

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Penggajian karyawan merupakan satu proses yang sangat penting dalam sebuah perusahaan. Penggajian melibatkan waktu yang dihabiskan karyawan dalam melakukan sebuah pekerjaan yang diberikan oleh sebuah perusahaan. Berbagai jenis penggajian yang terdapat dalam dunia pekerjaan, ada penggajian per hari yang mana gaji karyawan dihitung berdasarkan kerja per hari, ada penggajian per bulan yang mana karyawan digajian berdasarkan satu bulan pekerjaan. Ada penggajian per kontrak yang mana karyawan digaji berdasarkan kontrak yang telah disepakati.

Absensi adalah sebuah kegiatan yang bertujuan untuk melakukan pendataan kehadiran seseorang atau karyawan. Saat ini, proses absensi masih banyak dilakukan dengan cara manual atau konvensional seperti, dengan memanggil nama atau mengecek karyawan yang hadir dan menuliskannya ke dalam kertas absensi. Cara ini dianggap kurang efektif karena proses absensi dilakukan ketika admin atau seseorang yang mengabsen melakukan absen tidak tepat waktu, maka karyawan yang datang terlambat tidak dapat di record.

Pesatnya perkembangan teknologi saat ini dapat digunakan untuk menyelesaikan masalah di atas dengan cara menggunakan teknologi biometrik yang telah banyak diterapkan di berbagai bidang salah satunya yaitu pengenalan wajah (*Face recognition*). Dengan menggunakan teknologi tersebut, maka sebuah sistem dapat mengenali seseorang dengan menyimpan tekstur muka orang tersebut pada database dan menggunakan sebuah kamera digital untuk mencocokkan tekstur wajah pada database.

Sama halnya dalam proses perhitungan penggajian karyawan pada suatu perusahaan pabrik, khususnya di perusahaan CV.Tirta Rahman Parepare. Sistem penggajian karyawan masih dilakukan dengan cara manual dengan menuliskan nama-nama karyawan di sebuah buku dan menjumlah gaji tiap karyawan secara manual, yang mana hal tersebut tentunya sangat kurang efektif dan tidak efisien dalam melakukan penggajian di suatu perusahaan pabrik.

CV. Tirta Rahman Parepare adalah sebuah perusahaan pabrik yang memproduksi air minum dalam kemasan. Perusahaan tersebut memiliki cukup banyak karyawan dan staff yang bekerja setiap harinya (45 orang). Perusahaan tersebut dalam proses absensi dan penggajian masih dengan cara manual atau konvensional. Pendataan absensi dan penggajian karyawan masih menggunakan cara konvensional, sehingga masih kurang efektif. Hal ini dikarenakan tidak ada salinan dokumen yang telah dibuat sehingga dokumen sering hilang, dan sering terjadi kesalahan dalam perhitungan gaji.

Sehingga dampak yang terjadi adalah banyaknya *complain* dari karyawan karena tidak *valid*-nya data dan gaji yang mereka terima, terlebih lagi di CV. Tirta Rahman Parepare ada dua kategori dalam penggajian karyawan yang dilakukan yaitu, karyawan dengan gaji hitungan perhari kerja dan gaji hitungan perbulan.

Mengingat kelemahan sistem absensi dan perhitungan gaji karyawan di CV. Tirta Rahman Parepare, maka perlu adanya komponen-komponen yang dapat mendukung agar proses absensi dan perhitungan gaji lebih efektif dan efisien. Maka, penulis ingin merancang suatu alat absensi secara otomatis yang berbasis *face recognition* dan nantinya juga akan terhubung langsung ke sebuah database pada suatu aplikasi atau sistem *desktop* yang dibuat untuk mengelola absensi dan penggajian karyawan tersebut secara otomatis.

Metode pengenalan wajah yang digunakan dalam sistem presensi kelas adalah dengan metode haar cascade classifier karena metode haar cascade classifier mempunyai kelebihan yaitu komputasi yang sangat cepat, karena hanya tergantung pada jumlah piksel dalam persegi bukan setiap nilai piksel dari sebuah image.

Berdasarkan uraian di atas maka penulis ingin mengambil judul Rancang Bangun Sistem Absensi Karyawan berbasis Face Recognition Menggunakan Metode Haar cascade (Studi Kasus: CV. Tirta Rahman Parepare).

## **1.2 Rumusan Masalah**

Bagaimana membangun sistem absensi karyawan yang berbasis *face recognition* dengan menggunakan metode haar cascade, dan dapat menghasilkan output berupa slip gaji?

### **1.3 Tujuan Penelitian**

Membangun sistem absensi dan aplikasi penggajian karyawan yang berbasis face recognition dengan menggunakan metode haar cascade, dan dapat menghasilkan output berupa slip gaji.

### **1.4 Manfaat Penelitian**

1. Dapat membantu untuk mengawasi dan mendapatkan laporan absensi karyawan secara otomatis.
2. Memudahkan perusahaan untuk menghitung gaji karyawan dengan akurat dan pembuatan dokumen atau slip gaji karyawan.
3. Penelitian ini dapat menambah wawasan untuk penelitian selanjutnya dan dapat membantu perkembangan teknologi yang digunakan dalam sistem absensi dan penggajian karyawan di suatu perusahaan.

### **1.5 Batasan Masalah**

1. Data yang diolah berupa gambar wajah dan absensi karyawan
2. Gambar diambil menggunakan modul kamera Raspberry Pi 4
3. Sistem diolah menggunakan laptop dengan spesifikasi Core i7 Gen 7, Ram 16 Gb
4. Jarak antar kameran dan objek 1,5 m
5. Setelah objek terdeteksi akan menyimpan foto absensi ke sistem
6. Sistem menampilkan laporan absensi karyawan kepada atasan perusahaan.
7. Sistem penggajian dapat merincikan gaji tiap karyawan tiap bulannya
8. Sistem penggajian dapat mencetak slip gaji karyawan dalam bentuk pdf.

### **1.6 Sistematika Penulisan**

Untuk memberikan gambaran singkat mengenai isi tulisan secara keseluruhan, maka akan diuraikan beberapa tahapan dari penulisan secara sistematis, yaitu:

## **BAB I PENDAHULUAN**

Bab ini menguraikan secara umum mengenai hal yang menyangkut latar belakang, perumusan masalah dan batasan masalah, tujuan, manfaat, dan sistematika penulisan.

## **BAB II TINJAUAN PUSTAKA**

Bab ini berisi teori-teori tentang hal-hal yang berhubungan dengan sistem yang dianalisis, antara lain mengenai modul praktikum, metode yang digunakan, hingga bahasa yang digunakan dalam analisis tersebut.

## **BAB III METODE PENELITIAN**

Bab ini berisi tentang tahapan penelitian, waktu dan lokasi penelitian, instrumen penelitian, tahap persiapan, gambaran umum sistem, skenario pengujian dan analisis performa.

## **BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisi tentang hasil performa serta pembahasan yang disertai pemaparan hasil penelitian.

## **BAB V PENUTUP**

Bab ini berisi tentang kesimpulan yang didapatkan berdasarkan hasil penelitian yang telah dilakukan serta saran-saran untuk pengembangan lebih lanjut.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Landasan Teori**

##### **2.1.1 OpenCV**

*Open Computer Vision* (OpenCV) merupakan *library open source* yang tujuannya dikhususkan untuk melakukan pengolahan citra. Maksudnya adalah agar komputer mempunyai kemampuan yang mirip dengan cara pengolahan visual pada manusia (Purnama, 2021). *OpenCV* telah menyediakan banyak algoritma visi komputer dasar. *OpenCV* juga menyediakan modul pendeteksian objek yang menggunakan metode *computer vision*.

##### **2.1.2 Python**

Bahasa pemrograman *Python* merupakan salah satu bahasa level tinggi yang saat ini sedang berkembang. Sintaks di *Python* membantu para developer untuk melakukan pengkodean dengan langkah-langkah yang lebih sedikit jika dibandingkan dengan *Java* atau *C++*. Banyak perusahaan besar yang menggunakan *Python* untuk mengembangkan aplikasi mereka, seperti *Instagram*, *Youtube*, dan lain-lain.

Kelebihan dari bahasa *python* adalah seperti berikut:

1. *Library* yang luas dan banyak

Bahasa pemrograman *Python* memiliki *library* luas dengan beragam modul yang siap untuk Anda gunakan. Di dalamnya terdapat beragam kode untuk beragam keperluan seperti *regular expressions*, *documentation-generation*, *unit-testing*, *databases*, *CGI*, *email*, dan masih banyak lagi. Dengan kelebihan ini, maka Anda tidak perlu lagi menulis kode lengkap secara manual.

2. Meningkatkan produktivitas *developer*

Bahasa yang sederhana serta *library* yang luas dapat membuat developer menjadi lebih produktif. Selain itu, dengan *Python* Anda juga

hanya perlu menulis kode lebih sedikit sehingga Anda mempunyai lebih banyak waktu untuk bisa mengerjakan yang lain.

### 3. *Embeddable*

*Python* juga dapat ditanam atau disematkan. Anda dapat meletakkan kode *Python* Anda ke dalam sumber bahasa lain seperti *C++*. Kemampuan ini memungkinkan Anda untuk menambahkan kemampuan *scripting* ke dalam bahasa lain.

### 4. Mendukung *IoT*

*Python* mendukung *Internet of Things (IoT)* dengan sangat baik. *IoT* merupakan teknologi yang dapat menghubungkan benda-benda di sekitar Anda ke dalam sebuah jaringan yang menghubungkan satu dengan yang lain.

## 2.1.3 Cloudflare

*Cloudflare* adalah salah satu penyedia jaringan pengiriman konten (*Content Delivery Network*) terpopuler di dunia. *Content Delivery Network (CDN)* sendiri adalah jaringan yang berfungsi untuk mengirimkan konten dari *server* ke *website*.

Dengan kata lain, *Cloudflare* bertindak sebagai proxy antara website dan visitor. *Cloudflare* memiliki banyak server yang tersebar di beberapa negara, hal itu menjadikan loading sebuah website lebih cepat. Sebagai gambaran sederhana, *Cloudflare* bertindak sebagai penghubung antara pengunjung dan server website. Tujuannya, supaya website terhindar dari serangan berbahaya yang dilakukan oleh pengunjung website.

## 2.1.4 Cloudflare Tunnel

*Cloudflare Tunnel* adalah perangkat lunak terowongan yang memungkinkan Anda dengan cepat mengamankan dan mengenkripsi lalu lintas aplikasi ke semua jenis infrastruktur, sehingga Anda dapat menyembunyikan alamat IP server web, memblokir serangan langsung, dan kembali mengirimkan aplikasi hebat.

*Daemon Tunnel* membuat terowongan terenkripsi antara server web asal Anda dan pusat data terdekat *Cloudflare*, semuanya tanpa membuka port masuk publik apa pun.

Setelah mengunci semua port dan protokol server asal menggunakan firewall Anda, semua permintaan pada port HTTP/S akan dibatalkan, termasuk serangan DDoS volumetrik . Upaya pelanggaran data – seperti pengintaian data saat transit atau serangan login brute force – diblokir seluruhnya.

### 2.1.5 JSON

JSON adalah turunan JavaScript yang digunakan dalam transfer dan penyimpanan data. JavaScript object notation atau JSON adalah format yang digunakan untuk menyimpan dan mentransfer data. Berbeda dengan XML (extensive markup language) dan format lainnya yang memiliki fungsi serupa, JSON memiliki struktur data yang sederhana dan mudah dipahami. Itulah mengapa JSON sering digunakan pada API.

JSON sendiri terdiri dari dua struktur, yaitu:

- Kumpulan value yang saling berpasangan. Dalam JSON, contohnya adalah object.
- Daftar value yang berurutan, seperti array

### 2.1.6 Haar Cascade

Pengklasifikasi *Haar*, atau pengklasifikasi *cascade Haar*, adalah program deteksi objek pembelajaran mesin yang mengidentifikasi objek dalam gambar dan video. Deskripsi terperinci tentang pengklasifikasi *Haar* dapat dilihat di makalah *Paul Viola dan Michael Jones "Deteksi Objek Rap menggunakan Cascade yang Ditingkatkan dari Fitur Sederhana"*. Algoritma dapat dijelaskan dalam empat tahap:

1. Menghitung Fitur *Haar*
2. Membuat Gambar Integral
3. Menggunakan *Adaboost*
4. Menerapkan *Cascading Classifiers*

Penting untuk diingat bahwa algoritma ini membutuhkan banyak gambar positif wajah dan gambar negatif non-wajah untuk melatih pengklasifikasi, mirip dengan model pembelajaran mesin lainnya.

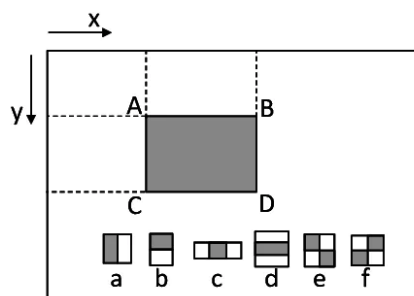
Langkah pertama adalah mengumpulkan fitur Haar. Fitur Haar pada dasarnya adalah perhitungan yang dilakukan pada wilayah persegi panjang yang berdekatan di lokasi tertentu di jendela deteksi. Perhitungannya melibatkan penjumlahan intensitas piksel di setiap wilayah dan menghitung perbedaan antara jumlah. Berikut adalah beberapa contoh fitur Haar di bawah ini.



Gambar 1 Jenis Fitur *Haar*

Sumber : <https://medium.com/analytics-vidhya/haar-cascades-explained-38210e57970d>

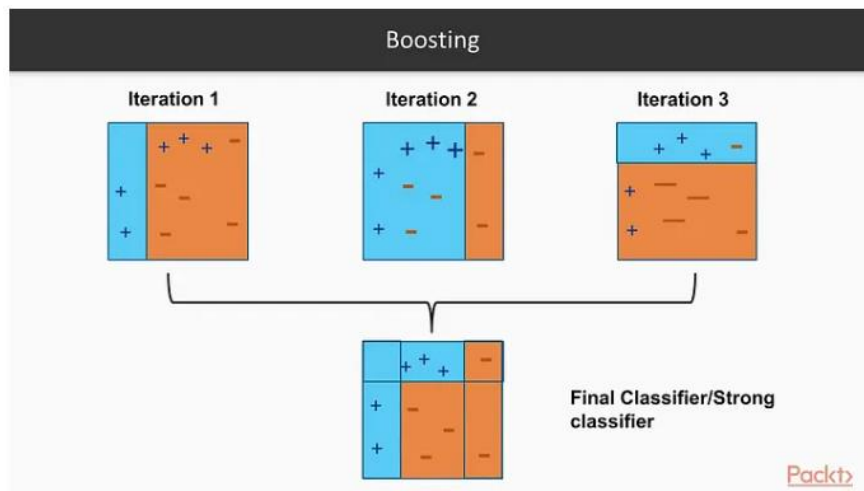
Fitur-fitur ini bisa sulit ditentukan untuk gambar besar. Di sinilah gambar integral berperan karena jumlah operasi dikurangi menggunakan gambar integral. gambar integral pada dasarnya mempercepat perhitungan fitur Haar ini. Alih-alih komputasi di setiap piksel, itu malah membuat sub-persegi panjang dan membuat referensi array untuk masing-masing sub-persegi panjang itu. Ini kemudian digunakan untuk menghitung fitur Haar.



Gambar 2 Ilustrasi untuk cara kerja gambar integral

Sumber : <https://medium.com/analytics-vidhya/haar-cascades-explained-38210e57970d>

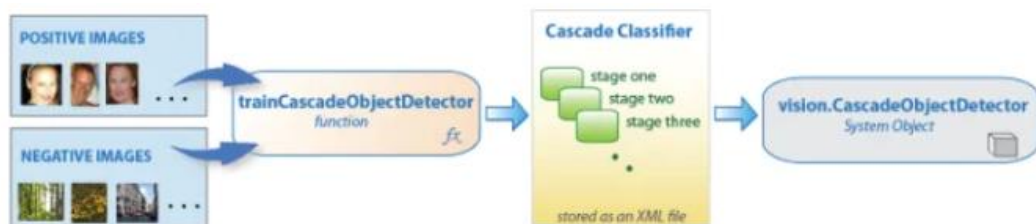
Penting untuk dicatat bahwa hampir semua fitur Haar tidak akan relevan saat melakukan deteksi objek, karena satu-satunya fitur yang penting adalah fitur objek. Adaboost pada dasarnya memilih fitur terbaik dan melatih pengklasifikasi untuk menggunakannya. Itu menggunakan kombinasi "pengklasifikasi lemah" untuk membuat "pengklasifikasi kuat" yang dapat digunakan algoritma untuk mendeteksi objek. Pembelajar yang lemah dibuat dengan menggerakkan jendela di atas gambar input, dan menghitung fitur Haar untuk setiap subbagian gambar. Perbedaan ini dibandingkan dengan ambang yang dipelajari yang memisahkan non-objek dari objek. Karena ini adalah "pengklasifikasi lemah," sejumlah besar fitur Haar diperlukan untuk akurasi untuk membentuk pengklasifikasi yang kuat.



Gambar 3 Representasi dari algoritma adaboost

Sumber : <https://medium.com/analytics-vidhya/haar-cascades-explained-38210e57970d>

Langkah terakhir menggabungkan pelajar yang lemah ini menjadi pembelajar yang kuat menggunakan pengklasifikasi cascading.



Gambar 4 Diagram alur pengklasifikasian cascade

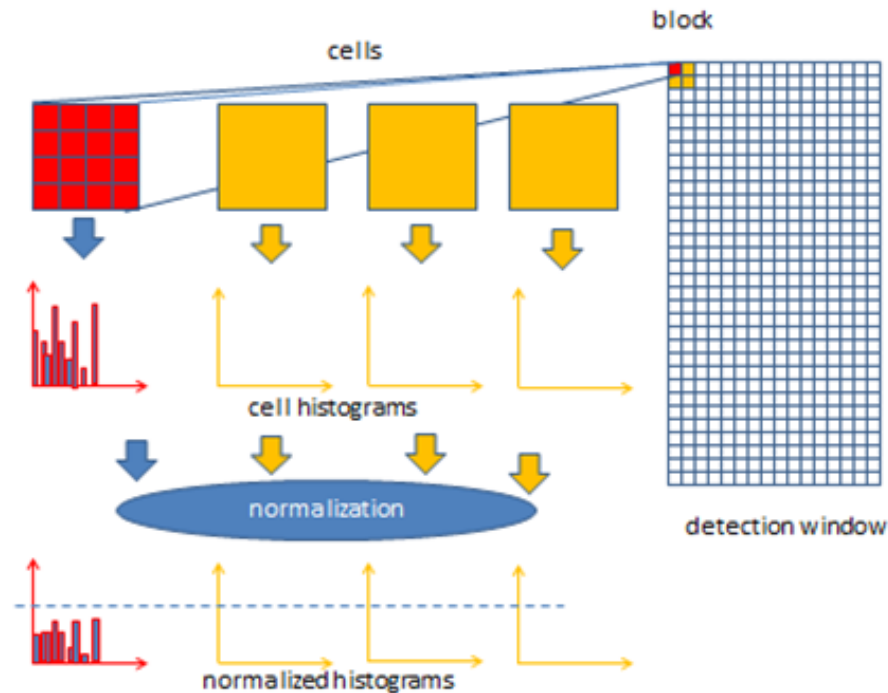
Sumber : <https://medium.com/analytics-vidhya/haar-cascades-explained-38210e57970d>

Pengklasifikasi kaskade terdiri dari serangkaian tahap, di mana setiap tahap adalah kumpulan pelajar yang lemah. Pembelajaran yang lemah dilatih menggunakan peningkatan, yang memungkinkan pengklasifikasi yang sangat akurat dari prediksi rata-rata semua pelajar yang lemah. Berdasarkan prediksi ini, pengklasifikasi memutuskan untuk menunjukkan objek ditemukan (positif) atau pindah ke wilayah berikutnya (negatif). Tahapan dirancang untuk menolak sampel negatif secepat mungkin, karena sebagian besar jendela tidak mengandung sesuatu yang menarik. Penting untuk memaksimalkan tingkat negatif palsu yang rendah, karena mengklasifikasikan objek sebagai non-objek akan sangat merusak algoritma deteksi objek Anda. Sebuah video di bawah ini menunjukkan riam Haar sedang beraksi. Kotak merah menunjukkan "positif" dari pelajar yang lemah. Haar cascades adalah salah satu dari banyak algoritma yang saat ini digunakan untuk deteksi objek. Satu hal yang perlu diperhatikan tentang kaskade Haar adalah sangat penting untuk mengurangi tingkat negatif palsu.

### **2.1.7 Alternatif HaarCascade (Histogram of oriented gradients)**

*Histogram gradien berorientasi (HOG)* adalah deskriptor fitur yang digunakan untuk mendeteksi objek dalam penglihatan komputer dan pemrosesan gambar. Teknik deskriptor *HOG* menghitung kejadian orientasi gradien di bagian lokal dari jendela deteksi gambar -, atau wilayah yang menarik (*ROI*). Implementasi algoritma deskriptor HOG adalah sebagai berikut:

Bagilah gambar menjadi daerah kecil yang terhubung yang disebut sel, dan untuk setiap sel menghitung histogram arah gradien atau orientasi tepi untuk piksel di dalam sel. Diskret setiap sel menjadi tempat sampah sudut sesuai dengan orientasi gradien. Setiap piksel sel berkontribusi gradien berbobot ke tempat sampah sudut yang sesuai. Kelompok sel yang berdekatan dianggap sebagai wilayah spasial yang disebut blok. Pengelompokan sel ke dalam blok adalah dasar untuk pengelompokan dan normalisasi histogram. Kelompok histogram yang dinormalisasi mewakili histogram blok. Kumpulan histogram blok ini mewakili deskriptor. Gambar berikut mendemonstrasikan skema implementasi algoritma:



Gambar 5 Histogram dari gradien berorientasi

Sumber : [https://www.researchgate.net/figure/HOG-descriptor-algorithm-Computation-of-the-HOG-descriptor-requires-the-following-basic\\_fig2\\_305391222](https://www.researchgate.net/figure/HOG-descriptor-algorithm-Computation-of-the-HOG-descriptor-requires-the-following-basic_fig2_305391222)

### 2.1.8 Face Recognition

*Face recognition* adalah sistem identifikasi dan autentikasi seseorang dengan menggunakan fitur wajah yang dimiliki. Sistem pada *face recognition* bisa mengenali wajah baik pada foto, video, maupun secara *real-time*. *Face recognition* memiliki tingkat keakuratan yang cukup tinggi. Walaupun tidak setinggi autentikasi iris atau retina, masalahnya teknologi pengenalan iris maupun retina ini membutuhkan biaya jauh lebih mahal. Teknologi face recognition ini sendiri sudah lebih aman jika dibandingkan metode keamanan konvensional. Selain itu bisa didapatkan dengan biaya yang jauh lebih terjangkau.

Secara umum, *face recognition* berfungsi untuk memverifikasi identitas seseorang apakah cocok dengan *database* yang dimiliki. Jika seandainya ditemukan kecocokan maka identitas tersebut bisa mengakses layanan atau aplikasi.

*Face recognition* dianggap sebagai sistem identifikasi keamanan yang paling natural karena yang pertama kali dilihat dari seseorang adalah wajahnya, wajah adalah apa yang membedakan seseorang dengan orang lainnya. Setiap orang di

muka bumi memiliki perbedaan wajah yang bisa dianalisis bahkan wajah antara kembar identik masih bisa dibedakan.

Cara kerja *face recognition* sendiri secara umum seperti yang kita gunakan pada ponsel yang memiliki fitur buka kunci biometrik pengenalan wajah. Ketika kamu menggunakan metode penguncian ponsel menggunakan wajah maka kamu sebagai pengguna akan memasukkan data wajah kamu ke sistem ponsel. Kemudian ponsel akan mencoba mengidentifikasi wajah tersebut menggunakan kamera ponsel. Apabila ponsel menemukan kecocokan antara wajah yang terdapat pada data maka kemudian ponsel akan bisa dibuka kuncinya.

#### 1. *Face Detection*

Kamera akan mendeteksi dan menangkap apakah pada sebuah gambar, video, atau kejadian *real-time* terdapat wajah. Sistem kemudian akan mengkonfirmasi apakah gambar yang tertangkap tersebut adalah sebuah wajah atau tidak.

#### 2. *Face Analysis*

Gambar yang sebelumnya sudah ditangkap kemudian akan dianalisa oleh sistem. Sistem analisisnya pada umumnya akan menggunakan pendekatan *2D* karena database yang menggunakan gambar *2D* jauh lebih nyaman dan gampang dicocokkan. Berbeda dengan analisa *3D* yang membutuhkan cetak wajah secara keseluruhan.

Terdapat beberapa pendekatan yang bisa dilakukan saat metode analisa. Misalnya, geometri wajah yang mengukur jarak antara fitur-fitur wajah seperti jarak antara bibir dan hidung, jarak antara mata kiri dan kanan, kedalaman mata, jarak antara jidat dan dagu, serta lain sebagainya.

Terdapat juga metode analisis menggunakan fotometris yaitu menggunakan warna kulit, bentuk hidung, warna iris mata, bentuk bibir, kontur wajah, dan lain sebagainya. Lalu terdapat pendekatan menggunakan tekstur wajah dimana pori-pori, keunikan wajah, dan tanda khusus pada wajah menjadi patokan.

### 3. *Face Verification*

Wajah yang telah ditangkap oleh kamera dan dianalisis oleh sistem kemudian akan dikonversi menjadi data agar pencocokan dapat dengan lebih mudah dilakukan oleh sistem. Fitur-fitur wajah yang sudah dianalisa diubah menjadi data-data berbasis matematika yang nantinya akan diterjemahkan oleh database.

### 4. *Face Identification*

Pada tahapan ini data wajah akan diidentifikasi menggunakan analisa dan verifikasi wajah yang telah ditentukan sebelumnya. *Scanning* dilakukan pada wajah secara menyeluruh mulai dari kontur wajah, geometris wajah, hingga tekstur wajah.

Data wajah tersebut akan diubah menjadi titik-titik yang akan dikonversi menjadi data. Pada tahap ini data yang masuk akan diidentifikasi ulang dengan membandingkan data yang masuk dengan data yang terdapat pada database hingga mencapai perbandingan 1:1. Terakhir sistem akan menilai apakah wajah tersebut cocok dengan database atau tidak.

### 5. *Face Match*

Data wajah yang telah teridentifikasi akan menunjukkan apakah wajah yang ditangkap adalah benar terdapat pada database. Kemudian sistem akan mengkonfirmasi apakah wajah tersebut sesuai dengan identitas yang ditangkap atau tidak. Setelah itu baru keputusan akan diambil apakah identitas tersebut berhak mengakses layanan, membuka aplikasi, atau masuk ke ruangan tertentu.

### 2.1.9 Raspberry Pi

*Raspberry Pi* adalah sebuah komputer papan tunggal (*single-board computer*) atau *SBC* seukuran kartu kredit yang dapat digunakan untuk menjalankan program perkantoran, permainan komputer, dan sebagai pemutar media hingga video beresolusi tinggi. *Raspberry Pi* dikembangkan oleh yayasan nirlaba, *Raspberry Pi Foundation* dengan tujuan untuk belajar pemrograman (Fadila, Darjat, & Sudjadi, 2015).



Gambar 6 Raspberry Pi Product

*Raspberry Pi* pertama kali dikembangkan di laboratorium Komputer Universitas Cambridge oleh Eben Upton, Rob Mullins, Jack Lang, dan Alan Mycroft. Mereka kemudian mendirikan yayasan *Raspberry Pi* bersama dengan Pete Lomas dan David Braben pada tahun 2009. Pada tahun 2012, *Raspberry Pi* Model B memasuki produksi massal. Dalam peluncuran pertamanya pada akhir Februari 2012 dalam beberapa jam saja sudah terjual 100.000 unit. Pada bulan Februari 2016, *Raspberry Pi Foundation* mengumumkan bahwa mereka telah menjual 8 juta perangkat *Raspi*, sehingga menjadikannya sebagai perangkat paling laris di Inggris.

Nama *Raspberry Pi* diambil dari nama buah, yaitu buah *Raspberry*, sedangkan *Pi* diambil dari kata *Python*, yaitu nama dari sebuah bahasa pemrograman. *Python* dijadikan bahasa pemrograman utama dari *Raspberry Pi*, namun tidak tertutup kemungkinan untuk menggunakan bahasa pemrograman lain pada *Raspberry Pi*. Keunggulan *python* dibanding dengan bahasa pemrograman yang lain adalah kode kode lebih mudah ditulis dan dibaca, dan juga banyak

terdapat modul modul yang beragam. Adapun kekurangannya adalah tidak realtime, sehingga untuk akan kesusahan untuk melakukan pekerjaan yang mempunyai *delay*, akibatnya tingkat presisi juga tidak tinggi.

*Raspberry Pi* memiliki komponen yang hampir serupa dengan komputer pada umumnya. Seperti *CPU*, *GPU*, *RAM*, *Port USB*, *Audio Jack*, *HDMI*, *Ethernet*, dan *GPIO*. Untuk tempat penyimpanan data dan sistem operasi *Raspberry Pi* tidak menggunakan *harddisk drive (HDD)* melainkan menggunakan *Micro SD* dengan kapasitas paling tidak 4 GB, sedangkan untuk sumber tenaga berasal dari *micro USB power* dengan sumber daya yang direkomendasikan yaitu sebesar 5V dan minimal arus 700 mA.

*Raspberry Pi* dapat digunakan layaknya *PC* konvensional, seperti untuk mengetik dokumen atau sekedar browsing. Namun *Raspberry Pi* juga dapat digunakan untuk membuat ide-ide inovatif seperti membuat robot yang dilengkapi dengan *Raspberry Pi* dan kamera, atau mungkin dapat membuat sebuah super komputer yang dibuat dari beberapa buah *Raspberry Pi*. Kelengkapan *Raspberry Pi* di antaranya memiliki port atau koneksi untuk display berupa *TV* atau *monitor* serta koneksi *USB* untuk *keyboard* serta *mouse*.

Fungsi dan penggunaan *raspberry pi* :

1. Komputer yang biasa digunakan untuk melakukan kegiatan ringan sehari-hari seperti menonton *HD movie*, *word processing*, mendengarkan musik dan kegiatan lainnya.
2. Dapat dihubungkan dengan aneka macam sensor seperti sensor cahaya, sensor suhu, sensor gerak dan yang lainnya.
3. Dapat dibuat *web server*.
4. Dapat dijadikan server *NAS( Network Attached Storage)* di rumah.
5. Dapat dikombinasikan bersama *Arduino*.
6. *Download Manager* dalam hal ini *Raspberry* dapat dijadikan sebagai komputer yang *mendownload* file-file film yang dapat ditinggal tanpa mengkhawatirkan konsumsi listrik.

7. *Printer Server* yang merupakan alat yang dihubungkan dengan *printer* sehingga menjadikan *printer* dapat digunakan bersama-sama dalam sebuah jaringan *LAN*.
8. *Wifi Internet Radio Player* dimana *Raspberry Pi* digunakan untuk memainkan musik yang di streaming dari internet radio, dapat mengatur *volume*, mengatur daftar lagu dari *handphone*.
9. Dapat digunakan untuk menjadi server untuk *hosting website*, berbasis *html*, *php* dan *mysql*.

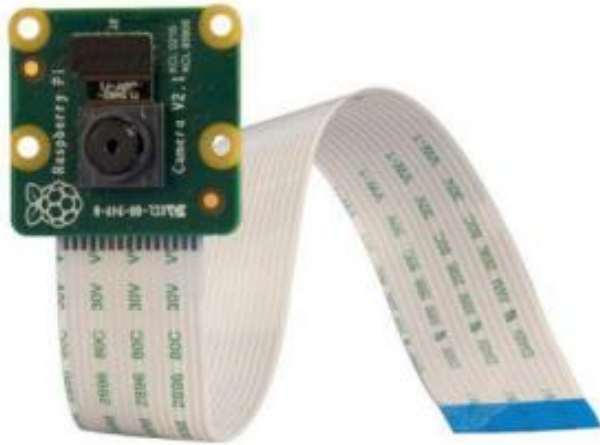
#### **2.1.10 Raspberry Pi Camera**

*Raspberry PI Camera* merupakan modul kamera webcam yang diterbitkan oleh *Raspberry Pi Foundation*. Modul kamera ini menggunakan *port CSI* antarmuka modul kamera dengan *Raspberry Pi Board*. Selain itu dengan menggunakan *Raspbian* pengguna dapat mengatur kamera sesuai kebutuhan misal mengatur *ISO*, *AWB*, *framerate*, dan lain sebagainya.

Modul kamera ini juga memiliki sensor yang baik sehingga dapat mengambil gambar dengan kualitas yang cukup baik.

Spesifikasi *Raspbeery Pi Camera* :

1. Resolusi kamera 5 *megapixel*.
2. Menggunakan kamera *fix*.
3. Menggunakan sensor gambar *Omnivision 5647*.
4. 15 pin *CSI interface* dengan kabel *Extremely* data tinggi.
5. Ukuran 29x25x9 mm.



Gambar 7 Raspberry Pi Camera

Koneksi kamera dengan *Raspberry Pi* menggunakan kabel *CSI* ( *Camera Serial Interface*) sebagai penghubung, port yang digunakan adalah *port CSI*. Kamera yang digunakan dalam penelitian ini yaitu *Raspbbery Pi Camera Rev 1.3*. Penggunaan Raspicam bertujuan untuk mendapatkan hasil yang maksimal dalam pengambila citra gambar.

#### 2.1.11 Absensi

Absensi adalah istilah yang mengacu pada ketidakhadiran atau tidak berpartisipasi seseorang dalam suatu kegiatan tertentu. Singkatnya, absensi itu berarti seseorang tidak hadir secara fisik di tempat yang diadakan. Konsekuensi yang didapat dari absensi juga berbeda-beda tergantung konteks berikut:

1. Absensi dalam pendidikan berarti siswa tidak hadir di lingkungan pembelajaran. Absensi yang sering dilakukan bisa mempengaruhi kemajuan akademik siswa ke depannya.

Contoh absensi dalam pendidikan: siswa sering tidak hadir di kelas tanpa alasan dan tidak mengikuti kegiatan akademik yang diberikan.

2. Absensi dalam suatu acara itu mengacu pada ketidakhadiran peserta yang diharapkan untuk hadir dalam suatu acara atau pertemuan.

Absensi dalam acara contohnya: peserta yang mendaftar tetapi tidak hadir pada hari acara, atau peserta yang keluar lebih awal sebelum acara berakhir.

3. Dalam dunia kerja absensi merujuk pada ketidakhadiran karyawan di tempat kerja. Absensi yang tidak teratur dapat berdampak negatif pada produktivitas dan kinerja tim secara keseluruhan. Karyawan yang sering absen dapat mengganggu jadwal kerja, proyek kolaboratif, dan menghambat kemajuan organisasi.

Contoh absensi kerja meliputi karyawan yang sering terlambat, sering absen tanpa pemberitahuan, atau tidak menghadiri pertemuan dan acara penting di tempat kerja.

#### **2.1.12 Flask Framework**

*Flask* adalah sebuah *web framework* yang ditulis dengan bahasa *Python* dan tergolong sebagai jenis *microframework*. *Flask* berfungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu web. Dengan menggunakan *Flask* dan bahasa *Python*, pengembang dapat membuat sebuah web yang terstruktur dan dapat mengatur *behaviour* suatu web dengan lebih mudah. *Flask* termasuk pada jenis *microframework* karena tidak memerlukan suatu alat atau pustaka tertentu dalam penggunaannya.

Sebagian besar fungsi dan komponen umum seperti validasi *form*, *database*, dan sebagainya tidak terpasang secara default di *Flask*. Hal ini dikarenakan fungsi dan komponen-komponen tersebut sudah disediakan oleh pihak ketiga dan *Flask* dapat menggunakan ekstensi yang membuat fitur dan komponen-komponen tersebut seakan diimplementasikan oleh *Flask* sendiri. Selain itu, meskipun *Flask* disebut sebagai *microframework*, bukan berarti *Flask* mempunyai kekurangan dalam hal fungsionalitas.

*Microframework* disini berarti bahwa *Flask* bermaksud untuk membuat *core* dari aplikasi ini sesederhana mungkin tapi tetap dapat dengan mudah ditambahkan. Dengan begitu, fleksibilitas serta skalabilitas dari *Flask* dapat dikatakan cukup tinggi dibandingkan dengan *framework* lainnya.

### **2.1.13 HTML**

*HTML (Hyper Text Markup Language)* adalah bahasa yang digunakan untuk menulis halaman web dan menampilkan berbagai informasi di browser Internet. *HTML* merupakan pengembangan dari standar format untuk dokumen teks, yaitu *Standard Generalized Markup Language (SGML)*. *HTML* saat ini merupakan standar Internet yang ditentukan dan dikendalikan untuk digunakan oleh *World Wide WEB Consortium (W3C)*.

*HTML* dikembangkan oleh Tim Berners-Lee saat dia masih bekerja di *CERN (Conseil Europeene Pour La Recherche Nuclaire)* dan dipopulerkan oleh *Mosaic Browser* pada tahun 1989 sebagai cara sederhana namun ampuh untuk menyandikan dokumen elektronik. Padahal, tujuan awal web browser adalah untuk bertindak sebagai pembaca dokumen dalam format *HTML*. Namun dua dekade kemudian, browser tersebut telah menjadi portal ke dunia media online. Oleh karena itu, versi *HTML 2* berikutnya bukan hanya versi *HTML*, tetapi juga standar komprehensif untuk fungsionalitas halaman web..

### **2.1.14 Javascript**

*Javascript* adalah bahasa pemrograman web yang dapat membuat halaman web Anda lebih menarik dan hidup. Bahasa pemrograman JavaScript berjalan di halaman web atau pengunjung, bukan *server*, jadi berbeda dengan *PHP*. *JavaScript* juga dikatakan terpisah dari *HTML* dan *CSS*.

*HTML* mengontrol tampilan konten dan *CSS* mengontrol tata letak, jadi ketiganya sangat berbeda. Banyak yang membandingkan *HTML*, *CSS*, dan *JavaScript* dengan membangun rumah, furnitur di dalamnya, lampu, dan pintu. Tapi sekarang *JavaScript* mulai bergerak di aplikasi tujuan umum seperti *Android* dan *iOS* juga, selain *WEB*. Pengembangan *Javascript* juga memungkinkan pengembang untuk membuat aplikasi *desktop*.

### **2.1.15 Pengujian Black Box**

*Black Box Testing* adalah metode pengujian perangkat lunak di mana struktur internal/desain/implementasi item yang diuji tidak diketahui oleh penguji. Hanya desain dan struktur luar yang diuji.

Pengujian kotak hitam terutama difokuskan pada pengujian fungsionalitas perangkat lunak, memastikan bahwa perangkat lunak tersebut memenuhi persyaratan dan spesifikasi. Pengujian *black box* tidak memerlukan pengetahuan apa pun tentang cara kerja internal perangkat lunak, dan dapat dilakukan oleh penguji yang tidak terbiasa dengan bahasa pemrograman.

Pengujian *black box* menggunakan metode seperti partisi kesetaraan, analisis nilai batas, dan tebakan kesalahan untuk membuat kasus pengujian. Pengujian *black box* mudah digunakan, tidak memerlukan pengetahuan pemrograman dan efektif dalam mendeteksi masalah fungsional. Namun, mungkin ada beberapa cacat internal penting yang tidak terkait dengan fungsionalitas.

Tabel 1 Contoh Pengujian *Black Box*

|                 |                                                                                |                                                                       |
|-----------------|--------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| <i>E-mail</i>   | Benar                                                                          | Benar                                                                 |
| <i>Password</i> | Salah                                                                          | Benar                                                                 |
| <b>Hasil</b>    | <i>E-mail</i> benar tapi <i>password</i> salah, tetap di halaman <i>log in</i> | <i>E-mail</i> dan <i>password</i> benar, diarahkan ke halaman Beranda |

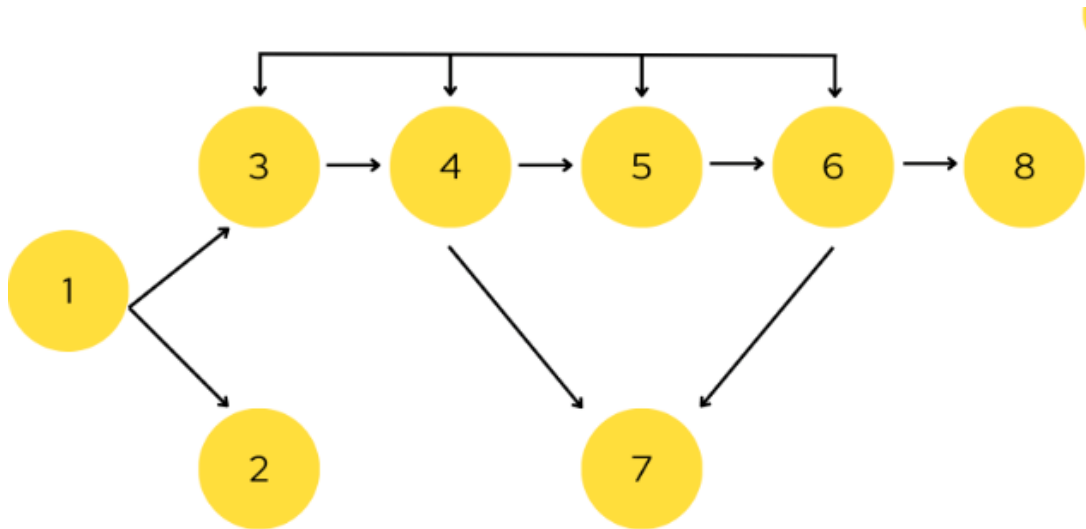
Diatas merupakan contoh pengujian black box dimana jika email dan password yang dimasukkan benar, maka diarahkan ke beranda, tapi jika email atau password yang dimasukkan salah, maka akan tetap di halaman login

#### 2.1.16 Pengujian White Box

*White Box Testing* adalah metode pengujian perangkat lunak di mana struktur internal/desain/implementasi item yang diuji diketahui oleh penguji. Implementasi dan dampak kode ini diuji. Tujuan *White Box* terutama difokuskan untuk memastikan bahwa kode internal perangkat lunak sudah benar dan efisien.

*White box* memerlukan pengetahuan tentang bahasa pemrograman, arsitektur perangkat lunak, dan pola desain. *White Box* digunakan untuk menguji perangkat lunak pada level unit, level integrasi, dan level sistem.

*White Box* efektif dalam mendeteksi cacat internal, dan memastikan bahwa kode tersebut efisien dan dapat dipelihara. Namun, hal ini memerlukan pengetahuan pemrograman dan dapat memakan waktu.



Gambar 8 Contoh Pengujian White Box

Diatas merupakan contoh pengujian *white box* , yang mana berdasarkan diagram di atas, kemungkinan jalur yang dilewati antara lain:

- 1-2
- 1-3-4-5-6-8
- 1-3-4-5-6-7
- 1-3-4-5-6-8 dan sebagainya.

- **Region**

Region dalam konteks pengujian white-box merujuk pada pembagian kode program menjadi beberapa wilayah atau bagian berdasarkan alur eksekusi. Setiap wilayah ini terbentuk dari kombinasi node dan edge dalam graf alur kontrol (control flow graph, CFG).

- **Graf Alur Kontrol (CFG)** adalah representasi diagram dari aliran eksekusi dalam kode program. Node dalam graf mewakili blok kode (instruksi atau pernyataan), sementara edge menunjukkan transisi antara blok kode.
- **Region (Wilayah)** dalam CFG adalah area yang dikelilingi oleh satu atau lebih edge (cabang).
- **Konsep Region** digunakan dalam metode perhitungan **Cyclomatic Complexity**, di mana jumlah wilayah independen dihitung untuk menentukan jumlah jalur independen dalam kode.

- **Cyclomatic Complexity (Kompleksitas Siklomatik)**

Cyclomatic Complexity adalah metrik yang digunakan untuk mengukur kompleksitas sebuah program berdasarkan jumlah jalur eksekusi yang mungkin terjadi dalam kode.

Rumus

- $CC = E - N + 2P$

- $CC = \text{Jumlah Keputusan (Decision Nodes)} + 1$

- **Independent Path (Jalur Independen)**

Jalur independen adalah jalur unik dalam program yang tidak bisa diperoleh dari kombinasi jalur lain. Dengan kata lain, setiap jalur independen merepresentasikan kemungkinan eksekusi yang berbeda dalam kode.

Cara menentukan jalur independen:

1. Buat **graf alur kontrol**.
2. Hitung **Cyclomatic Complexity (V(G))**, yang menunjukkan jumlah jalur independen.
3. Identifikasi **jalur independen** berdasarkan cabang dan percabangan dalam kode.

## 2.2 Penelitian Terdahulu

Tabel 2 Penelitian Terdahulu

| No. | Nama peneliti                                                                                                  | Judul Penelitian                                                                                    | Metodologi Penelitian      | Hasil Penelitian                                                                                                                                                                                                                                             |
|-----|----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.  | I Nyoman Arya<br>Wiradarma, Ayu Manik<br>Dirgayusari, Ni Wayan Wardani<br>(2022)<br><br>STMIK STIKOM INDONESIA | RANCANG BANGUN SISTEM INFORMASI ABSENSI DAN PENGGAJIAN PADA UD. MITRACO UTAMA BERBASIS WEB          | - Observasi<br>- Wawancara | Hasil dari pengujian sistem menunjukan bahwa sistem informasi pengelolaan absensi dan penggajian sudah mapu menghasilkan output yang sesuai dengan perancangan.                                                                                              |
| 2.  | Agit Ardian<br>(2020)<br><br>Universitas Muhammadiyah Magelang                                                 | APLIKASI PENGGAJIAN BERBASIS WEB SERTA PERHITUNGAN PPH 21 STUDI KASUS : ( CV.RESTU JAYA TANGERANG ) | - Observasi<br>- Wawancara | Berdasarkan hasil penelitian, perancangan, pembangunan dan pengujian website aplikasi penggajian berbasis web serta perhitungan pph 21 di CV. Restu Jaya Tangerang, Website ini dapat membantu user membuat lapaoran dan meminimalisir kesalahan input data. |

|    |                                                                                            |                                                                                                                                     |                                                                       |                                                                                                                                                                                                                                                                                                                                                                        |
|----|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3. | Munawir, Liza<br>Fitria,<br>Muhammad<br>Hermansyah<br>(2020)<br><br>Universitas<br>Samudra | IMPLEMENTASI<br>FACE<br>RECOGNITION<br>PADA ABSENSI<br>KEHADIRAN<br>MAHASISWA<br>MENGUNAKAN<br>METODE HAAR<br>CASCADE<br>CLASSIFIER | - Observasi<br><br>- Haar<br>cascade<br>classifier                    | Tingkat akurasi<br>implementasi face<br>recognition pada absensi<br>kehadiran mahasiswa<br>menggunakan metode<br>Haar Cascade Classifier<br>dengan pengujian satu<br>wajah adalah 76% dan<br>pengujian banyak wajah<br>adalah 33.33%. Hasil ini<br>menandakan bahwa<br>sistem pengenalan<br>dengan satu wajah dapat<br>diterapkan pada absensi<br>kehadiran mahasiswa. |
| 4. | Yunita<br>Kurniawati<br>(2019)<br><br>Universitas<br>Semarang                              | SISTEM PRESENSI<br>KELAS<br>MENGUNAKAN<br>PENGENALAN<br>WAJAH DENGAN<br>METODE HAAR<br>CASCADE<br>CLASSIFIER                        | - Observasi<br><br>- Wawancara<br><br>- Haar<br>cascade<br>classifier | Hasil penelitian tersebut<br>ialah pendeteksian wajah<br>menggunakan metode<br>haar cascade classifier<br>ialah sebesar 75% dan<br>Seluruh sistem terbukti<br>dapat berjalan dengan<br>baik dalam mendeteksi<br>serta mengabsen objek<br>yang ada secara tepat.                                                                                                        |

|    |                                                                                                   |                                                                                         |                                                  |                                                                                                                                                                                                                                                             |
|----|---------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|--------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5. | Nurul Afni, Roida Pakpahan, Astri Rezky Jumarah (2019)<br><br>Universitas Bina Sarana Informatika | RANCANG BANGUN SISTEM INFORMASI PENGGAJIAN DENGAN IMPLEMENTASI METODE WATERFALL         | - Observasi<br>- Wawancara<br>- Metode Waterfall | Dari hasil analisis melalui wawancara yang telah dilakukan, diketahui bahwa, sistem penggajian yang berjalan ini masih kurang baik dalam mengolah maupun mengarsipkan gaji sehingga masih kurang efisien cepat dan tepat dalam mengolah data gaji karyawan. |
| 6. | Bagus Eka Pratama (2019)<br><br>Institut Bisnis dan Informatika STIKOM Surabaya                   | RANCANG BANGUN SISTEM INFORMASI PENGGAJIAN PADA PT PRESHON ENGINEERING PLASTIC SURABAYA | - Observasi<br>- Wawancara<br>- Metode Waterfall | Dari hasil analisis melalui wawancara yang telah dilakukan, diketahui bahwa, sistem penggajian yang berjalan ini masih kurang baik dalam mengolah maupun mengarsipkan gaji sehingga masih kurang efisien cepat dan tepat dalam mengolah data gaji karyawan. |

|    |                                                                                                            |                                                                                                                                                         |                                                               |                                                                                                                                                                                                                                          |
|----|------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7. | <p>Sarah Hasna<br/>Shiba<br/>(2018)</p> <p>Institut Bisnis dan<br/>Informatika<br/>STIKOM<br/>Surabaya</p> | <p>RANCANG<br/>BANGUN APLIKASI<br/>ABSENSI DAN<br/>PENGGAJIAN<br/>KARYAWAN PADA<br/>PT MENTARI<br/>SMART GUNA<br/>SIDOARJO<br/>BERBASIS<br/>DESKTOP</p> | <p>- Observasi<br/>- Wawancara<br/>- Metode<br/>Waterfall</p> | <p>Aplikasi ini<br/>menghasilkan laporan<br/>jumlah karyawan yang<br/>masuk dan ijin, laporan<br/>detail status kehadiran<br/>karyawan, laporan gaji<br/>setiap karyawan dan<br/>laporan detail informasi<br/>gaji seluruh karyawan.</p> |
|----|------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|