

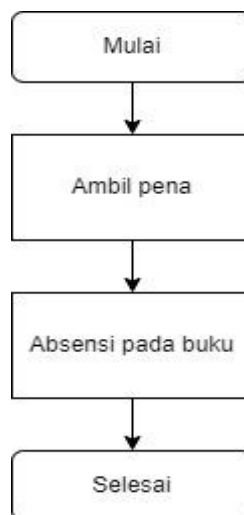
BAB III ANALISA DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Tahap proses analisis sistem dapat didefinisikan sebagai penguraian dari sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan.

3.1.1 Analisis Sistem Yang Berjalan Saat Ini

Pada sistem yang berjalan saat ini, presensi pada rumah makan Mbak Dwi masih menggunakan sistem manual dengan cara tanda tangan pada buku dan menggunakan pena yang sama. Setelah presensi selesai, hasil presensi dilaporkan kepenanggung jawab pada hari itu juga. Hal ini rentan terjadi manipulasi data presensi pegawai serta rentan terhadap penularan COVID-19 karena menggunakan pena yang sama saat melakukan presensi. Alur proses presensi pegawai yang berjalan saat ini dapat dilihat pada Gambar 3.1.

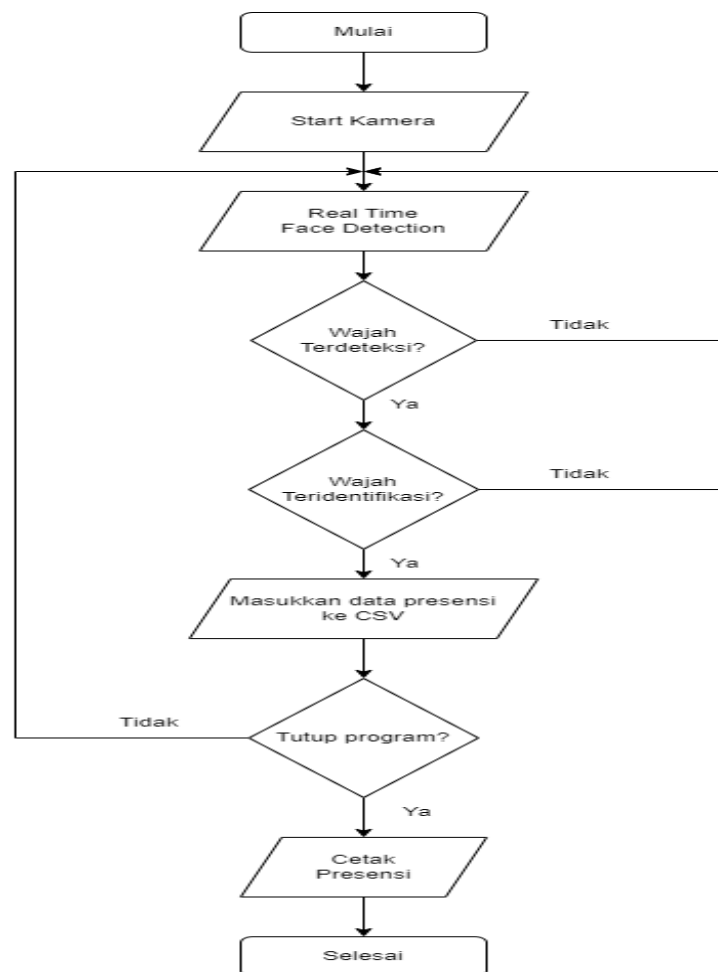


Gambar 3.1 Alur Proses Presensi Pegawai

Berdasarkan permasalahan tersebut, maka solusinya adalah dengan merancang dan membangun *face recognition* secara *real time* untuk meminimalisir kecurangan pegawai berupa memanipulasi data presensi serta dapat meminimalisir penyebaran COVID-19 pada rumah makan Mbak Dwi.

3.1.2 Analisis Sistem Yang Baru

Face recognition ini akan dibuat menggunakan Raspberry Pi 3B dan Raspberry Pi camera V2 sebagai komponen utama. Adapun cara kerja dari *face recognition* menggunakan Raspberry Pi camera V2 berbasis Raspberry Pi 3B, akan dijelaskan dengan alur proses pada Gambar 3.2.



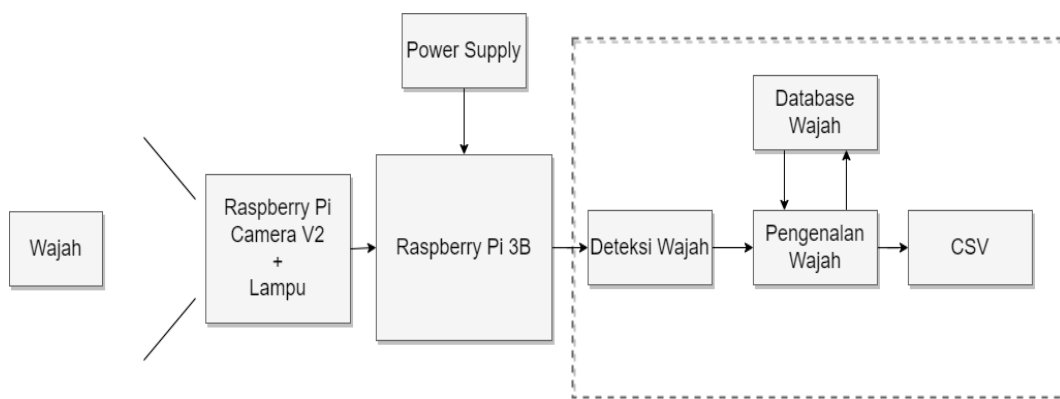
Gambar 3.2 Alur Proses *Face Recognition*

Face recognition dimulai dari kamera yang melakukan deteksi wajah secara *real time*, kemudian mengenali wajah apakah terdeteksi atau tidak, bila wajah terdeteksi kemudian melakukan identifikasi wajah dalam *database*. Jika nama terdeteksi, maka akan memunculkan nama identitas yang ada dalam *database*. Proses dilakukan secara terus menerus sampai program diberhentikan oleh pegawai.

3.2 Perancangan Sistem

3.2.1 Diagram Blok Sistem

Diagram blok merupakan gambaran dasar dari rangkaian sistem yang akan dirancang. Setiap blok dalam suatu diagram blok memiliki fungsi masing–masing baik dari segi *hardware* maupun *software*. Diagram blok banyak digunakan dalam dunia rekayasa seperti desain *hardware*, desain elektronik, *software* desain dan proses aliran diagram. Diagram blok sistem dapat dilihat pada Gambar 3.3.



Gambar 3.3 Diagram Blok *Face Recognition*

Blok diagram di atas menjelaskan bagaimana alur berjalannya *face recognition*. Raspberry Pi 3B mendapatkan energi listrik dari *power supply*. Raspberry Pi Camera V2 yang difungsikan sebagai input wajah akan mengirimkan citra wajah yang tertangkap oleh kamera ke dalam Raspberry Pi 3B. Raspberry Pi 3B akan menampilkan wajah untuk kemudian dengan menggunakan algoritma *haar cascade* sebagai penangkap wajah. Setelah berhasil mendeteksi wajah, algoritma *Local Binary Pattern Histogram* (LBPH) akan dapat mengenali wajah atau tidak mengenali wajah. Wajah yang dikenali merupakan wajah yang sudah terdaftar dalam *database*. Kemudian jika wajah yang dideteksi merupakan wajah yang cocok dalam *database*, maka nama wajah yang dikenali akan dikirim ke CSV sebagai tanda bukti hadir pada hari itu.

Berikut adalah proses pengenalan wajah oleh algoritma *Local Binary Pattern Histogram* (LBPH)

1. Ekstraksi matriks *histogram* citra wajah dengan menggunakan algoritma *Local Binary Pattern Histogram* (LBPH)

Pada proses ini, citra wajah akan di ubah menjadi *grayscale* lalu diubah menjadi matriks (nxn) dengan cara membagi citra wajah menjadi beberapa bagian berdasarkan jumlah pixel dari citra wajah tersebut. Setiap angka pada matriks memiliki rentang angka pixel 0-255. Nilai pixel 0 yaitu pixel tersebut hitam/gelap sedangkan nilai pixel 255 yaitu pixel tersebut putih/terang. Matriks citra wajah *original* ditunjukkan pada Gambar 3.4.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix}$$

Gambar 3.4 Matriks citra wajah *original*

Setelah mendapatkan matriks citra wajah *original* (nxn) tersebut, selanjutnya adalah membagi lagi menjadi matriks 3x3. Proses pembagian matriks citra wajah *original* menjadi matriks 3x3 ditunjukkan pada Gambar 3.5.

$$\left[\begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array} \right], \left[\begin{array}{ccc} a_{12} & a_{13} & a_{14} \\ a_{22} & a_{23} & a_{24} \\ a_{32} & a_{33} & a_{34} \end{array} \right] \dots \dots \dots \left[\begin{array}{ccc} a_{(n-2)(n-2)} & a_{(n-2)(n-1)} & a_{(n-2)n} \\ a_{(n-1)(n-2)} & a_{(n-1)(n-1)} & a_{(n-1)n} \\ a_{n(n-2)} & a_{n(n-1)} & a_{nn} \end{array} \right]$$

(n-2)²

Gambar 3.5 Proses pembagian matriks citra wajah *original* menjadi matriks 3x3

Setelah mendapatkan matriks 3x3, ambil satu nilai pixel tengah untuk menjadi pembanding. Jika nilai disekelilingnya lebih besar atau sama dari pixel pembanding, maka akan diubah nilainya menjadi 1 dan sebaliknya jika nilai disekelilingnya lebih kecil pixel pembanding, maka akan diubah nilainya menjadi 0. Proses pembagian dan perubahan nilai matriks 3x3 ini dilakukan dari a_{11} sampai a_{nn} sehingga didapatkan matriks *histogram* dari citra wajah tersebut. Proses perbandingan pixel pusat dengan sekelilingnya dan proses penggantian nilai pusat ditunjukkan pada Gambar 3.6.

$$P_{ij} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \quad T_{ij} = \begin{bmatrix} 1 & 2 & 4 \\ 128 & 0 & 8 \\ 64 & 32 & 16 \end{bmatrix}$$

$$P_{ij} = \{x; x=0 \text{ untuk } P_{ij} < p_{22}, x=1 \text{ untuk } P_{ij} \geq p_{22}\}$$

$$LBPH_k = \sum (P_{ij} * T_{ij})$$

Gambar 3.6 Proses perbandingan pixel pusat dengan sekelilingnya dan proses penggantian nilai pusat

Keterangan gambar:

P_{ij} = Pecahan matriks 3x3

T_{ij} = Nilai bobot algoritma LBPH

$LBPH_k$ = Nilai pusat baru pengganti nilai pusat *original*

$k = 1, 2, \dots, (n-2)^2$

$i, j = 1, 2, 3.$

Setelah mendapatkan nilai pusat baru dari semua matriks 3x3, maka didapatkan matriks *histogram* citra wajah (nxn). Matriks *histogram* citra wajah ditunjukkan pada Gambar 3.7.

$$H = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & LBPH_{22} & LBPH_{23} & \dots & a_{2n} \\ a_{31} & LBPH_{32} & LBPH_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix}$$

Gambar 3.7 Matriks *histogram* citra wajah

2. Mencari nilai *histogram* terkecil untuk menentukan citra wajah yang cocok

Setelah mendapatkan matriks *histogram* dari citra wajah, algoritma *Local Binary Pattern Histogram* (LBPH) akan menghitung jumlah nilai *histogram* terkecil untuk menemukan citra wajah yang cocok pada citra wajah di *database*. Rumus yang dipakai untuk menghitung nilai *histogram* terkecil ditunjukkan pada Gambar 3.8.

$$D = \sqrt{\sum_{i=1}^n (\text{histogram1}_i - \text{histogram2}_i)^2}$$

Gambar 3.8 Rumus menghitung nilai *histogram* terkecil

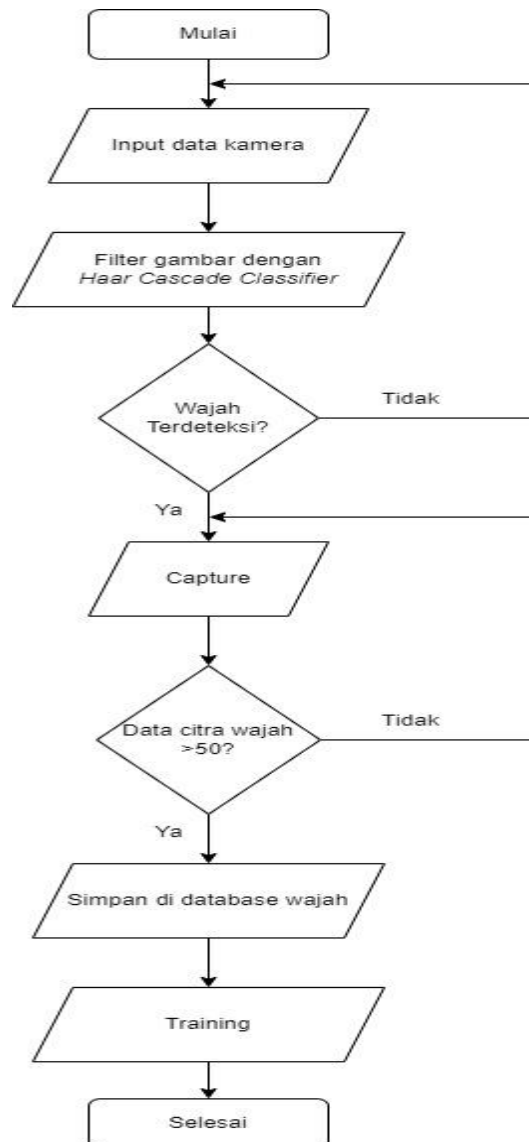
Nilai D merupakan nilai *histogram* terkecil citra wajah yang disimpan pada *database* dengan citra wajah uji.

3.2.2 *Flowchart*

Flowchart dapat didefinisikan sebagai bagan dengan simbol-simbol tertentu yang menggambarkan atau menunjukkan arus kerja secara keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur atau proses yang ada di dalam sistem. *flowchart* menunjukkan apa yang dikerjakan pada sistem.

3.2.2.1 Flowchart Database

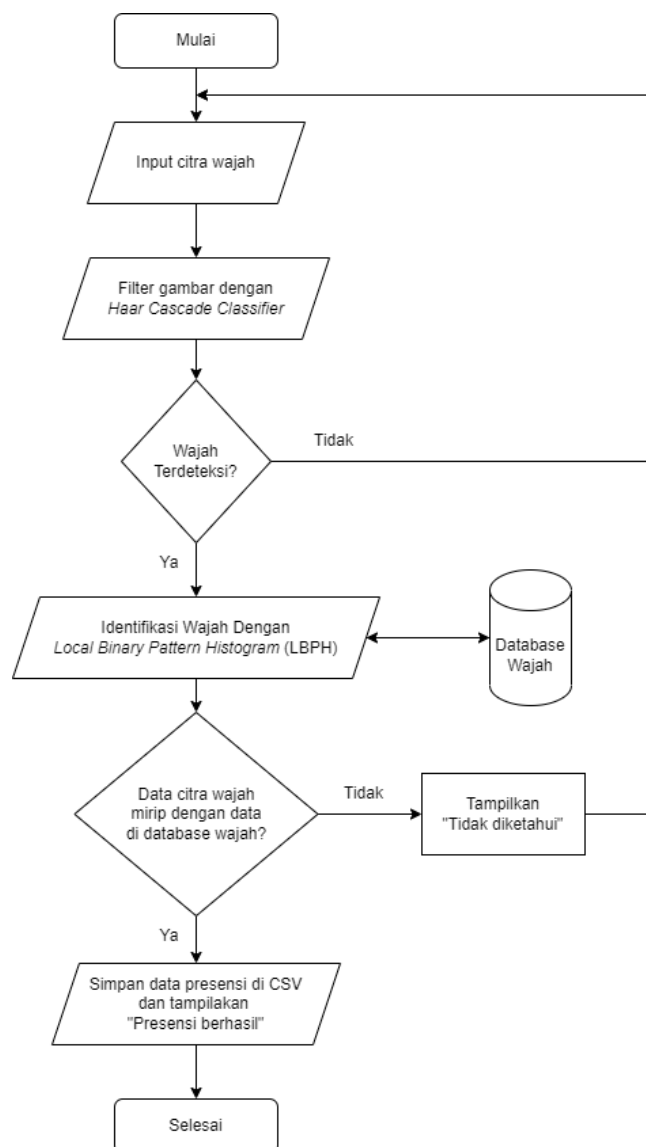
Flowchart database menjelaskan alur proses pengambilan citra wajah sampai penyimpanan citra wajah di *database* dan *training*. Kamera akan menjadi input data, sedangkan algoritma *haar cascade* akan menangkap wajah. Kamera akan meng-*capture* gambar wajah sebanyak 51 kali untuk kemudian disimpan dalam *database*. Data (gambar wajah) dalam *database* tersebut kemudian akan di *training* yakni dengan diberi nama dan ID yang berbeda untuk wajah yang berbeda pula sebelum diidentifikasi seperti dalam *flowchart* presensi. *Flowchart database* ditunjukkan pada Gambar 3.9.



Gambar 3.9 *Flowchart Database*

3.2.2.2 Flowchart Presensi

Flowchart presensi menjelaskan alur presensi pegawai. Kamera akan menjadi input data, sedangkan algoritma *haar cascade classifier* akan menangkap. Jika wajah sudah dideteksi, maka proses identifikasi dilakukan oleh algoritma *Local Binary Pattern Histogram (LBPH)*. Bila wajah sesuai dengan *database*, maka nama yang sudah terdaftar dalam *database* akan di tampilkan pada CSV sebagai pegawai yang sudah hadir. Jika tidak, akan muncul “Tidak diketahui” pada wajah yang dideteksi. Flowchart presensi ditunjukkan pada Gambar 3.10.



Gambar 3.10 Flowchart Presensi