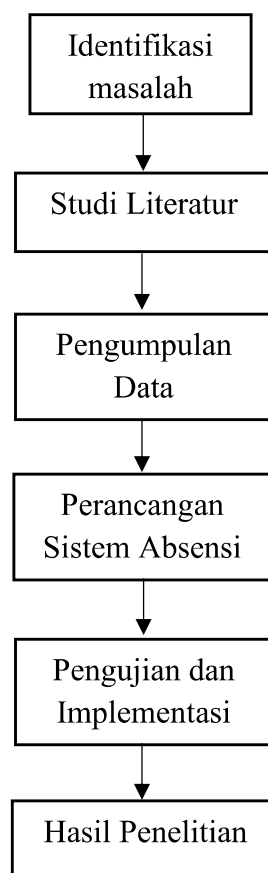


BAB III

METODE PENELITIAN

3.1 Desain Penelitian

Desain penelitian adalah sebuah rancangan dalam penelitian yang bertujuan untuk memberikan pedoman dalam melaksanakan proses penelitian. Desain penelitian menyediakan kerangka dan alur kerja mencakup sepanjang proses penelitian. Berikut ini adalah desain penelitian dalam penelitian ini.



Gambar 3. 1 Desain Penelitian

Berikut penjelasan untuk desain penelitian.

1. Identifikasi masalah

Tahap ini bertujuan untuk mengidentifikasi permasalahan nyata di lapangan, yaitu kurangnya efisiensi dan potensi kecurangan dalam proses absensi manual. Beberapa permasalahan seperti absensi titip atau pencatatan yang tidak valid menjadi dasar perlunya sistem absensi otomatis berbasis biometrik wajah.

2. Studi Literatur

Tahap ini dilakukan pengumpulan referensi ilmiah seperti jurnal, buku, dan laporan penelitian terdahulu yang berkaitan dengan teknologi *face recognition*, algoritma CNN, serta implementasi sistem absensi digital. Studi literatur bertujuan memperkuat landasan teori dan memberikan gambaran tentang metode yang tepat untuk digunakan.

3. Pengumpulan data

Pengumpulan data dilakukan dengan mengambil citra wajah langsung secara real-time sebelum dilakukan deteksi wajah. Dataset wajah yang digunakan merupakan komponen penting dalam pelatihan model pengenalan wajah.

4. Perancangan sistem absensi

Sistem dirancang menggunakan bahasa pemrograman *Python* dengan bantuan library *OpenCV* untuk pengolahan citra dan deteksi wajah. Proses encoding wajah menggunakan CNN dan *Haar Cascade*, kemudian data disimpan

dalam format CSV atau database ringan. Rancangan sistem mencakup antarmuka pengguna dan arsitektur backend yang menangani proses identifikasi.

5. Pengujian dan implementasi

Sistem akan diuji dengan berbagai posisi wajah seperti menghadap depan, menyamping, dan menunduk untuk mengukur tingkat akurasi berdasarkan jumlah dataset yang diperoleh. Proses implementasi dilakukan dengan mengintegrasikan seluruh komponen sistem dan menguji fungsionalitasnya dalam kondisi nyata. Pengujian bertujuan untuk mengetahui apakah sistem dapat mengenali wajah dengan benar dan mencatat kehadiran secara otomatis.

6. Hasil Penelitian

Hasil akhir dari penelitian ini berupa sistem absensi digital berbasis face recognition yang dapat mendeteksi kehadiran secara otomatis. Laporan hasil penelitian disusun berdasarkan pengamatan, pengujian, dan analisis akurasi sistem, yang kemudian disesuaikan dengan permasalahan awal untuk membuktikan bahwa metode CNN efektif dalam mendukung transformasi digital absensi.

3.2 Metode Pengumpulan Data

Dalam penelitian ini, pengumpulan data dilakukan untuk memperoleh dataset wajah yang digunakan sebagai data latih dan data uji dalam sistem absensi otomatis berbasis pengenalan wajah menggunakan *Convolutional Neural Network* (CNN). Adapun jenis data yang dikumpulkan bersifat primer, yaitu data citra wajah yang

diambil secara langsung dari individu yang akan digunakan dalam sistem. Proses pengumpulan data dilakukan melalui beberapa tahapan sebagai berikut.

1. Teknik Pengambilan Data Wajah

Pengambilan data dilakukan dengan menggunakan kamera laptop atau webcam eksternal untuk merekam wajah subjek secara langsung. Proses ini dilakukan dalam ruangan dengan pencahayaan yang cukup untuk memastikan kualitas gambar yang optimal. Setiap individu (misalnya mahasiswa, dosen, atau staf) akan difoto sebanyak 50-200 kali dengan beragam ekspresi, sudut pandang (frontal, miring), dan pencahayaan berbeda untuk meningkatkan keberagaman data dan ketahanan model terhadap variasi wajah. Berikut gambar 3.2 yang digunakan untuk mengambil dan mendaftarkan dataset wajah.

```
1 import cv2
2 import os
3 import sys
```

Gambar 3. 2 Library di register_face.py

Dimulai dengan memanggil library yang terdiri dari *cv2* memanggil modul *OpenCV*. *OpenCV* atau *Open Source Computer Vision* adalah sebuah pustaka (library) yang menyediakan berbagai perangkat lunak, database, serta plugin yang telah dirancang untuk memudahkan pengembangan aplikasi berbasis visi komputer. *OpenCV* menjadi salah satu toolkit yang paling banyak digunakan karena memiliki komunitas pengembang yang luas dan aktif, serta terbukti mampu diterapkan pada berbagai kebutuhan industri. Library ini mendukung bahasa pemrograman seperti C/C++, Python, dan Java, serta kompatibel di berbagai platform, baik komputer

desktop maupun perangkat mobile (Hasan & Sallow, 2021). *Os* yang berfungsi untuk membuat dan memanipulasi path direktori, dan *sys* agar argumen dari *command line* dapat dibaca.

```

5  # ✅ Ambil nama dari GUI melalui argumen
6  if len(sys.argv) < 2:
7      print("❌ Nama tidak diberikan. Gunakan: python register_face.py <nama>")
8      sys.exit()
9
10 name = sys.argv[1]
11
12 # 🛡️ Bersihkan nama dari karakter ilegal
13 karakter_tidak_valid = ['<', '>', ':', '"', '/', '\\', '|', '?', '*', '&']
14 for char in karakter_tidak_valid:
15     name = name.replace(char, "")
16 name = name.strip().replace(" ", "_")
17
18 if name == "":
19     print("❌ Nama tidak valid.")
20     sys.exit()
21
22 path = f"dataset/{name}"
23 if not os.path.exists(path):
24     os.makedirs(path)

```

Gambar 3. 3 Code untuk penyimpanan dan normalisasi nama/label

Setelah memanggil library yang diperlukan, Gambar 3.3 berikut ini diperlukan untuk memberikan label pada citra yang diambil, mengecek apakah label atau nama sudah dimasukkan dan terdaftar atau belum di database, jika nama sudah terdaftar akan di normalisasi dengan menghilangkan karakter yang tidak valid kemudian diubah menjadi spasi atau *underscore*.

```

26 cam = cv2.VideoCapture(0)
27 face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + "haarcascade_frontalface_default.xml")

```

Gambar 3. 4 Code untuk Mengakses kamera

Mengakses kamera default pada perangkat, dalam penelitian ini peneliti menggunakan laptop maka yang diakses adalah webcam bawaan dari laptop itu

sendiri, dan untuk mendeteksi wajah digunakan file *Haar Cascade* yang merupakan file bawaan dari *OpenCV*.

2. Kriteria Data yang Dikumpulkan

Agar data wajah yang dikumpulkan relevan dan dapat digunakan dalam pelatihan model CNN, ditentukan beberapa kriteria sebagai berikut:

1. Wajah terlihat jelas dan tidak terhalang (misalnya oleh masker atau rambut).
2. Ukuran gambar disesuaikan dengan kebutuhan model CNN, misalnya 100x100 piksel.
3. Format file gambar adalah .jpg.
4. Setiap gambar diberi label sesuai identitas pemiliknya (nama atau ID).

3.3 Metode Perancangan

Dalam perancangan sistem absensi berbasis face recognition ini, metode yang digunakan adalah pendekatan berbasis deep learning dengan algoritma *Convolutional Neural Network* (CNN) dengan bantuan algoritma *Haar Cascade*. Perancangan sistem melibatkan beberapa tahap utama, yaitu *preprocessing* citra wajah, integrasi dan pengelolaan data, transformasi data ke dalam bentuk yang dapat dikenali oleh model CNN, serta tahap klasifikasi wajah untuk melakukan proses identifikasi. Setiap tahapan dirancang secara sistematis untuk memastikan akurasi dan efisiensi sistem dalam mengenali wajah secara otomatis sebagai bentuk transformasi digital dalam proses absensi.

1. Preprocessing Data

Preprocessing dilakukan untuk meningkatkan kualitas data citra sebelum digunakan dalam pelatihan model CNN. Tahap ini dimulai dengan proses deteksi wajah menggunakan algoritma *Haar Cascade* yang berfungsi untuk menemukan area wajah pada gambar. Setelah wajah terdeteksi, citra dipotong (*cropping*) untuk hanya menyertakan bagian wajah dan mengurangi noise dari latar belakang. Selanjutnya, citra wajah diubah ukurannya secara seragam (*resizing*), misalnya ke resolusi 100x100 piksel, untuk memudahkan proses pelatihan. Terakhir, proses normalisasi dilakukan dengan mengubah nilai piksel dari rentang 0–255 menjadi 0–1 untuk mempercepat konvergensi saat pelatihan CNN.

```

29 count = 0
30 while True:
31     ret, frame = cam.read()
32     if not ret:
33         break
34
35     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
36     faces = face_cascade.detectMultiScale(gray, 1.3, 5)
37
38     for (x, y, w, h) in faces:
39         count += 1
40         face_img = gray[y:y+h, x:x+w]
41         face_img = cv2.resize(face_img, (100, 100))
42         cv2.imwrite(f"{path}/{count}.jpg", face_img)
43         cv2.rectangle(frame, (x,y), (x+w, y+h), (255,0,0), 2)
44
45     cv2.imshow("Daftarkan Wajah", frame)
46     if cv2.waitKey(1) == 27 or count >= 100:
47         break
48
49 cam.release()
50 cv2.destroyAllWindows()
51
52 print(f"✅ {count} gambar wajah berhasil disimpan di: {path}")

```

Gambar 3. 5 Code Pengambilan Citra Wajah

Count untuk menghitung jumlah citra wajah yang tersimpan, kemudian citra wajah diubah menjadi *grayscale* agar proses deteksi lebih cepat. Jika wajah ditemukan maka citra akan dipotong dari frame aslinya, diubah ukurannya menjadi 100x100 piksel dan disimpan dalam format file JPG.

2. Integrasi dan Pengelolaan Data

```

9   data = []
10  labels = []
11
12  dataset_path = "dataset"

```

Gambar 3. 6 Code untuk Menyimpan Label

Data yang digunakan dalam sistem terdiri atas dataset wajah yang dikumpulkan dari mahasiswa atau karyawan yang menjadi peserta absensi digital. Setiap data wajah diberi label sesuai dengan identitas pemiliknya. Proses integrasi mencakup pengumpulan data dari berbagai sumber, kemudian dilakukan penyimpanan terstruktur menggunakan database atau folder direktori berdasarkan nama pengguna. Tahapan ini penting untuk menjamin bahwa data dapat dengan mudah diakses dan digunakan baik pada proses pelatihan maupun pengujian model CNN.

3. Transformasi Data

```

14  for person in os.listdir(dataset_path):
15      person_path = os.path.join(dataset_path, person)
16      if os.path.isdir(person_path):
17          for img_name in os.listdir(person_path):
18              img_path = os.path.join(person_path, img_name)
19              img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
20              img = cv2.resize(img, (100, 100))
21              data.append(img.flatten()) # 100x100 → 1D
22              labels.append(person)
23
24  X = np.array(data)
25  y = np.array(labels)
26
27  le = LabelEncoder()
28  y_encoded = le.fit_transform(y)
29
30  X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.2, random_state=42)
31
32  model = SVC(kernel='linear', probability=True)
33  model.fit(X_train, y_train)
34
35  # Buat folder model jika belum ada
36  if not os.path.exists("model"):
37      os.makedirs("model")
38
39  # Simpan model dan label encoder
40  dump(model, 'model/face_svm_model.joblib')
41  dump(le, 'model/label_encoder.joblib')
42
43  print("Model SVM berhasil dilatih dan disimpan.")

```

Gambar 3. 7 Code untuk Transformasi Data

Transformasi data bertujuan untuk menyesuaikan format citra agar dapat diolah oleh CNN. Setelah proses *preprocessing*, seluruh data citra dikonversi menjadi *array* numerik (dalam bentuk matriks) menggunakan *library* seperti NumPy. Selain itu, dilakukan encoding pada label (nama pengguna) menjadi bentuk numerik menggunakan metode *one-hot encoding* atau *label encoding*. Citra kemudian dibagi menjadi dua bagian utama, yaitu data latih (*training data*) dan data uji (*testing data*). Data yang telah ditransformasi inilah yang kemudian digunakan sebagai input dalam pelatihan model CNN.

4. Klasifikasi Menggunakan Convolutional Neural Network

Tahapan terakhir adalah klasifikasi wajah menggunakan arsitektur CNN. CNN merupakan jenis jaringan saraf tiruan yang dirancang untuk mengenali pola dari data citra. Dalam penelitian ini, CNN digunakan untuk mengekstraksi fitur wajah secara otomatis dan mengklasifikasikannya ke dalam identitas tertentu. Arsitektur CNN terdiri dari beberapa lapisan, antara lain *convolutional layer* untuk mengekstraksi fitur lokal, *pooling layer* untuk mereduksi dimensi data, *flatten layer* untuk mengubah data menjadi vektor satu dimensi, dan *dense layer* sebagai lapisan klasifikasi. Model dilatih menggunakan data latih hingga mencapai akurasi yang optimal. Setelah pelatihan selesai, model diuji menggunakan data uji untuk mengevaluasi performanya dalam mengenali wajah. Hasil klasifikasi digunakan untuk mencocokkan identitas pengguna dalam sistem absensi digital.

```

8  model = load("model/face_svm_model.joblib")
9  label_encoder = load("model/label_encoder.joblib")
10 face_cascade = cv2.CascadeClassifier(cv2.data.harcascades + "haarcascade_frontalface_default.xml")
11
12 def mark_attendance(name):
13     now = datetime.now()
14     date_str = now.strftime("%Y-%m-%d")
15     time_str = now.strftime("%H:%M:%S")
16     file = "attendance.csv"
17     already_marked = set()
18
19     if os.path.exists(file):
20         with open(file, "r") as f:
21             reader = csv.reader(f)
22             for row in reader:
23                 if row[0] == name and row[1] == date_str:
24                     already_marked.add(name)
25
26     if name not in already_marked:
27         with open(file, "a", newline="") as f:
28             writer = csv.writer(f)
29             writer.writerow([name, date_str, time_str])
30             print(f"{name} dicatat hadir.")

```

Gambar 3. 8 Code untuk mencatat kehadiran pada wajah yang terdeteksi

Setelah melewati tahap transformasi data, wajah yang sudah diambil sebelumnya telah diberi label dan dapat dikenali oleh sistem. Jika wajah yang dideteksi cocok dengan salah satu citra wajah yang sudah tersimpan di dataset maka wajah akan dikenali dan dicatat waktu kehadirannya.

```

34 while True:
35     ret, frame = cap.read()
36     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
37     faces = face_cascade.detectMultiScale(gray, 1.3, 5)
38
39     for (x,y,w,h) in faces:
40         face = gray[y:y+h, x:x+w]
41         face = cv2.resize(face, (100, 100)).flatten().reshape(1, -1)
42
43         pred = model.predict(face)[0]
44         prob = model.predict_proba(face)[0].max()
45
46         name = label_encoder.inverse_transform([pred])[0]
47
48         if prob > 0.8:
49             mark_attendance(name)
50             cv2.putText(frame, name, (x,y-10), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,255,0), 2)
51         else:
52             cv2.putText(frame, "Unknown", (x,y-10), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,255), 2)
53
54         cv2.rectangle(frame, (x,y), (x+w, y+h), (255,0,0), 2)
55
56     cv2.imshow("Face Recognition Absensi (SVM)", frame)
57     if cv2.waitKey(1) == 27:
58         break
59
60 cap.release()
61 cv2.destroyAllWindows()

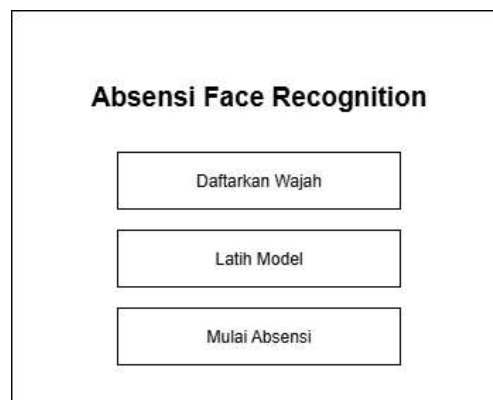
```

Gambar 3. 9 Code untuk mengenali wajah

Yang dimaksud dengan mengenali wajah di gambar 3.9 adalah saat wajah tertangkap oleh kamera maka akan dideteksi apakah wajah sudah dikenali atau belum. Jika wajah sudah dikenal maka akan terdeteksi oleh kotak beserta nama/label yang sudah disimpan, sedangkan wajah yang tidak dikenali akan terdeteksi kotak yang bertuliskan *Unknown*.

3.4 Desain Interface

Berikut tampilan desain interface aplikasi absensi yang dirancang pada penelitian ini yang dapat terlihat pada perangkat pengguna.



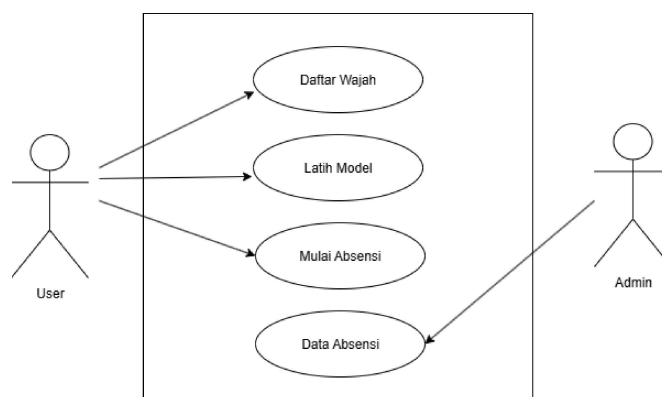
Gambar 3. 10 Interface Aplikasi Sistem Absensi

Peneliti merancang interface sederhana untuk aplikasi sistem absensi berbasis face recognition in. Dapat dilihat ada 3 tombol yang setiap tombolnya memiliki fungsi masing masing, tombol “daftarkan wajah” berfungsi untuk mendeteksi wajah kemudia diambil citra wajahnya sesuai yang ditentukan. Tombol “latih model” berfungsi untuk melatih langsung model citra yang diambil secara real-time untuk disimpan kedalam database agar dapat dideteksi. Tombol “mulai absensi” berfungsi untuk mencatat waktu kehadiran pada wajah yang terdeteksi.

Setelah merancang antarmuka aplikasi, langkah selanjutnya adalah melakukan pemodelan sistem secara visual menggunakan *Unified Modeling Language* (UML). UML digunakan untuk menggambarkan struktur dan alur kerja sistem agar lebih mudah dipahami dan diimplementasikan. Dalam penelitian ini, tiga jenis diagram UML digunakan, yaitu *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*. Ketiganya bertujuan untuk menjelaskan fungsionalitas sistem absensi berbasis face recognition mulai dari interaksi pengguna, alur proses absensi, hingga urutan komunikasi antar komponen dalam sistem.

1. Use Case Diagram

Use case diagram berfungsi untuk menggambarkan interaksi antara pengguna dan sistem absensi yang dirancang. Diagram ini membantu memvisualisasikan fungsionalitas utama sistem berdasarkan peran dari aktor yang terlibat, serta menunjukkan bagaimana sistem merespons tindakan yang dilakukan oleh pengguna, serta menjadi dasar dalam memahami ruang lingkup sistem dan kebutuhan fungsional dari sistem absensi berbasis *face recognition* yang dikembangkan.

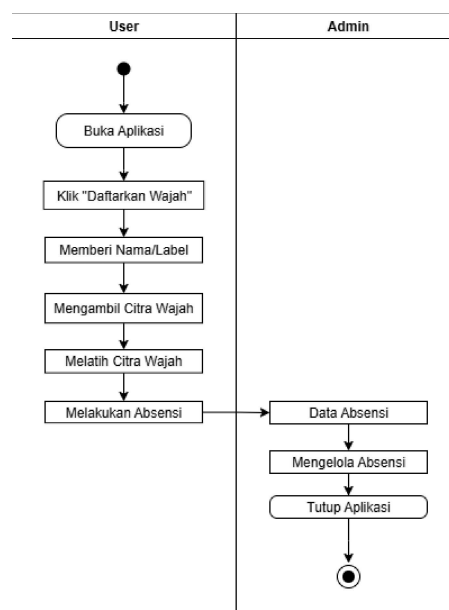


Gambar 3. 11 Use Case Diagram

Pada Gambar 3.11, ditunjukkan bahwa *User* merupakan aktor utama yang berinteraksi dengan sistem melalui tiga dari empat fungsi utama, yaitu Daftar Wajah, Latih Model, dan Mulai Absensi. *User* bertugas mendaftarkan wajah pengguna ke dalam sistem, melatih model CNN agar dapat mengenali wajah yang telah didaftarkan, dan memulai proses absensi yang dilakukan secara otomatis oleh sistem. Selanjutnya, data absensi akan tercatat secara otomatis dan dapat diakses oleh *admin* untuk keperluan dokumentasi atau rekap kehadiran.

2. Activity Diagram

Activity diagram digunakan untuk memodelkan alur aktivitas sistem secara terstruktur dan menggambarkan proses bisnis yang terjadi di dalam sistem absensi berbasis face recognition. Diagram ini membantu memahami bagaimana suatu aktivitas dimulai, dijalankan, dan berakhir, serta menunjukkan urutan langkah-langkah yang dilakukan oleh aktor dalam sistem.

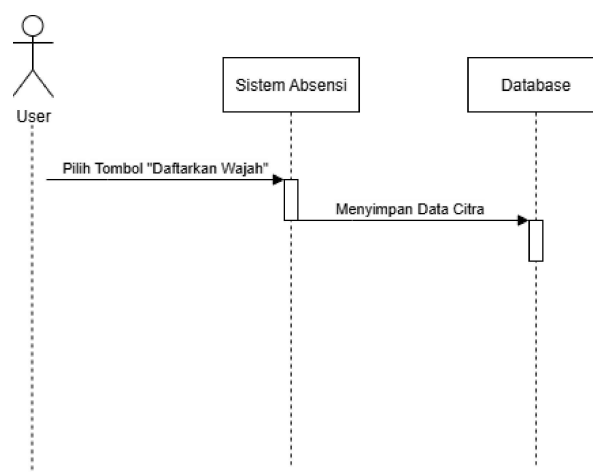


Gambar 3. 12 Activity Diagram

Pada Gambar 3.12, digambarkan aktivitas utama dalam sistem absensi berbasis *face recognition* yang dilakukan oleh dua aktor, yaitu *User* dan *Admin*. Proses dimulai dari *User* yang membuka aplikasi dan mendaftarkan wajah melalui pemberian label, pengambilan citra, dan pelatihan data wajah menggunakan metode CNN. Setelah itu, *User* dapat langsung melakukan absensi dengan sistem yang akan mengenali wajah secara otomatis. Sementara itu, *Admin* berperan dalam mengelola dan memantau data absensi yang telah terekam, hingga akhirnya menutup aplikasi. Diagram ini memperjelas alur operasional sistem dari awal hingga akhir secara efisien dan terotomatisasi.

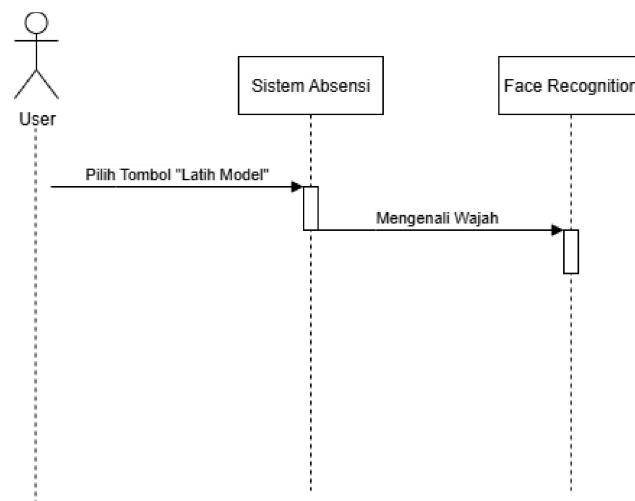
3. Sequence Diagram

Sequence diagram digunakan untuk menggambarkan urutan interaksi antar objek atau komponen dalam sistem sesuai dengan alur waktu. Diagram ini memodelkan bagaimana proses dalam sistem absensi berbasis *face recognition* dijalankan secara berurutan, mulai dari proses pendaftaran wajah, pelatihan model, hingga pencatatan absensi.



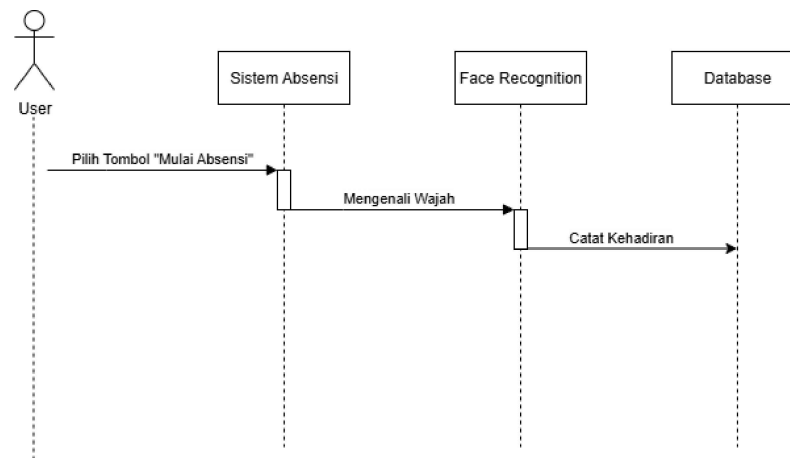
Gambar 3. 13 Sequence Diagram Daftar Wajah

Pada Gambar 3.13, proses pendaftaran wajah dimulai saat *admin* menekan tombol “Daftarkan Wajah” melalui antarmuka sistem. Sistem kemudian akan mengambil data citra wajah pengguna menggunakan metode *Haar Cascade* dan meneruskan informasi tersebut ke database untuk disimpan. Data ini nantinya akan digunakan sebagai referensi dalam proses pelatihan model dan pengenalan wajah saat absensi.



Gambar 3. 14 Sequence Diagram Latih Model

Berikutnya pada gambar 3.14, proses pelatihan dimulai ketika admin memilih tombol “Latih Model”. Sistem absensi akan mengakses data citra wajah dari database, lalu mengirimkan data tersebut ke modul face recognition untuk dilakukan proses pelatihan menggunakan algoritma Convolutional Neural Network (CNN). Modul ini akan menghasilkan model pengenalan wajah yang siap digunakan dalam proses absensi.



Gambar 3. 15 Sequence Diagram Mulai Absensi

Dijelaskan pada gambar 3.15, proses absensi dimulai ketika admin memilih tombol “Mulai Absensi”. Sistem akan mengaktifkan kamera untuk mendeteksi wajah pengguna, lalu mengirimkan citra wajah tersebut ke modul face recognition. Jika wajah dikenali, hasil identifikasi akan diteruskan ke database untuk dicatat sebagai data kehadiran beserta waktu absensi. Proses ini berjalan secara otomatis dan real-time.

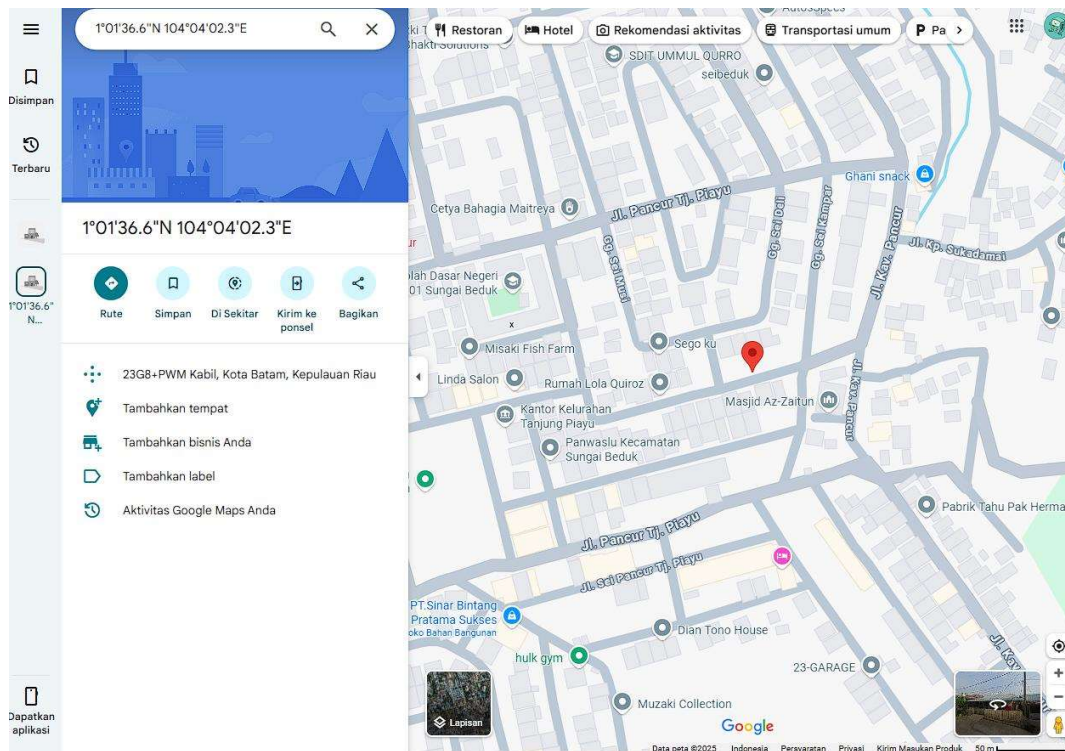
3.5 Lokasi dan Jadwal Penelitian

Penelitian ini dilaksanakan berdasarkan tahapan yang telah direncanakan secara sistematis, jadwal penelitian disusun untuk memastikan setiap tahap pengembangan, mulai dari perancangan hingga pengujian sistem, dapat berjalan secara terstruktur dan sesuai dengan tujuan yang ditetapkan. kegiatan yang peneliti lakukan dalam jangka waktu pembuatan skripsi disajikan pada tabel 3.1 berikut ini.

Tabel 3. 1 Tabel Jadwal Penelitian

No	Kegiatan	Jadwal Penelitian					
		2025					
		Maret	April	Mei	Juni	Juli	Agust
1.	Identifikasi dan Perumusan Masalah						
2.	Studi Literatur						
3	Pengumpulan Data Wajah						
4.	Implementasi Sistem Absensi						
5.	Evaluasi Sistem Absensi dan Analisis Hasil Pengujian						
6	Laporan Hasil Akhir						

Penelitian ini dilakukan di lokasi yang mendukung kebutuhan teknis dan operasional sistem absensi berbasis face recognition. Rincian lokasi secara lengkap ditampilkan pada Gambar 3.11.



Gambar 3. 16 Lokasi Penelitian