *python*. Hasil pengujian pada sensor *ultrasonic* mampu mendeteksi keberadaan kendaraan pada jarak 10-200 cm dengan tingkat keakurasian 96%. *Motor servo* berputar posisi 0° jika nilai *DutyCycle* diatur 2,5 dan berputar 90° jika nilai *DutyCycle* 7,5. Posisi *webcam* berada 2 meter dari letak sensor *fingerprint*. Posisi jari yang ditekan menghasil proses yang lebih cepat dalam mengidentifikasi karakter sidik jari pada saat perekaman dan pencocokan sidik jari.

## 2.2 Dasar Teori

Pada Dasar teori disini, penulis mencantumkan teori – teori yang mendukung untuk penelitian “RANCANG BANGUN DUAL SISTEM KEAMANAN PINTU RUMAH OTOMATIS MENGGUNAKAN PENGENALAN WAJAH DAN SIDIK JARI”.

### 2.2.1 Biometrik

Secara harfiah, biometrika atau *biometrics* berasal dari kata bio dan metrics. Bio berarti sesuatu yang hidup, dan metrics berarti mengukur. Biometrika berarti mengukur karakteristik pembeda (*distinguishing traits*) pada badan atau perilaku seseorang yang digunakan untuk melakukan pengenalan secara otomatis terhadap identitas orang tersebut, dengan membandingkannya dengan karakteristik yang sebelumnya telah disimpan pada suatu *database*. Secara umum karakteristik pembeda tersebut dapat dikelompokkan menjadi 2, yaitu karakteristik fisiologis atau fisik (*physiological/physical characteristic*) dan karakteristik perilaku (*behavioral characteristic*). Biometrika berdasarkan karakteristik fisiologis/fisik menggunakan bagian-bagian fisik dari tubuh seseorang sebagai 6 kode unik untuk pengenalan seperti DNA, telinga, jejak paras pada wajah, geometri tangan, pembuluh tangan, sidik jari, iris mata, telapak tangan, retina mata, gigi dan bau (komposisi kimia) dari keringat tubuh. Sedangkan biometrik berdasarkan karakteristik perilaku menggunakan perilaku seseorang berdasarkan kode unik untuk melakukan pengenalan seperti gaya berjalan, hentakan tombol, tanda tangan dan suara.

### 2.2.2 Sidik Jari

Sebuah sidik jari adalah pola seperti *ridge* (gundukan) dan alur – alur yang terletak di ujung setiap jari. sidik jari telah digunakan untuk identifikasi pribadi selama berabad – abad dengan akurasi kecocokan yang sangat tinggi (Mario, Maltoni, Cappelli, Wayman & Jain, 2002). Sidik jari adalah pilihan utama untuk keunggulan sebagai pengenalan biometrik, sidik jari telah lama digunakan untuk tujuan otentifikasi maka dari itu sidik jari banyak digunakan untuk berbagai bidang seperti sistem absensi, imigrasi, akses bangunan rumah dan lain – lain.

Sidik jari terdiri dari pola gunung *Interleaved* (bagian dari naik ke atas) dan sebuah lembah (dips). Langkah pertama dalam pengenalan sidik jari biasanya melibatkan pengkategorian sidik jari menjadi satu dari lima kelas dasar, yang disebut kelas Henry terdiri dari *Plain Arch*, *Tanted Arch*, *Left Loop*, *Right Loop* dan *Whorl* (Whorl terbagi menjadi dua lingkaran polos dan kembar) (D. Maltoni, D. Maio, A. Jain & S. Prabhakar, 2009).

Langkah kedua dalam pengenalan sidik jari adalah menganalisa sidik jari di tingkat lokal. Analisa tingkat lokal melibatkan pemeriksaan diskontinuitas ridge kecil yang disebut *minutiae*. Dua jenis *minutiae* yang paling umum adalah bifurkasi terjadi pada titik di mana garis punggung mengarah ke dua segmen terpisah, sedangkan penghentiannya adalah ujung punggungnya yang prematur. Sidik jari yang khas mengandung hingga 80 hal kecil, namun lebih sedikit lagi yang akan ada pada gambar yang diambil dari pemindai biasa yang digunakan dalam sistem biometri karena tangkapan – tangkapan kecil.

Sensor sidik jari bekerja dengan mengambil gambar dari sidik jari dan membedakan setiap pola atau alur dari sidik jari tersebut. Sebenarnya banyak cara dapat dilakukan untuk mengambil gambar dari sidik jari tersebut, namun metode umum yang dilakukan adalah dengan menggunakan 2 (dua) cara, yaitu dengan sensor optikal dan sensor kapasitansi.

### 2.2.3 Pengenalan Wajah

*Face recognition* atau pengenalan wajah adalah salah satu teknologi biometrik yang telah banyak diaplikasikan dalam sistem *security* selain pengenalan retina mata, pengenalan sidik jari dan iris mata. Dalam aplikasinya sendiri pengenalan wajah menggunakan sebuah kamera untuk menangkap wajah seseorang kemudian dibandingkan dengan wajah yang sebelumnya yang telah disimpan di dalam *database* tertentu. *Face recognition* adalah teknologi dari komputer yang memungkinkan kita untuk mengidentifikasi atau memverifikasi wajah seseorang melalui sebuah gambar digital. Caranya ialah dengan mencocokkan tekstur lekuk wajah kita dengan data wajah yang tersimpan di *dataset*.



Tahap awal dalam pengenalan bentuk wajah (*face recognition*) yang sangat penting dilakukan adalah tahap pendeteksian wajah (*face detection*). Dan bidang-bidang dalam penelitian yang berkaitan dengan pemrosesan wajah (*face processing*) antara lain adalah:

1. Pengenalan wajah (*face recognition*) yaitu proses membandingkan sebuah citra wajah masukan dengan database wajah dan menemukan database wajah yang paling cocok dengan citra masukan tersebut.
2. Autentikasi wajah (*face authentication*) yaitu menguji keaslian/kesamaan suatu wajah dengan data wajah yang telah diinputkan sebelumnya.
3. Lokalisasi wajah (*face localization*) yaitu pendeteksian wajah namun dengan asumsi hanya ada satu wajah didalam citra.
4. Penjejakan wajah (*face tracking*) yaitu memperkirakan lokasi suatu wajah didalam video secara *real time*.
5. Pengenalan ekspresi wajah (*facial expression recognition*) untuk mengenali kondisi emosi manusia.

#### **2.2.4 Citra Digital**

Citra digital merupakan representatif dari citra yang diambil oleh mesin dengan bentuk pendekatan berdasarkan sampling dan kuantisasi. Sampling menyatakan besarnya kotak-kotak yang disusun dalam baris dan kolom. Dengan kata lain, sampling pada citra menyatakan besar kecilnya ukuran pixel (titik) pada citra, dan kuantisasi menyatakan besarnya nilai tingkat kecerahan yang dinyatakan dalam nilai tingkat keabuan (*grayscale*) sesuai dengan jumlah bit biner yang digunakan oleh mesin, dengan kata lain kuantisasi pada citra menyatakan jumlah warna yang ada pada citra.

Citra digital merupakan suatu matriks yang terdiri dari baris dan kolom, setiap pasangan indeks baris dan kolom menyatakan suatu titik pada citra. Nilai matriksnya menyatakan nilai kecerahan titik tersebut. Titik-titik tersebut dinamakan sebagai elemen citra, atau pixel (*picture element*).

## 2.2.5 Jenis - Jenis Citra Digital

Ada banyak cara untuk menyimpan citra digital di dalam memori. Cara penyimpanan menentukan jenis citra digital yang terbentuk. Beberapa jenis citra digital yang sering digunakan adalah citra warna, citra *grayscale* dan citra biner.

### 2.2.5.1 Citra Warna (*True Color*).

Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB = *Red Green Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte (nilai maksimum 255 warna), jadi satu piksel pada citra warna diwakili oleh 3 byte. Pengolahan citra digital adalah salah satu bentuk pemrosesan informasi dengan inputan berupa citra (*image*) dan keluaran yang juga berupa citra atau dapat juga bagian dari citra tersebut. Tujuan dari pemrosesan ini adalah memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin computer.



Gambar 2.1 Representasi Warna pada Citra RGB

Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna. Berarti setiap piksel mempunyai kombinasi warna sebanyak  $28 \times 28 \times 28 = 224 = 16$  juta warna lebih. Itulah sebabnya format ini dinamakan true color karena mempunyai jumlah warna yang cukup besar sehingga bisa dikatakan hampir mencakup semua warna di alam.



### 2.2.5.2 Citra *Grayscale* (Skala Keabuan)

Citra *Grayscale* merupakan citra yang setiap pixelnya bisa memiliki nilai lain diantara warna hitam dan putih, yang disebut nilai abu-abu. Banyaknya kemungkinan warna abu-abu tergantung besarnya kedalaman bit dari citra tersebut. Misalnya suatu citra mempunyai kedalaman sebesar 8 bit. Berarti citra tersebut mempunyai skala warna sebanyak 28 atau 256 skala yaitu skala nol sampai 255 ( 0-255). Angka nol mewakili warna hitam dan angka 255 mewakili warna putih sedangkan lainnya mewakili warna abu-abu. Warna abu-abu skala 200 lebih gelap jika dibandingkan warna abu-abu skala 201.

Perhitungan yang digunakan untuk mengubah citra berwarna yang mempunyai nilai matriks masing-masing R, G, dan B menjadi citra *grayscale* dapat dilakukan dengan mengambil rata-rata dari nilai R, G, dan B.

Untuk mengubah citra berwarna menjadi *Gray-scale* digunakan rumus berikut:

$$gray = 0,2989R + 0,5870G + 0,1140B$$

Keterangan:

*Gray* : Nilai *grayscale*

R : Nilai pada komponen R-layer

G : Nilai pada komponen G-layer

B : Nilai pada komponen B-Layer



Gambar 2.2 Skala Keabuan

### 2.2.5.3 Citra Biner (Monokrom)

Banyaknya dua warna, yaitu hitam dan putih. Dibutuhkan 1 bit di memori untuk menyimpan kedua warna ini. Citra biner (*binary image*) adalah citra digital yang hanya memiliki 2 kemungkinan warna, yaitu hitam dan putih. Citra biner

disebut juga dengan citra W&B (*White&Black*) atau citra monokrom. Piksel dengan derajat keabuan lebih besar dari nilai batas akan diberi nilai 1 dan sebaliknya piksel dengan derajat keabuan lebih kecil dari nilai batas akan diberi nilai 0.

Citra biner sering sekali muncul sebagai hasil dari proses pengolahan, seperti segmentasi, pengambangan, morfologi ataupun dithering. Fungsi dari binerisasi sendiri adalah untuk mempermudah proses pengenalan pola, karena pola akan lebih mudah terdeteksi pada citra yang mengandung lebih sedikit warna.



Gambar 2.3 Citra Biner dan Array Citra Biner

### 2.2.6 Pengolahan Citra

Pengolahan citra atau *image processing* adalah setiap bentuk pengolahan sinyal yang masukannya berupa gambar, sedangkan keluaran dari pengolahan tersebut dapat berupa gambar atau sejumlah karakteristik yang berkaitan dengan gambar. Singkatnya *image processing* adalah segala proses yang digunakan untuk mengolah suatu gambar.

Citra adalah suatu representasi (gambaran), kemiripan atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optic berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan.

Pengolahan citra digital merupakan proses yang bertujuan untuk memanipulasi dan menganalisis citra dengan bantuan komputer. Pengolahan citra digital dapat dikelompokkan dalam dua jenis kegiatan :

1. Memperbaiki kualitas suatu gambar, sehingga dapat lebih mudah diinterpretasi oleh mata manusia.
2. Mengolah informasi yang terdapat pada suatu gambar untuk keperluan pengenalan objek secara otomatis.

### 2.2.7 Raspberry Pi

*Raspberry Pi* (juga dikenal sebagai RasPi) adalah sebuah SBC (*Single Board Computer*) seukuran kartu kredit yang dikembangkan oleh Yayasan *Raspberry Pi* di Inggris (UK) dengan maksud untuk memicu pengajaran ilmu komputer dasar disekolah-sekolah. *Raspberry Pi* menggunakan *system on a chip* (SoC) dari Broadcom BCM2835 hingga BCM 2837 (*Raspberry Pi 3*), juga sudah termasuk prosesor ARM1176JZF-S MHz bahkan 1.2GHz 64-bit *quad-core* ARMv8 CPU untuk *Raspberry Pi 3*, GPU *VideoCore IV* dan kapasitas RAM hingga 1 GB (Astri,2016). Tidak menggunakan hard disk, namun menggunakan *SD Card* untuk proses booting dan penyimpanan data jangka-panjang. Dapat dilihat pada gambar 2.5



Gambar 2.5 Raspberry Pi

#### 2.2.7.1 Raspberry pi 3 model B

*Raspberry Pi 3* adalah generasi ketiga dari *Raspberry Pi*, menggantikan *Raspberry Pi 2 Model B* pada Februari 2016. *Raspberry Pi 3* memiliki bentuk yang identik dengan *Raspberry Pi 2* sebelumnya (dan *Pi 1 Model B*) dan memiliki kompatibilitas lengkap dengan *Raspberry Pi 1* dan 2. Pada perangkat terbarunya ini *Raspberry* menambahkan fitur *built-in wireless* dan *processor* yang lebih

bertenaga yang belum pernah dimiliki pada versi sebelumnya. *Spesifikasi Raspberry pi 3* dapat dilihat pada tabel 2.1

Tabel 2.1 Spesifikasi Raspberry pi 3 model B

NO	Spesifikasi	Keterangan
1	<i>Processor</i>	<i>Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz</i>
2	<i>Memory</i>	<i>1GB LPDDR2 SDRAM</i>
3	<i>Connectivity</i>	<i>2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps) 4 × USB 2.0 ports</i>
4	<i>Access</i>	<i>Extended 40-pin GPIO header</i>
5	<i>Video and Sound</i>	<i>1 × full size HDMI MIPI DSI display port MIPI CSI camera port 4 pole stereo output and composite video port</i>
6	<i>Multimedia</i>	<i>H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics</i>
7	<i>SD card support</i>	<i>Micro SD format for loading operating system and data storage</i>
8	<i>Input power</i>	<i>5V/2.5A D via micro USB connector 5V DC via GPIO header Power over Ethernet (PoE)–enabled (requires separate PoE HAT)</i>
9	<i>Environment</i>	<i>Operating temperature, 0–50°C</i>

Selain penjelasan spesifikasi diatas *Raspberry Pi* ini juga memiliki kelebihan dan kelemahan yaitu sebagai berikut :

Kelebihan *Raspberry Pi* :

1. *Raspberry Pi* ini menggunakan *Micro SD Card* untuk menyimpan data, baik itu data Operating System ataupun untuk media penyimpanan data jangka panjang.
2. Memiliki keunggulan pada grafis 3D dan tampilan Blu-ray pada video.
3. Mendukung *overclock* dan *overvolting* dengan cara mengedit file *config.txt*
4. Dapat menjalankan program - program perkantoran

Kelemahan *Raspberry Pi* :

1. *Raspberry Pi* ini cukup sensitive dengan listrik statis sehingga jika ingin menggunakan perangkat ini harus berhati – hati dalam memegangnya.
2. Jika mengedit file *config.txt* untuk membuat *Raspberry Pi* agar bisa *overclock* dan *overvolting* maka dapat memperpendek usia perangkat SoC (*System On Chip*)

### 2.2.7.2 Arsitektur *Raspberry Pi 3*

Arsitektur *Raspberry Pi* didasarkan seputar SoC (*System-on-a-chip*) Broadcom BCM2837, yang telah menanamkan *prosesor 1.4GHz 64-bit quad-core ARMv8, VideoCore IV 3D Graphics Core GPU, dan 1 Gigabyte RAM*. Penyimpanan data didesain tidak untuk menggunakan *hard disk* atau *solid-state drive*, melainkan mengandalkan kartu SD (*SD memory card*) untuk booting dan penyimpanan jangka panjang dilihat pada gambar 2.6



Gambar 2.6 Arsitektur *Raspberry Pi 3* model B

Keterangan:

1. Pin GPIO (40 Pin)
2. On Board Bluetooth 4.2 and BCM 43143 Wi-fi
3. DSI Display Port
4. BCM2837 14GHz -64-bit quad-core ARMv8 CPU and 1GB RAM
5. Micro USB Power Input Up to 2.5A
6. HDMI Video Output
7. CSI Camera Port
8. 3.5 mm 4-pole Composite Video and Audio Output Jack
9. Ethernet Port
10. 4 Usb Port

### 2.2.7.3 Konfigurasi Pin Raspberry Pi 3 model B

Jumlah pin GPIO berbeda untuk *Raspberry Pi* dengan *Raspberry Pi 2* dan *Raspberry Pi 1* memiliki 26 pin, sedangkan *Raspberry Pi 2* dan *3* memiliki 40 pin. Berikut Konfigurasi Pin GPIO pada gambar 2.7



Gambar 2.7 Konfigurasi Pin *Raspberry Pi* model B

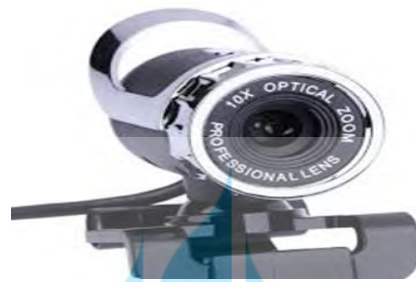
Beberapa istilah yang harus diperhatikan antara lain:

1. Pin 3.3V dan 5V : Pin ini merupakan pin yang berfungsi untuk memberikan tegangan ke komponen seperti *sensor*, *led*, motor dan *relay*. Pin ini dihubungkan ke pin vcc pada komponen.



2. Pin GND atau *Ground*, pin ini dihubungkan ke pin *ground* atau negatif (-) pada *led*, *sensor*, motor maupun *relay*.
3. Pin GPIO : Pin ini yang akan kita Kontrol melalui bahasa pemrograman Python. Dengan *Python* kita dapat mengatur apakah pin ini aktif atau mati maupun nyala berdasarkan kondisi tertentu dengan program yang dibuat

### 2.2.8 Webcam atau USB Camera



Gambar 2.8 USB Camera

*Webcam* (singkatan dari web camera) adalah sebutan bagi kamera *real-time* (bermakna keadaan pada saat ini juga) yang gambarnya bisa diakses atau dilihat melalui *World Wide Web*, *program instant messaging*, atau aplikasi *video call*. Istilah webcam merujuk pada teknologi secara umumnya, sehingga kata web terkadang diganti dengan kata lain yang mendeskripsikan pemandangan yang ditampilkan di kamera, misalnya *StreetCam* yang memperlihatkan pemandangan jalan.

*Webcam* ini berfungsi untuk memudahkan kita dalam mengolah pesan cepat seperti chat melalui video dan bertatap muka melalui video secara langsung dan webcam ini berfungsi sebagai alat untuk mentransfer sebuah media secara langsung.

### 2.2.9 Sensor Sidik jari



Gambar 2.9 Sensor *Fingerprint*

Sebuah sistem pemindai sidik jari memiliki dua pekerjaan, yakni mengambil gambar sidik jari, dan memutuskan apakah pola alur sidik jari dari gambar yang diambil sama dengan pola alur sidik jari yang ada di *database*. Ada beberapa cara untuk mengambil gambar sidik jari seseorang, namun salah satu metode yang paling banyak digunakan saat ini adalah *optical scanning*.

Inti dari pemindai optik adalah *charge coupled device* (CCD, Peranti tergendang–muatan), sistem sensor cahaya yang sama digunakan pada kamera digital dan *camcorder*. CCD merupakan sebuah larik sederhana dari diode peka cahaya yang disebut *photosite*, yang menghasilkan sinyal elektrik yang merespon foto cahaya. Setiap *photosite* merekam sebuah pixel, titik kecil yang merepresentasikan cahaya dan membenturnya. Pixel-pixel ini membentuk pola terang dan gelap dari sebuah gambar hasil scan sidik jari seseorang.

Proses scan mulai berlangsung saat jari diletakkan pada lempengan kaca dan sebuah kamera CCD mengambil gambarnya. Pemindai memiliki sumber cahaya sendiri, biasanya berupa *larik light emitting diodes* (LED), untuk menyinari alur sidik jari. Sistem CCD menghasilkan gambar jari yang terbalik, area yang lebih gelap merepresentasikan lebih banyak cahaya yang dipantulkan (bagian punggung dari alur sidik jari), dan area yang lebih terang merepresentasikan lebih sedikit cahaya yang dipantulkan (bagian lembah dari alur sidik jari)

#### 2.2.10 *OpenCV*

Dunia *opensource* seakan-akan tidak berhenti berinovasi dan menunjukkan eksistensinya diberbagai bidang ilmu didunia. Salah satu produk *opensource* itu adalah *openCV* (*Open Computer Vision*) yang merupakan sebuah API (*Application Programming Interface*) *Library* yang sudah sangat familiar pada Pengolahan Citra *Computer Vision*. *OpenCV* sendiri direlease dalam lisensi BSD dan bebas digunakan untuk keperluan akademik maupun komersial. Produk ini mendukung *interface*

*C/C++*, *python* dan *java* serta bisa berjalan diberbagai *platform* seperti *Windows*, *Linux*, *Mac OS*, *iOS* dan *Android*.

*Computer Vision* itu sendiri adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra atau dikenal sebagai *Image Processing* yang memungkinkan komputer dapat melihat seperti manusia. Dengan *vision* tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek. Beberapa pengembangan dan *project-project* dari produk ini adalah *Face Recognition*, *Face Detection*, *Face/Object Tracking*, *Road Tracking*, dll.

*OpenCV* memiliki segudang fitur yang bisa kita manfaatkan dalam melakukan riset atau pekerjaan kita yang berhubungan dengan *computer vision* (*image processing*, *video processing* dll) diantaranya :

1. Manipulation data citra (*alokasi*, *copying*, *setting*, *konversi*).
2. Citra dan *video I/O* (*file* dan *kamera based input*, *image/video file output*).
3. Manipulasi *Matriks* dan *Vektor* beserta aljabar linear (*products*, *solvers*, *eigenvalues*, *SVD*).
4. Data struktur dinamis (*lists*, *queues*, *sets*, *trees*, *graphs*).
5. Pemroses Citra fundamental (*filtering*, *edge detection*, *corner detection*, *sampling* and *interpolation*, *color conversion*, *morphological operations*, *histograms*, *image pyramids*).
6. Analisis struktur (*connected components*, *contour processing*, *distance transform*, *various moments*, *template matching*, *Hough transform*, *polygonal approximation*, *line fitting*, *ellipse fitting*, *Delaunay triangulation*).
7. Kalibrasi kamera (*calibration patterns*, *estimasi fundamental matrix*, *estimasi homography*, *stereo correspondence*).
8. Analisis gerakan (*optical flow*, *segmentation*, *tracking*).
9. Pengenalan obyek (*eigen-methods*, *HMM*).
10. Graphical User Interface (*display image/video*, penanganan *keyboard* dan *mouse handling*, *scroll-bars*).

#### 11. Pelabelan citra (*line, conic, polygon, text drawing*).

Didalam *openCV* terdapat 3 library utama yang bisa dipakai sesuai kebutuhan kita yakni :

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

Selain itu OpenCV juga dilengkapi dengan ***Machine Learning library*** yang memiliki algoritma berikut :

1. *Naive Bayes classifier*
2. *k-nearest neighbor algorithm*
3. *Support Vector Machine*
4. *Decision Trees*
5. *Boosting*
6. *Random forest*
7. *Expectation Maximization*
8. *Neural Networks*

*Library* ini bisa di-compile dengan Visual C++ 6.0, Visual C++.Net 2003/2005/2008 baik Standard, Professional maupun express edition ataupun menggunakan varian linux (hampir semua distribusi linux mendukung produk ini).

#### 2.2.11 Python

*Python* adalah bahasa pemrograman yang mudah dipelajari dan *powerful*. Python memiliki struktur data tingkat-tinggi yang efisien dan merupakan pendekatan yang sederhana tetapi efektif pada pemrograman yang berorientasi pada objek (*Object-oriented Programming*). Syntax elegan dan *dynamic typing* yang dimiliki oleh python, bersama dengan *interpreted nature* dari Python, menjadikannya bahasa pemrograman yang ideal untuk melakukan '*scripting*' dan pengembangan aplikasi yang pesat dalam banyak area pada kebanyakan platform.

##### 2.2.11.1 Fitur Fitur dari Python

Beberapa fitur yang dimiliki Python adalah:

1. memiliki kepustakaan yang luas; dalam distribusi Python telah disediakan modul-modul 'siap pakai' untuk berbagai keperluan.
2. memiliki tata bahasa yang jernih dan mudah dipelajari.
3. memiliki aturan *layout* kode sumber yang memudahkan pengecekan, pembacaan kembali dan penulisan ulang kode sumber.
4. berorientasi objek.
5. memiliki sistem pengelolaan memori otomatis (*garbage collection*, seperti java) modular, mudah dikembangkan dengan menciptakan modul-modul baru; modul-modul tersebut dapat dibangun dengan bahasa Python maupun C/C++.
6. memiliki fasilitas pengumpulan sampah otomatis, seperti halnya pada bahasa pemrograman Java, python memiliki fasilitas pengaturan penggunaan ingatan komputer sehingga para pemrogram tidak perlu melakukan pengaturan ingatan komputer secara langsung.
7. memiliki banyak fasilitas pendukung sehingga mudah dalam pengoperasiannya.

#### 2.2.11.2 Bahasa Pemrograman *Python*

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. *Python* merupakan salah satu bahasa pemrograman favorit saat ini, karena *Python* menawarkan banyak fitur seperti:

1. Kepustakaan yang luas, menyediakan modul-modul untuk berbagai keperluan.
2. Mendukung pemrograman berorientasi objek.
3. Memiliki tata bahasa yang mudah dipelajari.
4. Memiliki sistem pengelolaan memori otomatis.
5. Arsitektur yang dapat dikembangkan (*extensible*) dan ditanam (*embeddable*) dalam bahasa lain, misal *objek oriented Python* dapat digabungkan dengan modul yang dibuat dengan C++.

*Python* telah digunakan pada berbagai aplikasi saat ini, contohnya adalah BitTorrent, Yum, Civilization 4, bahkan saat ini *Python* merupakan bahasa resmi dari *Raspberry Pi*. Kata “Pi” dalam *Raspberry Pi* merujuk pada kata *Python*. *Python* mendukung beberapa modul khusus untuk *Raspberry Pi* seperti modul *picamera*, dan modul GPIO.

### 2.2.11.3 Deteksi Obyek Dengan Menggunakan Metode *Haar Cascade Classifier*

Pendeteksian obyek sangat mudah dilakukan oleh manusia tetapi merupakan tantangan tersendiri oleh sistem berbasis *computer vision*. Pada dasarnya pengembangan keilmuan *computer vision* adalah meniru bagaimana manusia melihat (*human vision*). Akan tetapi proses peniruan ini melibatkan studi yang sangat kompleks. Manusia melihat melalui mata yang kemudian dibawa ke otak untuk ditejemahkan atau dikenal (pambudi & simorangkir, 2012). Hasil pendeteksi ini dapat dikembangkan pada aplikasi-aplikasi yang lain. Sebagai contoh pengenalan wajah atau pola.

Meskipun tampak mudah dilakukan oleh manusia, ternyata pendeteksi wajah sangat rumit dilakukan oleh komputer karena terdapat beberapa kesulitan yang terkait dengan lokasi, sudut pandang, cahaya dan oklusi.

Pada awalnya deteksi obyek dilakukan menggunakan metode-metode deteksi tepi seperti *Sobel*, *Prewitts* atau *Canni*. Deteksi tepi dengan



metode-metode tersebut sangat sangat menguras sumber daya karena tingginya komputasi yang dilakukan *picture element (pixel)* (sajati & Astuti, 2013). Penggunaan deteksi tepi untuk pendeteksian obyek bergerak seperti pada format video dimana pergerakan *frame* dalam sebuah video dapat mencapai 20 *frame* per detik (fps). Metode deteksi tepi juga tidak efektif untuk diterapkan sebagai sistem pendeteksian obyek pada *object tracking* atau *fast object tracking*.

Deteksi obyek yang akan digunakan ini menggunakan sebuah metode 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”. Kalsifikasi wajah ini menggunakan sebuah pemisahan skala yang tetap, misalnya 50x50 *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*. Metode ini 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.

#### 2.2.11.4 Haar – Like Feature

Secara umum, *Haar – Like Feature* digunakan dalam pendeteksi objek pada *image digital*. Nama *Haar* merujuk pada suatu fungsi matematika (*Haar Wavelet*) yang berbentuk kotak, prinsipnya sama seperti pada fungsi *fourter*. Awalnya pengolahan gambar hanya dengan melihat dari nilai RGB setiap *pixel*, namun metode ini ternyata tidaklah efektif. Viola dan Jones kemudian mengembangkannya sehingga berbentuk *Haar – Like Feature*.

*Haar – Like Feature* memproses gambar dalam kotak-kotak, dimana dalam satu kotak terdapat beberapa pixel. Perkotak itupun kemudian di proses dan didapatkan perbedaan nilai (*thershold*) yang menandakan daerah gelap dan terang. Nilai-nilai inilah yang nantinya dijadikan dasar dalam *image processing*.

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*.

#### 2.2.12.1 Konsep Pendeteksian Wajah

Metode *face detector* menggunakan metode Paul-Viola dan Michael Jones yang dipublikasikan pada tahun 2001. Pendeteksian ini mendeteksi objek dengan menggunakan empat konsep :

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

Sebuah pengklasifikasi *cascade* untuk mengkombinasi banyak fitur secara efisien. Fitur yang digunakan Viola dan Jones menggunakan bentuk gelombang *Haar*. Bentuk gelombang *Haar* ialah sebuah gelombang kotak. Pada dua dimensi, gelombang kotak ialah pasangan persegi yang bersebelahan, satu terang satu gelap. *Haar* ditentukan oleh pengurangan pixel rata-rata daerah gelap dari *pixel* rata-rata daerah terang. Jika

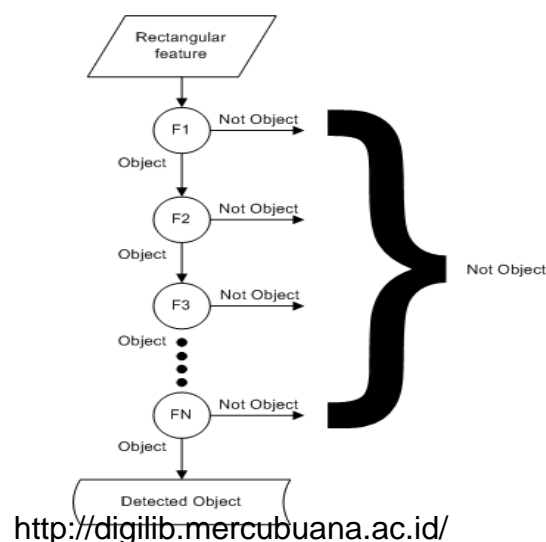
perbedaan tersebut *threshold* (diset selama *learning*), fitur tersebut dikatakan ada. Untuk menentukan ada atau tidaknya *Haar Feature* disetiap lokasi *Image* (gambar), Viola dan Jones menggunakan teknik yang disebut *Integral Image*. Umumnya *integral* menambahkan unit kecil secara bersamaan. Dalam hal ini unit kecil ini disebut dengan nilai dari *pixel*. Nilai dari *integral* atau *Integral Value* pada masing-masing *pixel* merupakan penjumlahan dari semua *pixel* diatas nya dan disebelah kirinya. Dimulai dari kiri atas sampai kanan bawah, *Image* (gambar) dapat diintegrasikan sebagai operasi matematika per *pixel*.

*Haar like feature* memproses citra dalam sebuah kotak persegi dengan ukuran tertentu misalnya 24 x 24 pixel seperti pada Gambar 2.10



Gambar 2.10 Rectangular Feature Haar cascade

Didalam kotak inilah proses *filtering* obyek dilakukan untuk diketahui apakah ada atau tidak obyek yang akan di deteksi. Proses filterisasi ini dilakukan secara bertingkat yang menyebabkan metode ini nantinya disebut sebagai *Haar Cascade Classifier* seperti ditunjukkan pada skema *filter* di Gambar 2.10.

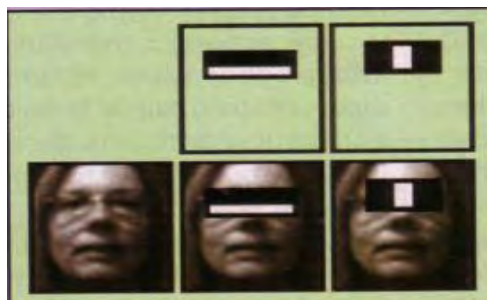


Gambar 2.11 Skema Pendeteksi Objek

Hasil deteksi *Haar-like Feature* kurang akurat jika hanya menggunakan satu fungsi saja. Semakin tinggi tingkatan *filter* pendeteksiannya maka semakin tepat pula sebuah obyek dideteksi akan tetapi, akan semakin lama proses pendeteksiannya. Pemrosesan *Haar-like Feature* yang banyak tersebut diatur dalam *Classifier Cascade*.

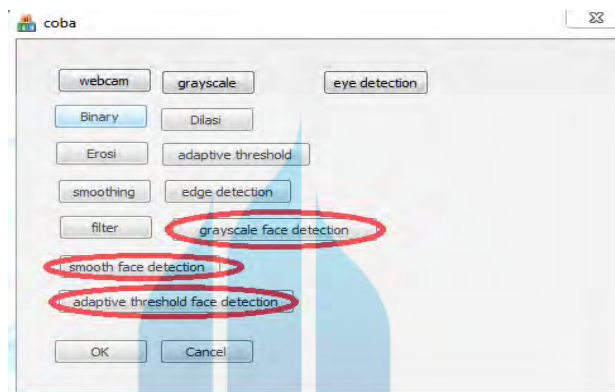
Untuk memilih fitur *Haar* yang digunakan dan untuk mengubah nilai *threshold*, Viola dan Jones menggunakan metode *machine – learning* yang disebut *AdaBoost*. *AdaBoost* menggabungkan banyak *classifier* untuk membuat *saty classifier*, masing – masing *classifier* menetapkan satu bobot dan gabungan 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”.



Gambar 2.12 Fitur Haar Cascade Classifier

Sebelum *OpenCv* melakukan tahap pendeteksian wajah hal yang perlu diperhatikan sebelumnya yaitu dapat melakukan beberapa *image processing* terlebih dahulu. Pada bagian ini, dapat melakukan *image processing* yaitu, *grayscale*, *smoothing (filtering)* dan *adaptive threshold*. Hasilnya adalah

Gambar 2.13 Hasil dari melakukan *image processing*

- *Grayscale*

Menggunakan fungsi `cv_RGB2GRAY` untuk merubah gambar RGB (3 channel) menjadi gambar *grayscale*.

Gambar 2.14 *Grayscale*

Setelah menjadi *grayscale* maka dilakukan *face detection* dan hasilnya *Haar Cascade* masih data pendeteksian bagian wajah.

- *Smoothing*

Menggunakan fungsi `cv_Smooth()` dengan tipe CV-BLUR untuk memburamkan gambar.



Gambar 2.15 *Smoothing*

*Smoothing filter* yang digunakan adalah *Gaussian*. *Filter Gaussian* mengkonvolusikan masing-masing titik pada *input array* dengan *Gaussian kernel* dan kemudian menjumlahkan *array output*. Untuk *Gaussian blur* kali ini, *karnel* yang digunakan adalah 15,15 ini menunjukkan lebar dan tinggi dari *filter window*. Gambar menjadi lebih gurat namun tetap dalam warna *RGB*, kemudian dilakukan *face detection* dan hasilnya bagian wajah masih dapat terdeteksi.

- *Adaptive Threshold*

Gambar *RGB* mula-mula di *convert* terlebih dahulu menjadi *grayscale*, kemudian dengan menggunakan fungsi `cvAdaptiveThreshold` maka dilakukan *thresholding* untuk memisahkan antara bagian gelap dan terang.





Gambar 2.15 Adaptive Threshold

Setelah itu dilakukan *face detection*, awalnya agak sulit bagi Haar Cascade untuk menemukan daerah wajah karena perbedaan kontras antara gelap dan terang telah menjadi tipis, namun setelah beberapa saat akhirnya didapatkan pula.

### 2.3 Posisi Penelitian

Tabel 2.2 Parameter Penelitian

No	JUDUL	PENGARANG	METODE PENELITIAN	KETERBATASAN	PROSPEK
1	Kinerja Algoritma Pengenalan Wajah Untuk Sistem Penguncian Pintu Otomatis Menggunakan Raspberry Pi	Raden Budiarto. (2017)	Peneliti menggunakan metode: <i>Eigenface/ PCA</i> Alat yang digunakan yaitu <i>Raspberry Pi 3</i>	Penelitian Hanya membuat <i>analysis</i> sistem algoritma pengenalan wajahnya saja dan Tidak membuat Rancang Bangun alatnya, pengujian ini hanya dapat diukur bersifat statistika, antara lain <i>detection rate</i> , tingkat akurasi validasi, waktu validasi, dan waktu pengenalan wajah.	Disini saya Membuat Rancang Bangun alat nya menggunakan <i>Raspberry pi 3</i> , sensor <i>camera Webcam</i> dan sensor sidik jari dan <i>solenoid</i> . menggunakan metode <i>Harr cascade classifier</i> dan saya tambahkan sistem pengamannya menjadi dua keamanan menggunakan pengenalan wajah dan

					sidik jari.
2	Rancang Bangun Sistem Biometrik Pengenalan Wajah Menggunakan <i>Principal Component Analysis</i> .	Nico dan Imam Fahruzi (2015).	Peneliti menggunakan metode: <i>Eigenface/ PCA</i> . Alat Yang Digunakan : Webcam, CPU Komputer, <i>Driver Selenoid</i>	Dalam penelitian ini, peneliti mengembangkan pengenalan struktur wajah menjadi sistem keamanan pintu ruangan secara <i>real-time</i> . Penelitian ini menggunakan sebuah webcam untuk meng-capture citra wajah pengguna dan kemudian dibandingkan dengan wajah yang tersimpan di database	penulis menambahkan komponen <i>Raspberry Pi 3</i> dan sensor sidik jari. Dan metode <i>Haar Cascade</i> untuk proses deteksi wajah. <i>Haar Feature</i> memiliki kelebihan antara lain adalah komputasinya yang sangat cepat.
3	Pengaman Rumah Dengan sistem <i>Face Recognition Secara Real Time</i> Menggunakan Metode <i>Principal Component Analysis (PCA)</i>	Sinar Monika, Abdul Rakhman dan Lindawati (2017).	Peneliti menggunakan metode ( <i>PCA</i> ) / <i>Eigenface</i> . Alat yang digunakan Yaitu Camera dan Menggunakan Software Visual Studio 2015.	Dalam penelitian Terdapat beberapa faktor yang dapat mempengaruhi hasil pengenalan citra masukan, yaitu pada saat pencahayaan yang kurang terang, sudut pengambilan masukan citra wajah, kemiripan nilai <i>eigenface</i> antar penghuni, serta ekspresi wajah.	Penulis Menambahkan alat <i>raspberry Pi</i> sebagai pengolah datanya. <i>Raspberry Pi</i> telah dilengkapi dengan semua fungsi layaknya sebuah komputer lengkap, menggunakan SoC (Sistem-on-Chip) Perangkat ini menggunakan kartu SD untuk booting dan penyimpanan jangka panjang. <i>Raspberry Pi</i> bersifat <i>open source</i> (berbasis <i>Linux</i> ) dan bisa dimodifikasi sesuai kebutuhan

					penggunanya. Sistem operasi utama Raspberry Pi menggunakan Debian GNU/Linux dan bahasa pemrograman <i>Python</i> .
4	Perancangan Implementasi Keamanan Pintu Berbasis Pengenalan Wajah Dengan Metode <i>Eigenface</i> .	Derian Indra Bramantio (2016).	Peneliti menggunakan metode <i>Eigenface</i> . Alat yang digunakan Yaitu Camera <i>webcam</i> , <i>Raspberry Pi</i> , <i>motor servo</i> .	Penelitian hanya Meneliti tingkat Keakurasian <i>Face Recognition</i> untuk keamanan pintu menggunakan pengenalan wajah.	Penulis Menambahkan Output sensor fingerprint, Menjadi dua keamanan sistem pintu rumah pengenalan wajah Dan sidik jari yang mana kedua nya saling sinkron dengan dataset.
5	Implementasi Sidik Jari Sebagai Otentikasi Parkir Kendaraan Menggunakan <i>Raspberry Pi</i>	Mohamad Dimiyati Ayatullaha, Devit Suwardiyanto, I WayanSuardinata (2018).	Alat yang digunakan <i>Raspberry Pi</i> , sensor <i>ultrasonic</i> , sensor <i>Fingerprint</i> , <i>webcam</i> , <i>motor servo</i>	Penelitian disini mengimplementasikan sensor sidik jari untuk mempercepat keluar masuk parkir kendaraan dan menambah sistem keamanan kendaraan Ketika parkir, Kendaraan yang akan keluar tempat parkir, diawali dengan pencocokan sidik jari pengemudi yang telah di scan Ketika kendaraan masuk.	Penulis mengimplementasikan dan mengembangkan sidik jari ini ke keamanan sistem pintu rumah dan dipadukan dengan pengenalan wajah.

Dari hasil perbandingan jurnal penelitian yang sudah ada, disini penulis menambahkan alat yaitu menambahkan sebuah sensor sidik jari sehingga adanya pengembangan dari jurnal jurnal penelitian sebelumnya.

