

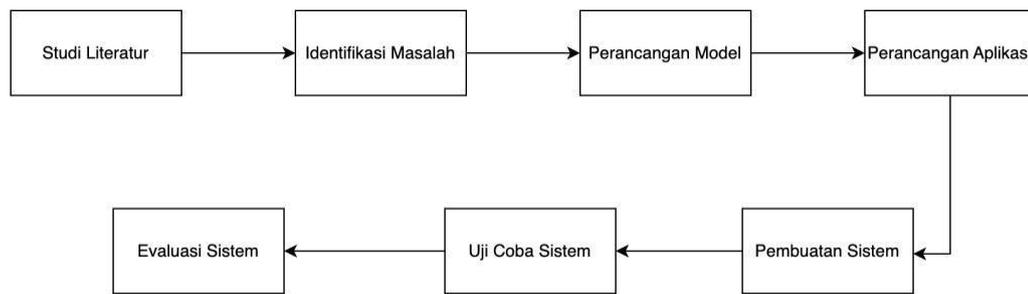
## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1 Metode Penelitian

Dalam penelitian ini, metode penelitian yang dilakukan pada penelitian adalah menggunakan studi pustaka, perancangan, implementasi, dan pengujian. Studi pustaka adalah tahapan dimana peneliti mencari referensi mengenai masalah yang akan diteliti oleh penulis dalam project ini yaitu tentang LBPH, Face Recognition dan implementasi pada web. Tahap perancangan adalah tahap yang dimana peneliti melakukan perancangan sistem dalam menyelesaikan masalah penelitian. Tahap implementasi adalah tahap dimana peneliti melakukan implementasi yang sudah dibuat dan dirancang. Tahap pengujian adalah tahap dimana peneliti sudah melakukan implementasi sistem dan penelitian sudah siap untuk diuji.

Alur dari penelitian ini adalah tahap pertama melakukan studi literatur dengan mempelajari hal-hal yang berkaitan dengan Face Recognition dan Local Binary Pattern Histogram. Dan juga melakukan pembelajaran terhadap *framework* Django dan Vuejs sebagai salah satu *framework* yang digunakan untuk pembuatan aplikasi. Tahap kedua yaitu mengumpulkan data untuk melakukan percobaan untuk face recognition dengan cara melakukan pengambilan gambar manual dan mengumpulkan data-data untuk dijadikan sebagai pengguna. Tahap ketiga yaitu melakukan perancangan model untuk memastikan setiap tahap dari penelitian dibuat sesuai dengan arahan dari model sehingga tidak melenceng dari model. Tahap keempat yaitu tahap perancangan aplikasi, yang terdapat beberapa tambahan uji sistem bagaimana integrasi antara sistem *backend* and frontend dan perubahan data pada database Sebagai tahap akhir dari penelitian. Flow terdapat gambar gambar 3.1



Gambar 3. 1 Flow Penelitian

### 3.2 Studi Literatur

Studi pustaka yang dilakukan oleh peneliti adalah dengan mempelajari jurnal yang berkaitan dengan *Face Recognition* dan *Local Binary Pattern Histogram* dan implementasi pada *web based*. Peneliti juga melakukan studi pustaka melalui video pada platform *Youtube*. Peneliti melakukan studi pustaka untuk mempelajari bahasa pemrograman bahasa Python dan Javascript dengan *framework* yang sudah ditentukan. Peneliti juga melakukan survei beberapa aplikasi *face recognition*. Selanjutnya pada tahap perancangan akan dibahas pada sub-bab dari bab III dan implementasi dari sistem beserta hasil pengujian akan dibahas pada bab IV

### 3.3 Identifikasi Masalah

Pada tahap ini, mengidentifikasi permasalahan yang muncul pada PT.XYZ dilakukan dengan cara melakukan interview menggunakan aplikasi kirim pesan bersama salah satu *staf HRD* pada PT.XYZ. Wawancara tersebut dilakukan untuk mengetahui permasalahan yang ada pada PT.XYZ perihal sistem absensi yang sedang digunakan pada PT.XYZ apakah berfungsi sesuai dengan harapan dari karyawan atau tidak. Pembahasan dari wawancara yang dilakukan yaitu tentang masalah efisiensi dari sistem absensi, masalah manipulasi absensi dan rekap absensi. Setelah mendapatkan jawaban dari hasil wawancara yang dilakukan dengan salah satu staf HRD pada PT.XYZ maka akan di lakukan pembuatan kuisisioner untuk mengukur pendapat dari karyawan untuk menilai sistem absensi yang sedang digunakan oleh karyawan. Kuisisioner tersebut berisi pertanyaan-pertanyaan yang berhubungan dengan sistem absensi yang digunakan , dengan

jawaban *linear* yang dibuat dengan menggunakan aplikasi *form* yang kemudian nanti akan dibagikan ke karyawan. Maka dari itu dilakukan pembagian kuisisioner untuk karyawan yang bekerja pada PT.XYZ. Kuisisioner tersebut dilakukan untuk melakukan apakah karyawan mengalami masalah yang serupa dengan hasil interview tentang sistem absensi yang digunakan pada PT.XYZ. Hasil dari interview dan pembagian kuisisioner didapatkan hasil yang sesuai antara interview dan kuisisioner maka dapat disimpulkan bahwa karyawan mengalami masalah dengan sistem absensi yang sedang digunakan. Hasil dari kuisisioner maka dijadikan salah satu acuan untuk pembuatan sistem absensi yang baru dengan beberapa fitur tambahan sesuai dengan pertanyaan pada kuisisioner yang ditanyakan. Untuk mengembangkan sistem absensi pada PT.XYZ maka akan dilakukan pembuatan sistem absensi dengan *face recognition* yang dapat mempermudah dan dapat menjawab dari hasil interview dan kuisisioner dari karyawan.

### 3.4 Perancangan Model

Pada perancangan model pembuatan sistem absensi dengan menggunakan biometrik face recognition dengan menggunakan metode pengolahan citra *local binary pattern histogram*. Proses dari absensi dengan menggunakan face recognition dilakukan dengan penerimaan data berupa file image ketika melakukan Registrasi absensi, absensi masuk dan absensi keluar.

- Pengambilan Gambar Sebelum Proses

Pada gambar berikut menampilkan tentang pengambilan dan proses gambar yang diambil.

```
image_file = request.FILES['image']
nik = request.data['nik']
image_data = image_file.read()

try:
    message, status_code = detect_faces(image_data, settings.BASE_DIR, nik)
    return JsonResponse({'message': message, 'status': status_code}, status=status_code)
```

Gambar 3. 2 Proses Ambil gambar

Apada gambar disebutkan bahwa sistem akan menerima *image\_data* akan melaukan pengambilan data lalu akan dikirimkan untuk di proses dengan *function*

detect\_faces.

- Pendeteksian Wajah

Pada *function detect\_faces* proses gambar akan dilakukan penamaan file dan melakukan penyimpanan gambar dengan hasil gambar yang sudah di proses grayscale pada gambar.

```
def train_lbph_model(nik):
    faces = FaceImage.objects.filter(nik=nik)
    labels = []
    images = []

    for face in faces:
        image = cv2.imread(face.image.path)
        gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        images.append(gray)
        labels.append(face.nik)

    labels = np.array(labels)

    lbph_model = cv2.face.LBPHFaceRecognizer_create()
    lbph_model.train(images, labels)

    lbph_model_path = os.path.join(settings.BASE_DIR, 'trained_models', f'lbph_model_{nik}.yml')
    lbph_model.save(lbph_model_path)
```

Gambar 3. 3 Proses Train LBPH

Pada gambar diatas Fungsi `train_lbph_model` mengambil ID pengenalan (NIK) sebagai argumen, memuat gambar wajah dari database yang sesuai dengan ID tersebut. Proses “for face in faces” adalah melakukan pencarian wajah dari gambar yang sudah diterima kemudian akan melakukan pemotongan terhadap gambar yang hanya terdapat wajah. Pencarian wajah dilakukan dengan proses melakukan *grayscale* pada gambar yang diterima kemudian sistem akan melakukan pendeteksian terhadap wajah pada gambar.

- Proses LBPH

Setelah menemukan wajah pada pendeteksian wajah kemudian sistem akan melatih model pengenalan wajah menggunakan algoritma LBPH (Local Binary Patterns Histograms) menggunakan gambar wajah tersebut, dan akhirnya menyimpan model yang dilatih ke dalam file YAML yang sesuai dengan ID pengenalnya. Objek model pengenalan wajah menggunakan algoritma LBPH (Local

Binary Patterns Histograms) dengan memanggil `cv2.face.LBPHFaceRecognizer_create()`. `cv2.face.LBPHFaceRecognizer_create()` adalah metode yang digunakan untuk membuat objek model pengenalan wajah menggunakan algoritma LBPH (Local Binary Patterns Histograms) dalam library OpenCV. Saat metode ini dipanggil, ia menginisialisasi objek model yang akan digunakan untuk melatih dan mengenali wajah menggunakan metode LBPH. Di dalamnya, proses pembuatan model melibatkan pengaturan konfigurasi default untuk model LBPH dan alokasi sumber daya yang diperlukan untuk operasi pengenalan wajah. Metode ini mengembalikan objek model yang sudah siap digunakan, yang kemudian bisa dilatih dengan menggunakan gambar wajah dan label yang sesuai. Kemudian, model tersebut dilatih dengan menggunakan metode `train`, yang menerima dua argumen: images yang berisi gambar wajah dalam skala keabuan (grayscale) dan labels yang berisi label yang sesuai dengan setiap gambar wajah. Selanjutnya, lokasi untuk menyimpan model yang dilatih ditentukan menggunakan `os.path.join` dengan direktori dasar dari proyek Django, yang menghasilkan jalur lengkap ke direktori "trained\_models" dengan nama file yang sesuai dengan ID pengenalan (NIK) dalam format YAML. Terakhir, model yang sudah dilatih disimpan menggunakan metode `save` pada objek model dengan jalur yang telah ditentukan sebelumnya.

- Proses Face Recognition

```
recognizer = cv2.face.LBPHFaceRecognizer_create()

model_path = os.path.join(base_dir, 'trained_models', f'lbph_model_{nik}.yaml')
recognizer.read(model_path)

for idx, (x, y, w, h) in enumerate(faces):
    face_region = gray[y:y + h, x:x + w]

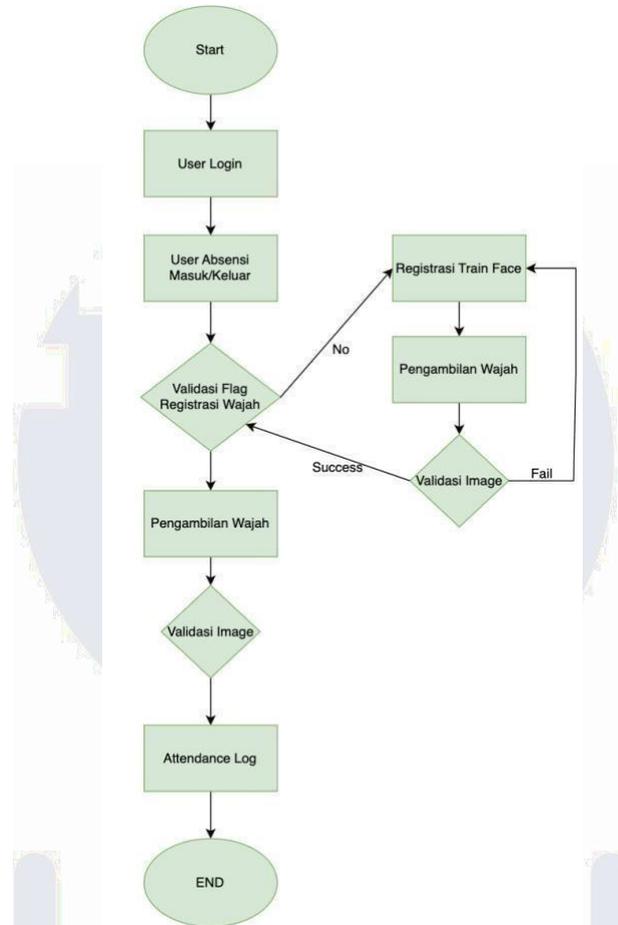
    label, confidence = recognizer.predict(face_region)
```

*Gambar 3. 4 Proses LBPH Recognizer*

Pada proses sesuai dengan gambar diatas yaitu foto yang didapatkan dari menu absensi akan dilakukan proses seperti sebelumnya terlebih dahulu yaitu proses

pengambilan wajah, pendeteksian wajah dan proses LBPH. setelah mendapatkan file yam1 dari hasil gambar yang didapatkan dari proses absensi maka akan dilakukan perbandingan gambar wajah antara hasil gambar yang didapatkan dari menu absensi dan hasil gambar yang didapatkan pada saat registrasi wajah. Pada gambar di atas adalah bagian dari proses pengenalan wajah menggunakan metode Local Binary Patterns Histograms (LBPH). Pertama, sebuah objek pengenalan wajah LBPH dibuat menggunakan fungsi `cv2.face.LBPHFaceRecognizer_create()`. Jalur file untuk model yang telah dilatih kemudian ditentukan menggunakan `os.path.join()`, yang menggabungkan direktori dasar, subdirektori, dan nama file model yang spesifik untuk pengguna berdasarkan NIK. Model ini dibaca ke dalam objek pengenalan menggunakan metode `recognizer.read()`. Kemudian, model LBPH digunakan untuk memprediksi identitas dari wajah yang diekstrak dengan fungsi `recognizer.predict()`, yang menghasilkan label prediksi dan nilai kepercayaan (confidence) dari prediksi tersebut. Proses ini memungkinkan sistem untuk mengenali wajah-wajah dalam gambar berdasarkan model yang telah dilatih sebelumnya. Sehingga dari proses yang didapatkan mendapatkan hasil nilai kepercayaan ( confidence ) yang digunakan sebagai nilai untuk membuktikan bahwa gambar yang diambil sudah cocok atau tidak.

### 3.4.1 Perancangan Flow Chart Sistem Absensi



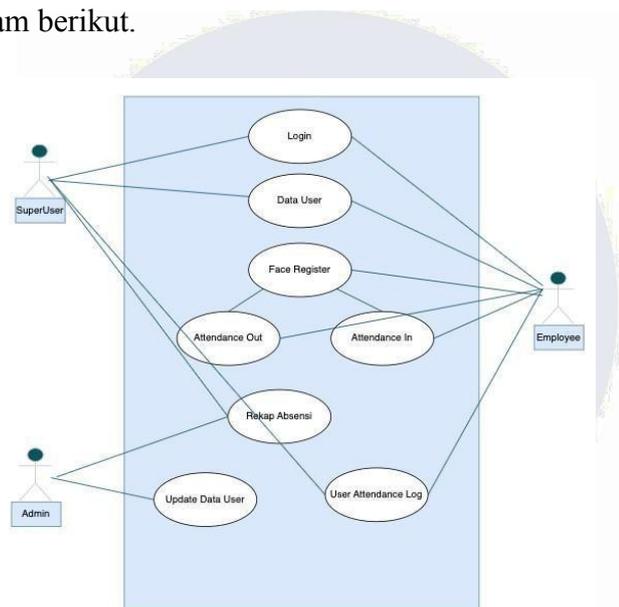
Gambar 3. 5 Flow Chart Sistem Absensi

Dari gambar 3.2 diatas pada perancangan flow chart sistem absensi disini membutuhkan setiap user untuk melakukan login dengan nik dan password yang sudah di registrasikan oleh setiap admin. Sebelum user dalam melakukan absensi masuk atau keluar setiap user diwajibkan untuk melakukan registrasi *train face* yaitu melakukan registrasi wajah setelah itu user dapat melakukan user attendance masuk atau keluar dengan cara melakukan pengambilan gambar wajah yang sudah di deteksi dan jika berhasil maka setiap user dapat melihat hasil dari absensinya. Sebuah flag akan di jadikan salah satu penanda apakah user sudah melakukan registrasi

atau belum. Jika flag yang terbaca false maka user tidak dapat melakukan absensi masuk atau keluar dan akan di arahkan untuk melakukan registrasi wajah terlebih dahulu, setelah melakukan registrasi wajah maka flag akan menjadi true dan user dapat melakukan absensi masuk ataupun keluar.

### 3.4.2 Perancangan Use Case Diagram

Hubungan user dan usecase pada sistem yang berjalan dapat dilihat pada *use case diagram* berikut.



Gambar 3. 6 Use Case Diagram Pada Sistem

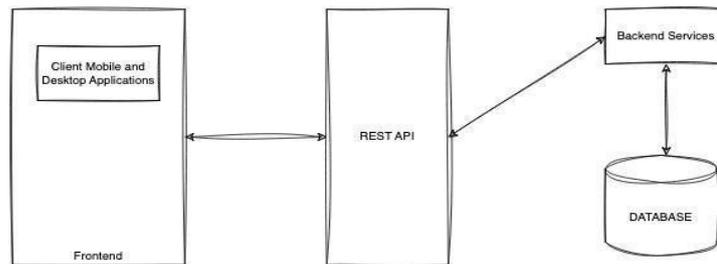
Pada gambar 3.3 dijelaskan bahwa terdapat 3 bagian yaitu Superuser, Admin dan Employee. Pada bagian Superuser tersebut dibagi lagi menjadi beberapa *role* yaitu dalamnya terdapat role yang masing-masing mempunyai akses tertentu role tersebut yaitu *Coordinator, Junior Manager, Senior Manager, General Manager* dan *Admin*.

### 3.5 Perancangan Sistem

Dalam penelitian ini, membutuhkan data dalam berjalannya sistem yang akan dibuat dalam penelitian ini. Data berupa data user dan data image, data tersebut akan diproses oleh sistem untuk mendapatkan hasil dalam penelitian ini. Pengambilan data user didapat dari registrasi secara manual menggunakan fungsi yang telah disediakan oleh sistem *backend* dan pengambilan data image didapatkan

dari pengambilan manual menggunakan kamera smartphone/laptop data yang dibuat berupa data wajah menggunakan masker, kacamata, dan tidak menggunakan kacamata dan masker.

Perancangan aplikasi dari sistem yang pertama adalah pada pembentukan Frontend. Frontend dari aplikasi akan menggunakan bahasa pemrograman Javascript dengan menggunakan *framework* *Vuejs* dengan tambahan *Tailwind css*. Pembuatan aplikasi pada penelitian ini berbasis pada web dengan menggunakan Single Page Application. Pada frontend melakukan komunikasi dengan *backend* sebagai salah satu sistem yang mengatur untuk mendapatkan data menggunakan *library* *axios* yang berfungsi untuk memanggil REST API yang telah dibuat oleh *backend*. Proses integrasi antara frontend dan *backend* dalam perancangan aplikasi ini memainkan peran yang penting. Setelah frontend aplikasi selesai dibuat menggunakan bahasa pemrograman JavaScript dan *framework* *Vue.js* dengan tambahan Tailwind CSS, langkah selanjutnya adalah melakukan komunikasi dengan *backend*. *Backend* berperan sebagai sistem yang mengatur pengolahan dan penyediaan data yang dibutuhkan oleh frontend. Dalam hal ini, komunikasi antara frontend dan *backend* dilakukan melalui penggunaan library *Axios*. Library ini memungkinkan frontend untuk melakukan panggilan ke REST API yang telah dibuat di *backend*. Dengan bantuan *Axios*, frontend dapat mengirim permintaan ke *backend* dan menerima respons yang sesuai. Proses ini memungkinkan aplikasi untuk berinteraksi dengan data yang disediakan oleh *backend*, sehingga memberikan pengalaman pengguna yang dinamis dan responsif. Contoh dari integrasi antara *Frontend*, *Backend* dan *Database* ada pada gambar 3.4.



Gambar 3. 7 Integrasi Frontend, Backend dan Database

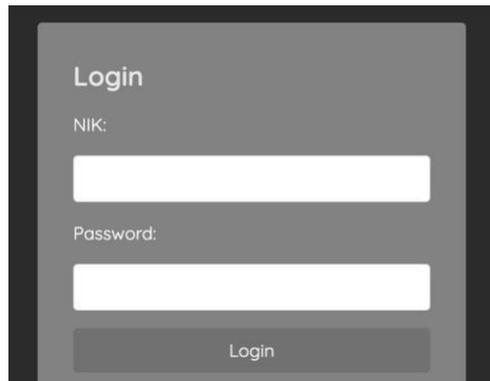
Dengan pendekatan *Single Page Application* (SPA) yang digunakan dalam pengembangan aplikasi ini, pengguna dapat menjelajahi berbagai fitur dan konten tanpa perlu melakukan *reload* halaman. Hal ini memberikan pengalaman pengguna yang lebih mulus dan interaktif dalam menggunakan aplikasi web yang dirancang. Pada aplikasi hanya digunakan oleh pengguna ketika ingin melakukan absen masuk dan absen keluar beserta dengan catatan absen. Dalam menentukan use case yang merupakan deskripsi dari urutan cara-cara yang menghasilkan hasil aksi untuk user.

### 3.5.1 Perancangan Tampilan Antarmuka

Pada perancangan pertama akan dibuat *mock up* page yang akan di gunakan oleh setiap user untuk menggunakan website absensi yang dapat digunakan untuk melakukan absensi dan melihat rekap absensi dari setiap user. Berikut adalah halaman-halaman *mockup* yang dapat yang diakses oleh user

#### 3.5.1.1 Halaman Login

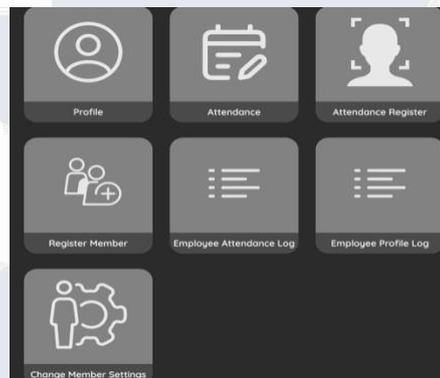
Pada halaman login akan terdapat 2 form input yaitu nik dan password yang akan menjadi parameter untuk fungsi login. Login akan berhasil jika pengguna memasukkan nik dan password yang sudah terdaftar pada database, ketika pengguna melakukan *input* dengan benar pada nik dan password maka page akan melakukan *route* menuju page Home.



*Gambar 3. 8 Login Mockup*

### **3.5.1.2 Halaman Home**

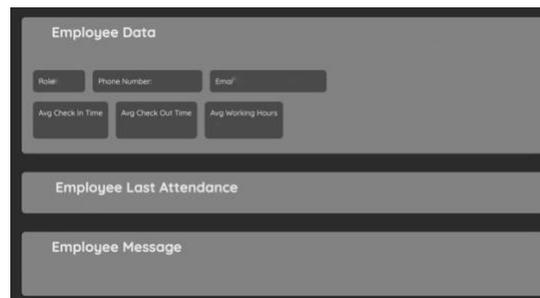
Pada Halaman Home terdapat beberapa menu yang akan muncul tergantung sesuai dengan role yang didapatkan oleh setiap user . menu yang utama yaitu registrasi wajah dan absensi masuk atau keluar.



*Gambar 3. 9 Home Mockup*

### **3.5.1.3 Halaman Dashboard User**

Halaman Dashboard berisi beberapa info dari profile user dan ada beberapa info yang berkaitan dengan status absensi dari setiap user.



Gambar 3. 10 Dashboard Mockup

### 3.5.1.4 Register Member

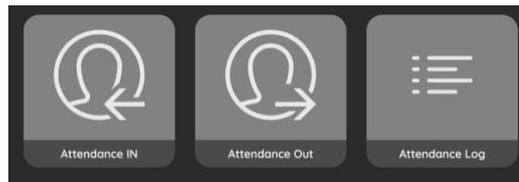
Halaman Register Member adalah halaman yang berguna digunakan oleh admin untuk melakukan registrasi untuk user dapat melakukan masuk kedalam sistem.

A registration form mockup titled "Registration". It contains several input fields: "NIK", "Full Name", "Phone Number", "Date of Birth", "Place of Birth", "Email", "Password", "Role ID", and "Department ID". A "Register" button is located at the bottom right of the form.

Gambar 3. 11 Regist Mockup

### 3.5.1.5 Attendance

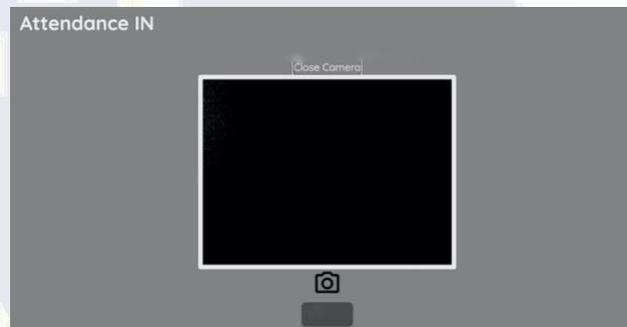
Halaman attendance adalah halaman pada yang ada pada home yang merupakan menu untuk melihat menu *attendance* dan *attendance log*



Gambar 3. 12 Attendance Mockup

### 3.5.1.6 Attendance In

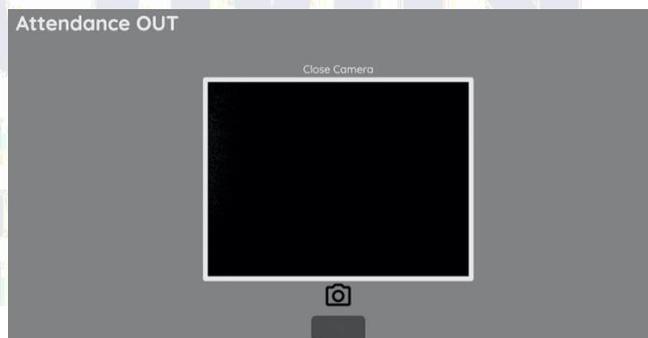
Halaman Attendance In adalah halaman yang digunakan oleh setiap user untuk melakukan absensi masuk dengan cara melakukan *face recognition*



Gambar 3. 13 Attendance In Mockup

### 3.5.1.7 Attendance Out

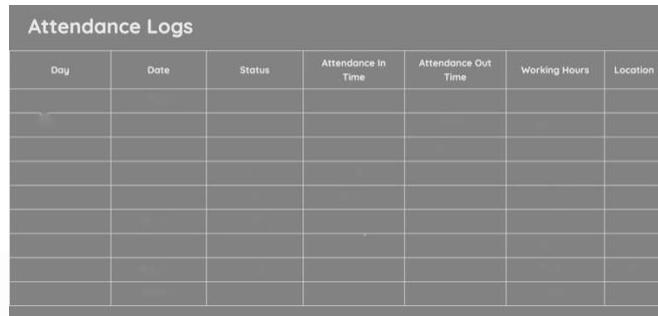
Halaman Attendance In adalah halaman yang digunakan oleh setiap user untuk melakukan absensi keluar dengan cara melakukan *face recognition*



Gambar 3. 14 Attendance Out Mockup

### 3.5.1.8 Attendance Log

Halaman Attendance Log adalah halaman yang menampilkan hasil absensi log dari setiap user



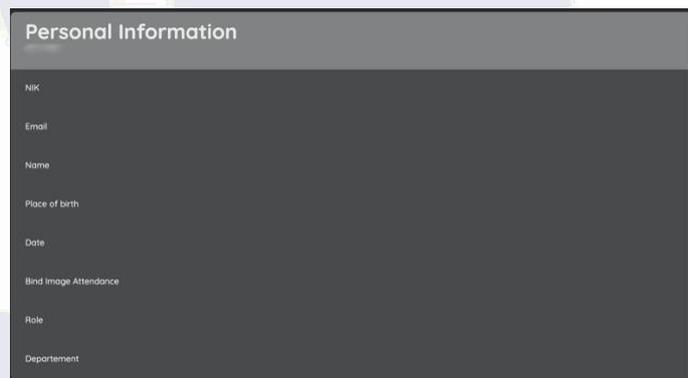
A mockup of an 'Attendance Logs' table. The table has a dark grey header with the title 'Attendance Logs' and seven columns: 'Day', 'Date', 'Status', 'Attendance In Time', 'Attendance Out Time', 'Working Hours', and 'Location'. The table body consists of several empty rows, indicating a list of attendance records.

Day	Date	Status	Attendance In Time	Attendance Out Time	Working Hours	Location

Gambar 3. 15 Attendance Log Mockup

### 3.5.1.9 Profile

Halaman Profile adalah halaman yang menampilkan semua data dari user.



A mockup of a 'Personal Information' profile form. The form has a dark grey header with the title 'Personal Information'. Below the header, there are several input fields for user data: 'NIK', 'Email', 'Name', 'Place of birth', 'Date', 'Bind Image Attendance', 'Role', and 'Departement'.

Personal Information
NIK
Email
Name
Place of birth
Date
Bind Image Attendance
Role
Departement

Gambar 3. 16 Profile Mockup

### 3.5.1.10 Attendance Register

Halaman Attendance Register adalah halaman yang digunakan user untuk mendaftarkan wajah mereka kedalam sistem sebelum melakukan absensi masuk atau keluar.



*Gambar 3. 17 Attendance Register Mockup*

### **3.5.1.11 Employee Attendance Log**

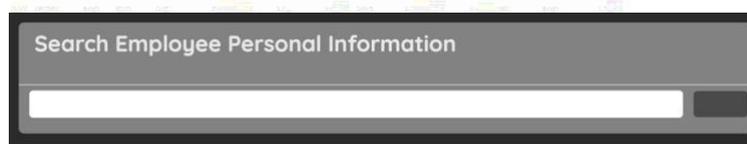
Halaman Employee Attendance Log adalah halaman yang menampilkan menampilkan rekap absensi dari user , menu ini hanya dapat dibuka oleh beberapa role yang sudah ditentukan.



*Gambar 3. 18 Employee Attendance Log Mockup*

### **3.5.1.12 Employee Personal Information**

Halaman Employee Personal Information adalah halaman yang digunakan oleh user dengan role tertentu untuk mencari profile dari setiap user.



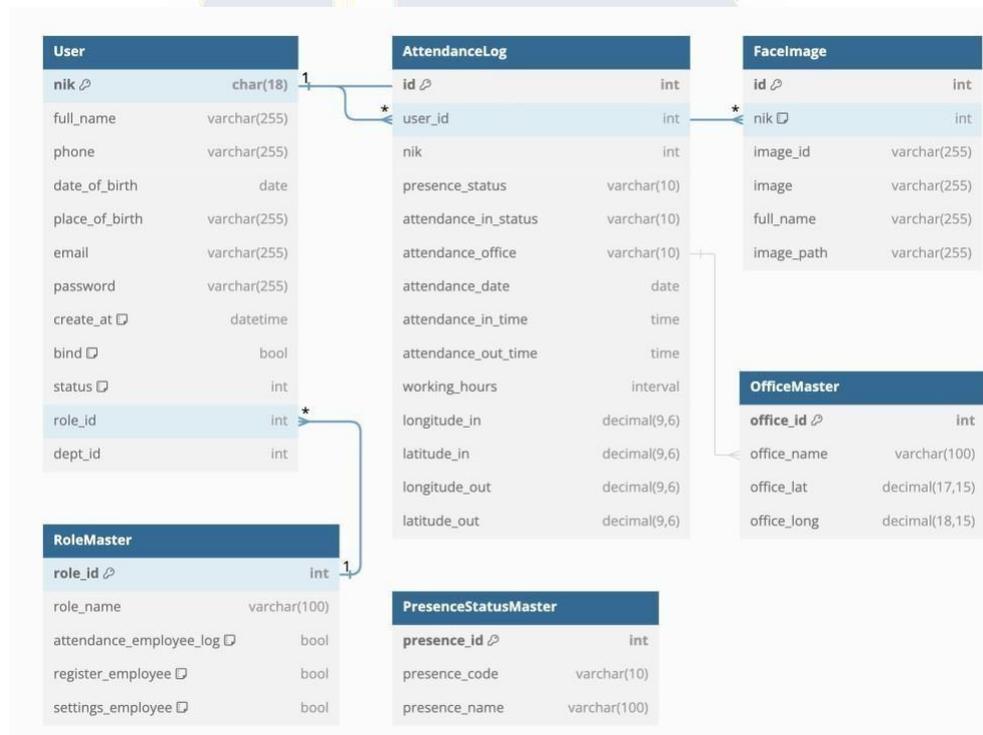
*Gambar 3. 19 Employee Personal Information Mockup*

### **3.5.1.13 Change Settings Member**

Halaman Change Settings Member digunakan oleh admin untuk mengubah profile atau role dari setiap user.

### 3.5.2 Perancangan Tabel Data

Pada perancangan tabel data akan dibuat beberapa tabel yang akan digunakan pada sistem absensi yaitu User, AttendanceLog, FaceImage, RoleMaster, PresenceStatusMaster, dan OfficeMaster.



Gambar 3. 20 Tabel Database sistem

### 3.6 Uji Coba Sistem

Uji coba sistem dilakukan dengan cara melakukan kunjungan ke PT.XYZ. Karyawan mencoba melakukan registrasi dan juga beberapa role yang sudah ditentukan serta melakukan absensi dengan menggunakan *face recognition*. Pada saat uji coba sistem akan dilakukan dokumentasi dan pembagian kuisioner setelah uji coba sistem.

### 3.7 Evaluasi Sistem

Pada tahap evaluasi sistem akan dilakukan beberapa uji dan analisa dari hasil implementasi sistem absensi yang telah dibuat yaitu dengan melakukan uji statistik dan analisa hasil. Pada tahap awal akan dilakukan untuk mendapatkan hasil evaluasi dari karyawan yang telah mencoba sistem absensi *face recognition* yang baru diterapkan sebagai salah satu sistem absensi uji coba . setelah melakukan uji coba sistem absensi maka setelah proses tersebut akan dilakukan pembagian kuisisioner kembali dengan kuisisioner yang berbeda tetapi mempunyai inti pertanyaan yang sama dengan kuisisioner yang sebelumnya dibuat pada identifikasi masalah. Setelah mendapatkan hasil dari kuisisioner uji coba maka akan dilakukan perhitungan untuk mendapatkan jumlah sample minimal yang akan di uji pada uji statistik dan analisa hasil dari kedua kuisisioner.

Sebelum melakukan uji statistik akan dilakukan perhitungan untuk mendapatkan jumlah minimal sampel yang diambil dari kuisisioner. Pada penelitian ini akan dilakukan perhitungan pengambilan sampel dengan menggunakan rumus *clovin*.

$$n = \frac{N}{1 + Ne^2}$$

Rumus 3. 21 rumus clovin menghitung sampel

- n = Jumlah Sample
- N = Jumlah Populasi
- e = persentase batas toleransi (margin of error)

Uji statistik akan dilakukan dengan melakukan uji antara kedua kuisisioner yaitu kuisisioner sebelum uji coba dan kuisisioner setelah uji coba. Uji statistik

dilakukan yaitu dengan mencari nilai signifikansi dari kedua kuisisioner. Metode untuk mendapatkan nilai signifikansi adalah yaitu dengan melakukan uji *wilcoxon signed test* yaitu merupakan uji nonparametris yang digunakan untuk mengukur perbedaan antara 2 kelompok data berpasangan berskala ordinal atau interval. Dasar pengambilan keputusan dalam uji *wilcoxon signed test* adalah sebagai berikut :

- a. Ketika nilai probabilitas *Asym.sig 2 failed*  $< 0,05$  maka terdapat perbedaan rata-rata
- b. Ketika nilai probabilitas *Asym.sig 2 failed*  $> 0.05$  maka tidak terdapat perbedaan rata-rata

Tujuan dari melakukan uji signifikansi tersebut adalah untuk mendapatkan hasil uji dari kuisisioner sebelum uji coba dan sesudah uji coba terdapat perbedaan yang signifikan atau tidak signifikan. Beberapa bagian yang akan dilakukan untuk pengujian adalah kemudahan dalam melakukan absensi, manipulasi data, dan rekap absensi.