

PENGEMBANGAN APLIKASI FACE IDENTIFICATION DENGAN LIBRARY DLIB DAN OPENCV UNTUK MENGIKUTAKAN AKTIVITAS MAHASISWA DI KAMPUS**Pramana Afriandy¹, Yus Sholva², Khairul Hafidh³**

Universitas Tanjungpura Pontianak

e-mail: pramanaafriandy@gmail.com

Abstract (English)

Student academic engagement within the campus environment plays a crucial role in their success in higher education. One method to assess student activity in academic settings is through an attendance system. However, conventional attendance systems merely record whether a student is present or absent, without providing information on the actual duration of their attendance. To address this limitation, this study developed a face identification system using the Dlib and OpenCV libraries. The application operates in real time by detecting and recognizing students whose faces are captured by classroom cameras, identifying them, processing the data into measurable indicators of student activity, and presenting the information to end-users (students, parents, and lecturers). The system was developed using the prototyping model. Results of black-box testing indicate that the application successfully performed all required functionalities. In terms of face recognition accuracy, the system achieved a rate of 81.77%. Furthermore, the application is capable of determining the duration of student presence across multiple rooms simultaneously, with a duration calculation accuracy of 82.96%.

Article History

Submitted: 16 November 2025

Accepted: 19 November 2025

Published: 20 November 2025

Key WordsActivity; Dlib; Face
Identification; Campus;
Student; OpenCV**Abstrak (Indonesia)**

Keterlibatan atau keaktifan mahasiswa secara akademis di lingkungan kampus berperan penting untuk keberhasilan mereka di perguruan tinggi. Salah satu cara untuk mengetahui tingkat aktivitas mahasiswa dalam kegiatan akademik di kampus adalah melalui sistem presensi. Namun, sistem presensi hanya mencatat hadir atau tidaknya seseorang, sementara jumlah besar waktu durasi berapa lama kehadiran tidak dapat diketahui. Pada penelitian ini dibuat sebuah sistem dengan aplikasi *face identification* menggunakan library Dlib dan OpenCV, yang bekerja secara *real-time* mengenali setiap mahasiswa yang wajahnya tertangkap oleh kamera ruangan, mengidentifikasi, mengolahnya menjadi nilai aktivitas mahasiswa, dan menyajikannya sebagai informasi kepada pengguna (mahasiswa, orang tua dan dosen). Sistem dikembangkan dengan model pengembangan *prototyping*. Dari pengujian *black-box* yang dilakukan aplikasi ini telah mampu menjalankan semua fungsionalitasnya. Pada pengujian akurasi identifikasi wajah didapatkan akurasi sebesar 81,77%. Aplikasi ini dapat mengetahui durasi keberadaan mahasiswa yang berada di beberapa ruangan secara bersamaan, melalui identifikasi wajah dengan akurasi perhitungan durasi sebesar 82,96%.

Sejarah Artikel

Submitted: 16 November 2025

Accepted: 19 November 2025

Published: 20 November 2025

Kata KunciAktivitas; Dlib; Face
Identification; Kampus;
Mahasiswa; OpenCV**PENDAHULUAN**

Keterlibatan atau keaktifan mahasiswa, baik secara akademis maupun sosial, dengan dosen, staf, dan rekan mahasiswa memiliki pengaruh (*influence*) besar terhadap keberhasilan mereka di perguruan tinggi (Tinto, 2012). Keterlibatan ini tidak hanya memberikan dukungan sosial-emosional, tetapi juga mendorong mahasiswa untuk lebih aktif dalam kegiatan belajar. Salah satu indikator yang dapat digunakan untuk menilai tingkat aktivitas mahasiswa di lingkungan akademik adalah kehadiran mereka di kampus yang diperoleh melalui sistem absensi.

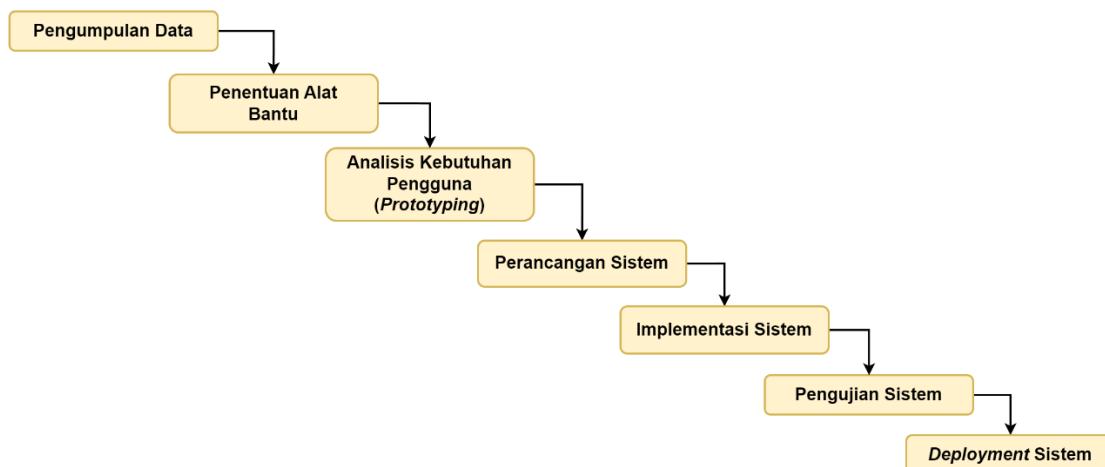
Sistem absensi manual di kampus masih memiliki kelemahan, seperti potensi kecurangan, pencatatan yang kurang akurat, serta keterbatasan dalam memantau durasi kehadiran mahasiswa. Kondisi ini menyebabkan aktivitas mahasiswa, baik saat mengikuti kegiatan terjadwal maupun di luar kegiatan, tidak dapat dipantau secara menyeluruh. Selain itu penggunaan sistem yang masih manual dapat mengakibatkan tingginya biaya operasional dan mahasiswa harus melakukan tugas berulang yang mana memakan banyak waktu (Adam et al., 2023).

Berdasarkan permasalahan tersebut, penelitian ini mengusulkan pengembangan aplikasi terintegrasi yang terdiri dari dua komponen utama, yaitu aplikasi desktop dan aplikasi web. Aplikasi *desktop* berfungsi sebagai *Face Station* di setiap ruangan kampus untuk memantau keberadaan mahasiswa melalui identifikasi wajah menggunakan Dlib dan OpenCV, sedangkan aplikasi web menyajikan informasi aktivitas mahasiswa kepada mahasiswa, orang tua dan dosen. Data yang dikumpulkan meliputi identitas mahasiswa, waktu masuk, waktu keluar, dan lokasi ruangan, serta dapat diklasifikasikan sebagai kehadiran pada kegiatan terjadwal atau keberadaan di kampus di luar jadwal kegiatan. Sistem ini diharapkan mampu menyajikan informasi aktivitas mahasiswa secara akurat dan *real-time*, sekaligus meningkatkan transparansi dan akuntabilitas di lingkungan akademik.

Penelitian ini menggunakan metode prototipe, yang sesuai untuk pengembangan perangkat lunak bersifat kustom karena dirancang berdasarkan kebutuhan spesifik. Metode ini juga tepat digunakan dalam tugas akhir atau skripsi yang berfokus pada implementasi metode atau algoritma tertentu pada kasus nyata (Susanto & Andriana, 2016).

METODE PENELITIAN

Beberapa tahapan metodologi yang dilaksanakan pada penelitian ini terdiri dari tujuh (7) langkah utama, mulai dari pengumpulan data hingga *deployment* sistem. Gambar 1 menunjukkan alur dari tiap langkah yang dilalui pada penelitian ini.



Gambar 1 . Tahapan Penelitian

1. Pengumpulan Data

Pengumpulan data adalah metode yang digunakan untuk mengumpulkan bahan nyata yang digunakan dalam penelitian (Mochamad Nashrullah et al., 2023). Langkah pertama yang dilakukan dalam pengumpulan data adalah observasi gedung kampus Jurusan Informatika Universitas Tanjungpura, guna mengetahui bagaimana sistem kehadiran atau pemantauan aktivitas mahasiswa yang dilakukan saat ini, bagaimana arus masuk-keluar mahasiswa di

ruangan kampus, serta ketersediaan perangkat pendukung (kamera, komputer, internet) yang tersedia di ruangan kampus.

Kemudian dilakukan studi pustaka terhadap teknologi yang akan digunakan yakni Dlib, OpenCV, Flask dan MongoDB untuk mengetahui bagaimana teknologi tersebut bekerja, bagaimana cara menggunakannya, serta apa spesifikasi minimum perangkat keras yang dapat menjalankannya.

Dataset wajah untuk pengembangan aplikasi diperoleh dari dua (2) sumber utama, yaitu: (1) partisipasi mahasiswa melalui kuesioner yang memungkinkan mereka mengunggah foto wajah secara mandiri, dan (2) pencarian foto wajah publik dari internet, khususnya tokoh-tokoh terkenal untuk memperkaya variasi data uji. *Dataset* wajah yang digunakan dapat dilihat pada Tabel 1.

Tabel 1 Dataset Wajah yang Digunakan

Nama	Foto Wajah				Nama	Foto Wajah			
Doni Winata					Mhs_C (Leonardo DiCaprio)				
Irman Kevin					Mhs_D (Margot Robbie)				
Pramana Afriandy					Mhs_E (Jessica Chastain)				
Raihan Akhmadin					Mhs_F (Quentin Tarantino)				
Mhs_A (Anne Hathaway)					Mhs_G (Christopher Nolan)				
Mhs_B (Brad Pitt)					Mhs_H (Matthew McConaughey)				

2. Penentuan Alat Bantu

Setelah mengetahui spesifikasi minimum (*minimum requirements*) dari teknologi yang akan digunakan, peneliti menentukan alat bantu penelitian yang sesuai kebutuhan dan kemampuan. Alat bantu pada penelitian ini terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*), beberapa diantaranya sebagai berikut.

- Laptop Asus Vivobook, Processor: Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz, RAM: 8,00 GB, GPU: NVIDIA GeForce MX250 2GB, System Type: 64-bit operating system. Kamera: USB2.0 HD UVC WebCam.
- Smartphone Xiaomi Poco M4 Pro, Chipset: Mediatek MT6781 Helio G96, GPU: Mali-G57 MC2, RAM: 6,00 GB. Kamera utama: 64 MP 1080p@30fps.
- Aplikasi Draw.io (versi 24.7.8) berfungsi untuk membuat diagram perancangan aplikasi.
- Aplikasi Visual Studio Code (versi 1.83.1) berfungsi menyunting kode program aplikasi.
- Python 3 (versi 3.11). Bahasa pemrograman utama untuk pengembangan aplikasi desktop dan web.
- Aplikasi MongoDB Compass (versi 1.40.4) berfungsi untuk melakukan interaksi dengan basis data MongoDB melalui GUI (*Graphical User Interface*).
- Aplikasi Google Chrome (versi 129) berfungsi untuk mengakses laman dan debug aplikasi web yang dibangun melalui penyimpanan server lokal (*localhost*). Aplikasi

iVCam (versi 7 untuk Windows dan Android) berfungsi untuk mengakses kamera dari smartphone melalui kabel USB.

3. Analisis Kebutuhan Pengguna

Setelah data diakusisi dan alat bantu teridentifikasi, lalu dilakukan diskusi langsung dengan klien untuk merumuskan gambaran umum sistem. Klien dalam penelitian ini adalah dosen Jurusan Informatika Universitas Tanjungpura, yaitu Dr. Ir. Yus Sholva, S.T., M.T. Hasil diskusi tersebut menetapkan bahwa sistem yang akan dikembangkan terdiri dari dua aplikasi yang saling terintegrasi, kedua di antaranya sebagai berikut.

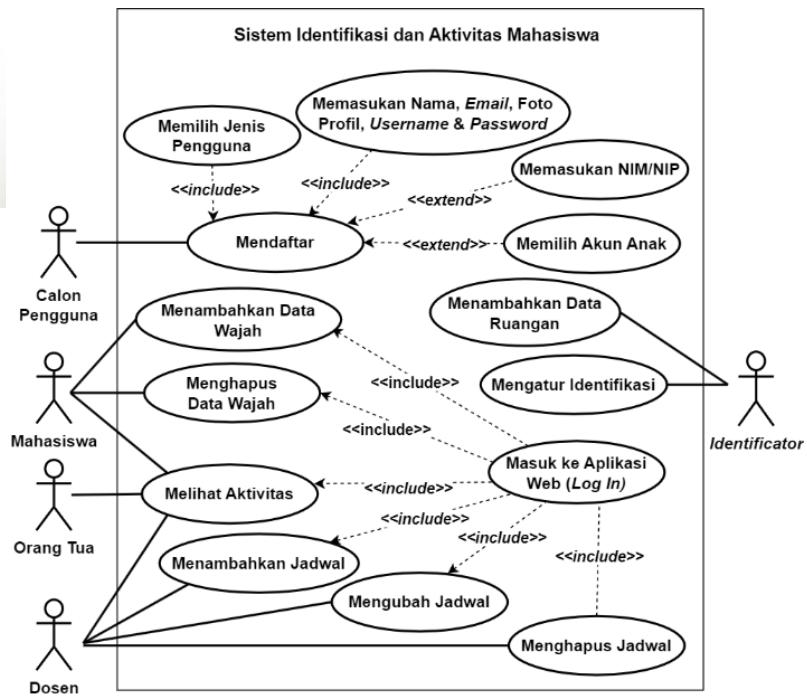
- a) Aplikasi *desktop* (untuk sistem operasi Windows) yang berfungsi untuk menjalankan proses identifikasi wajah melalui beberapa kamera yang terhubung melalui kabel USB.
- b) Aplikasi web yang berfungsi untuk menyajikan data hasil identifikasi wajah dalam bentuk informasi aktivitas kepada Mahasiswa, Orang Tua dan Dosen.

Fitur-fitur utama yang akan dikembangkan terdiri dari lima (5) fitur, yaitu: Identifikasi Wajah, Pendaftaran Pengguna, Login, Akses Informasi Aktivitas, dan Pengaturan Data Wajah. Jenis pengguna yang akan terlibat dalam sistem juga terdiri dari lima (5) kategori pengguna, yakni sebagai berikut.

- a) Calon Pengguna, yakni pengguna yang ingin mendaftar agar dapat menggunakan aplikasi sebagai Mahasiswa, Orang tua atau Dosen.
- b) Mahasiswa, yakni pengguna yang dapat melihat/mengakses informasi aktivitasnya sendiri. Mahasiswa juga dapat menambahkan/menghapus data wajahnya sendiri.
- c) Orang tua, yakni pengguna yang dapat melihat/mengakses informasi aktivitas anaknya.
- d) Dosen, yakni pengguna yang dapat melihat/mengakses informasi aktivitas semua Mahasiswa.
- e) *Identifier*, yakni pengguna yang menjalankan proses identifikasi wajah pada sebuah komputer (*face station*), melalui salinan aplikasi *desktop*.

Pada tahap Analisis Kebutuhan Pengguna, dilakukan proses prototyping. Model prototipe atau prototyping adalah metode pengembangan perangkat lunak yang membuat versi awal sistem untuk memahami kebutuhan pengguna, terutama saat kebutuhan belum jelas atau terdapat ketidakpastian terkait algoritma, sistem operasi, dan interaksi manusia dengan mesin (Pressman, 2002).

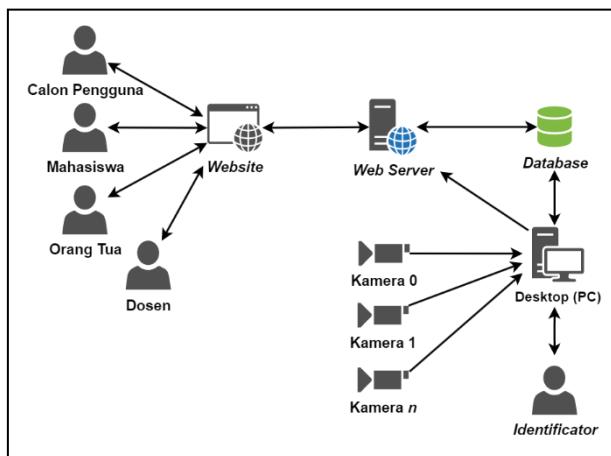
Pada penelitian ini, prototyping ini dilakukan dalam tiga (3) kali iterasi. Pada setiap iterasi, disusun sebuah use case diagram sebagai dasar dalam pengembangan prototipe. Rancangan use case diagram dari prototipe final yang dibuat (prototipe versi ketiga) dapat dilihat pada Gambar 3.

Gambar 2. Rancangan *Use Case Diagram*

4. Perancangan Sistem

Setelah prototipe dinilai memadai oleh klien, tahap selanjutnya adalah mengembangkan prototipe menjadi sistem utuh yang siap digunakan. Perancangan sistem dilakukan untuk menggambarkan secara rinci komponen-komponen sistem yang akan dibangun. Rancangan sistem yang disusun meliputi Arsitektur Sistem, Skema Basis Data dan *Flowchart* Identifikasi Wajah.

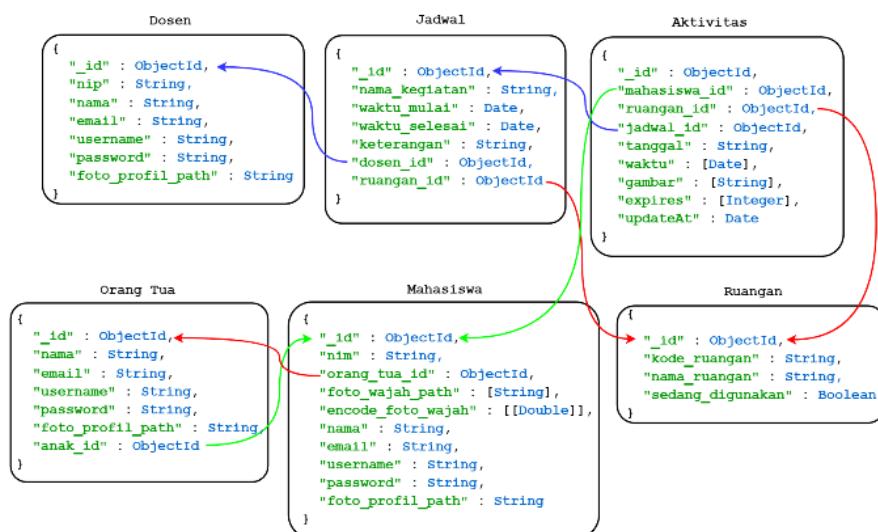
Arsitektur sistem (*system architecture*) adalah cara dimana sebuah sistem yang terdiri dari *network*, *hardware*, dan *software* di strukturkan (Saepudin et al., 2022). Arsitektur sistem yang ditunjukkan pada Gambar 3.5 dirancang berdasarkan prototipe versi final, sistem melibatkan lima aktor yaitu Calon Pengguna, Mahasiswa, Orang Tua, Dosen, dan Identifier.



Gambar 3. Rancangan Arsitektur Sistem

Sistem terdiri dari dua (2) aplikasi yang terhubung ke satu basis data yaitu Aplikasi Web untuk informasi aktivitas mahasiswa, dan Aplikasi *Desktop* untuk identifikasi wajah dan pencatatan aktivitas.

Basis data dapat dikatakan sebagai kumpulan data yang terintegrasi dan diatur sehingga datatersebut dapat dikumpulkan, dimanipulasi, diambil, dan dicari secara tepat (Mukhlis & Santoso, 2023). Gambar 3.6 menunjukkan skema basis data beserta relasi antar koleksi data. Sistem ini menggunakan enam koleksi data: Dosen, Jadwal, Aktivitas, Orang Tua, Mahasiswa, dan Ruangan. Setiap koleksi memiliki field sesuai kebutuhan dari prototipe final, kecuali `_id` yang otomatis dibuat MongoDB sebagai identitas unik bertipe ObjectId.

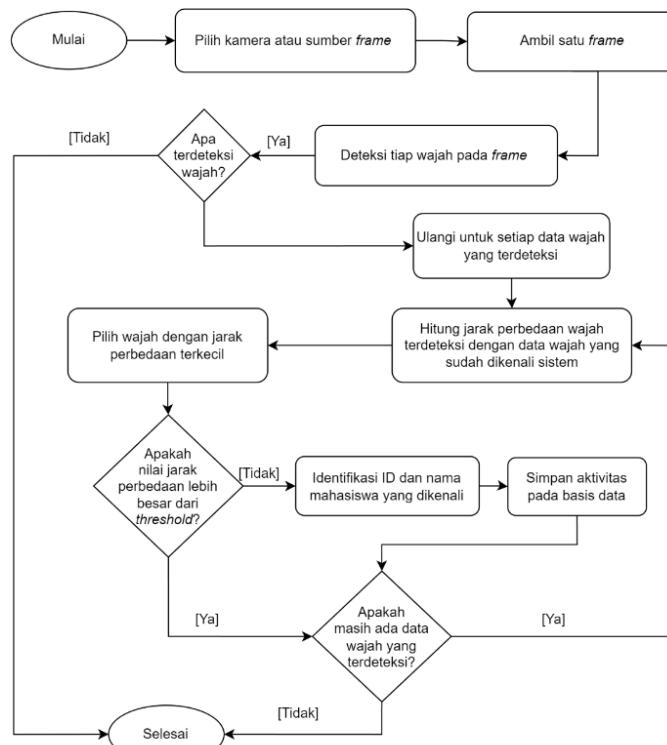


Gambar 4. Rancangan Skema Basis Data

Beberapa field berfungsi sebagai referensi antar koleksi yakni `dosen_id` (Jadwal) ke koleksi data Dosen, `mahasiswa_id`, `ruangan_id`, dan `jadwal_id` (Aktivitas) masing-masing ke koleksi data Mahasiswa, Ruangan, dan Jadwal, `orang_tua_id` (Mahasiswa) ke Orang Tua, serta `anak_id` (Orang Tua) ke Mahasiswa.

Flowchart (diagram alir) adalah diagram yang digunakan untuk menggambarkan sebuah algoritma yang terstruktur dan mudah dipahami oleh orang lain (Ni Nyoman Emang Smrti et al., 2023). *Flowchart* proses identifikasi wajah pada Gambar 5 menunjukkan algoritma yang dirancang untuk menentukan apakah wajah dalam sebuah *frame* teridentifikasi atau tidak.

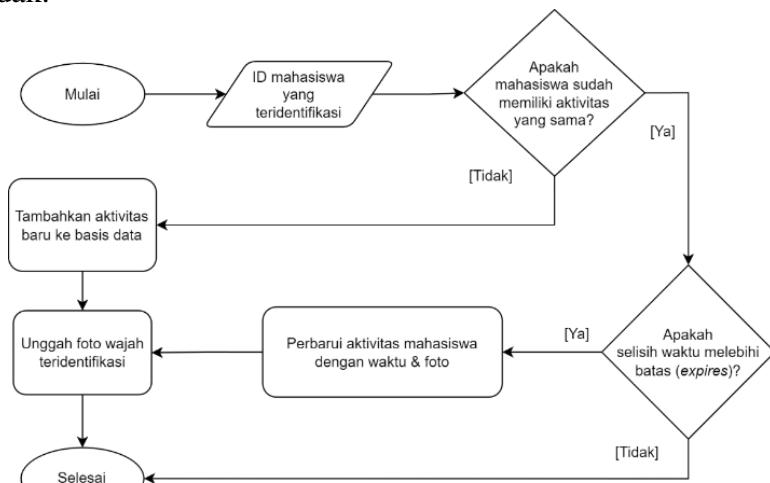
Jurnal Ilmiah Sain dan Teknologi



Gambar 5. Rancangan Flowchart Proses Identifikasi

Proses identifikasi wajah mencakup pemilihan frame dari kamera, deteksi wajah, encoding, perhitungan jarak perbedaan, hingga pengubahan hasil identifikasi menjadi data aktivitas yang disimpan ke basis data. Jika tidak ada wajah terdeteksi, proses dihentikan. Jika ada, setiap wajah dibandingkan dengan data wajah yang dikenali sistem. Hasil perbandingan menghasilkan jarak; jika jarak terendah melebihi ambang toleransi, wajah dianggap tidak dikenali dan proses dilanjutkan ke wajah berikutnya. Jika tidak, wajah dikenali, dan ID mahasiswa digunakan untuk menyimpan data aktivitas.

Flowchart Simpan Aktivitas pada Gambar 6 menunjukkan alur kerja penyimpanan data aktivitas mahasiswa, untuk menentukan apakah data dari wajah yang teridentifikasi dapat disimpan atau tidak.



Gambar 6. Rancangan Flowchart Proses Simpan Aktivitas

Proses simpan aktivitas menerima masukan berupa ID mahasiswa yang teridentifikasi. Sistem kemudian mencari data aktivitas berdasarkan ID mahasiswa, ID ruangan, tanggal, dan

ID jadwal (jika ada kegiatan). Jika data ditemukan, sistem akan memperbarui field waktu dan gambar, tetapi hanya jika selisih waktu terakhir melebihi nilai *expires*. Jika tidak ditemukan, sistem langsung menyisipkan data aktivitas baru tanpa pengecekan waktu.

5. Implementasi Sistem

Rancangan sistem diimplementasikan melalui pengkodean *back-end* yang terintegrasi dengan *front-end* (prototipe versi final). *Back-end* dikembangkan untuk mengolah data dan berkomunikasi dengan basis data serta komponen lainnya. Sistem dibangun menggunakan Python, aplikasi web dengan Flask dan Dlib untuk ekstraksi serta validasi wajah, sedangkan aplikasi *desktop* menggunakan OpenCV dan Dlib untuk identifikasi wajah. Keduanya menggunakan Pymongo untuk mengakses MongoDB Atlas.

Pengenalan wajah merupakan bagian dari visi komputer, yang digunakan untuk mengidentifikasi seseorang dalam metode biometrik berdasarkan citra pada wajah mereka (Teoh et al., 2021). Proses pengenalan wajah dibagi menjadi tiga (3) tahap yaitu *face detection*, *face encoding* dan *face comparing*. Setiap tahap menggunakan library Dlib, dan implementasinya ditunjukkan melalui potongan sumber kode program berikut. Sumber kode diambil dari API pengenalan wajah bernama *face_recognition* yang dikembangkan oleh Adam Geitgey (ageitgey@gmail.com).

```
def face_locate(img, number_of_times_to_upsample=1, model="hog"):
    """
    Returns an array of bounding boxes of human faces in a image
    :param img: An image (as a numpy array)
    :param number_of_times_to_upsample: How many times to upsample the image
    looking for faces. Higher numbers find smaller faces.
    :param model: Which face detection model to use. "hog" is less accurate but
    faster on CPUs. "cnn" is a more accurate
    deep-learning model which is GPU/CUDA accelerated (if
    available). The default is "hog".
    :return: A list of tuples of found face locations in css (top, right,
    bottom, left) order
    """
    return [_trim_css_to_bounds(_rect_to_css(face), img.shape) for face in
    _raw_face_locations(img, number_of_times_to_upsample)]
```

Gambar 7. Fungsi *face_locate*

Deteksi wajah (*face detection*) dimulai dengan menyiapkan objek gambar dari kamera menggunakan fungsi *VideoCapture* OpenCV. Gambar kemudian diproses dengan fungsi *face_locate* (lihat Gambar 7) untuk menentukan lokasi wajah. Fungsi *face_locate* menerima array NumPy dari gambar berformat RGB dan mengembalikan koordinat kotak wajah (a, b, c, d). Jika tidak ada wajah, array kosong. Secara default, fungsi ini menggunakan model HOG, namun juga mendukung CNN. Penelitian ini menggunakan model *default*, yaitu HOG.

```
def face_encode(face_image, known_face_locations=None, num_jitters=1,
model="small"):
    """
    Given an image, return the 128-dimension face encoding for each face in the
    image.

    :param face_image: The image that contains one or more faces
    :param known_face_locations: Optional - the bounding boxes of each face if
    you already know them.
    :param num_jitters: How many times to re-sample the face when calculating
    encoding. Higher is more accurate but slower (i.e. 100 is 100x slower)
    :param model: Optional - which model to use. "large" (default) or "small"
    which only returns 5 points but is faster
    :return: A list of 128-dimensional face encodings (one for each face in the
    image)

    raw_landmarks = _raw_face_landmarks(face_image, known_face_locations, model)
    return [np.array(face_encoder.compute_face_descriptor(face_image,
    raw_landmark_set, num_jitters)) for raw_landmark_set in raw_landmarks]
```

Gambar 8. Fungsi *face_encode*

Proses *face encoding* menggunakan fungsi *face_encode* (lihat Gambar 8) dengan parameter *face_image* dan *known_face_locations* dari hasil fungsi *face_locate*. Jika parameter *known_face_locations* tidak diisi, fungsi akan memanggil *face_locations* secara otomatis. Dlib menyediakan dua model deteksi *landmark* yaitu *shape_predictor_5_point* dan *shape_predictor_68_point*. Penelitian ini menggunakan *shape_predictor_68_point*, serta model ResNet-34 (*dlib_face_recognition_resnet_model_v1.dat*) untuk ekstraksi *embedding*. Fungsi *face_encode* menghasilkan array NumPy berdimensi 128 dari hasil ekstraksi tiap wajah.

```
def compare_faces(known_face_encodings, face_encoding_to_check, tolerance):
    distances = face_distance(known_face_encodings, face_encoding_to_check)

    if True in tuple(distances <= tolerance):
        return np.argmin(distances)
    else:
        return False
```

Gambar 9. Fungsi compare_faces

Fungsi compare_faces (lihat Gambar 3.25) membandingkan data hasil fungsi face_encoding dari gambar baru dengan data known_face_encodings dari basis data. Parameter tolerance menentukan batas maksimal jarak agar dua wajah dianggap sama semakin mirip wajah, semakin kecil jaraknya.

```
def face_distance(face_encodings, face_to_compare):
    """
    Given a list of face encodings, compare them to a known face encoding and
    get a euclidean distance
    for each comparison face. The distance tells you how similar the faces are.

    :param face_encodings: List of face encodings to compare
    :param face_to_compare: A face encoding to compare against
    :return: A numpy ndarray with the distance for each face in the same order
    """
    if len(face_encodings) == 0:
        return np.empty((0))

    return np.linalg.norm(face_encodings - face_to_compare, axis=1)
```

Gambar 10. Fungsi face_distance

Fungsi compare_faces memanggil face_distance (Gambar 3.26) untuk menghitung jarak euclidean antara *encoded* wajah terdeteksi dan data di basis data. Jika ada jarak di bawah toleransi, fungsi mengembalikan indeks dengan jarak terkecil; jika tidak, mengembalikan nilai *false*.

Penelitian ini menggunakan OpenCV untuk pengolahan citra. Dengan beberapa fungsi yang digunakan yaitu VideoCapture, cvtColor, resize, rectangle, dan putText (lihat Gambar 3.27).

```
if not ret:
    capture = cv2.VideoCapture("static/images/Fix-Windows-10-Camera-not-working.jpg")
    ret, frame = capture.read()

# cv2 uses 'BGR' but 'GUI' needs 'RGB'
frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
frame = cv2.resize(frame, (555, 380))

x, y, w, h = 0, 0, 130, 50
cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 0), -1)
cv2.putText(frame, f"KAMERA {kamera_value}", (x + int(w/10), y + int(h/1.6)), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 255, 255), 2)
```

Gambar 11. Penggunaan Fungsi dari OpenCV

Fungsi VideoCapture OpenCV digunakan untuk mengakses video dari kamera, file, atau stream daring. Dalam penelitian ini, fungsi tersebut digunakan untuk membaca video langsung dari kamera sebagai masukan proses identifikasi wajah. Fungsi cvtColor OpenCV mengubah ruang warna gambar. Dalam penelitian ini, digunakan untuk mengonversi format BGR dari VideoCapture menjadi RGB. Fungsi resize OpenCV digunakan untuk menyesuaikan ukuran gambar dari kamera agar sesuai dengan tampilan jendela aplikasi. Fungsi putText OpenCV digunakan untuk menambahkan teks pada video sebagai keterangan asal kamera.

6. Pengujian Sistem

a. Pengujian *Black-Box*

Black-Box Testing adalah metode pengujian yang berfokus pada fungsionalitas sistem aplikasi dengan memberikan input acak untuk memastikan sistem dapat menolak data yang salah dan menerima data yang benar (Nur Ichsanudin et al., 2022). Pengujian ini pada aplikasi web bertujuan untuk memastikan setiap fitur berfungsi sesuai peran pengguna, melalui 94 skenario yang dikelompokkan berdasarkan jenis pengguna, yaitu Pengunjung, Mahasiswa, Orang Tua, dan Dosen. Pengujian *Black-Box* terhadap aplikasi *desktop*

bertujuan untuk memastikan fungsionalitas sesuai peran pengguna (*identifier*), dimana terdapat 15 skenario untuk pengujian ini.

b. Pengujian Akurasi Pengenalan Wajah

◆ Pengujian ini bertujuan mengukur akurasi dari aplikasi *desktop* dalam mengenali wajah dengan benar, menggunakan rumus (lihat Gambar 12) yang melibatkan nilai TP (*True Positive*), TN (*True Negative*), FP (*False Positive*) dan FN (*False Negatives*).

$$Akurasi (\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

Gambar 12. Rumus Akurasi Pengenalan Wajah

Pengujian dilakukan menggunakan kamera belakang Xiaomi Poco M4 Pro dengan pencahayaan ruangan normal (± 500 lux), seperti ditunjukkan pada Gambar 12. Jarak antara kamera dan wajah berkisar 3-4 meter.

Pengujian dilakukan terhadap empat (4) peserta yang telah terdaftar dalam basis data aplikasi melalui data wajah yang tersimpan sebelumnya. Adapun peserta tersebut adalah Doni Winata, Irmans Kevin, Pramana Afriandy, dan Raihan Akhmadin.

c. Pengujian Kecepatan Pengenalan Wajah

Pengujian kecepatan pengenalan wajah bertujuan mengukur waktu yang dibutuhkan sistem untuk mengidentifikasi wajah. Pengujian ini penting untuk menilai kemampuan sistem bekerja secara *real-time*, terutama dalam situasi dinamis seperti keluar-masuknya mahasiswa di ruang kelas. Waktu proses diukur dari saat frame diterima hingga proses ekstraksi *face embedding* dan pencocokan selesai.

Tabel 2 Rencana Pengujian Kecepatan Pengenalan Wajah

No.	Resolusi	Kondisi Daya	Kecepatan	Rata-rata
1	480p (854x480)	<i>Plugged in</i>		
2	720p (1280x720)	<i>Plugged in</i>		
3	1080p (1920x1080)	<i>Plugged in</i>		
4	480p (854x480)	<i>Unplugged</i>		
5	720p (1280x720)	<i>Unplugged</i>		
6	1080p (1920x1080)	<i>Unplugged</i>		

Pengujian dilakukan dalam dua variasi utama yaitu resolusi frame dan jumlah wajah per frame (lihat Tabel II). Resolusi yang digunakan mencakup 640×480, 1280×720, dan 1920×1080 piksel untuk mengetahui pengaruh ukuran citra terhadap waktu pemrosesan. Jumlah wajah juga divariasikan yakni dari 1 hingga 30 untuk menguji performa sistem dalam mengenali banyak individu secara bersamaan. Data yang dicatat mencakup resolusi frame, kondisi daya perangkat (terhubung atau tidak), dan rata-rata waktu identifikasi.

d. Pengujian Ketangguhan Pengenalan Wajah

Pengujian ketangguhan pengenalan wajah bertujuan untuk mengevaluasi akurasi dan keandalan sistem dalam kondisi yang tidak ideal. Kondisi yang diuji antara lain sebagai berikut.

- Jarak Wajah dari Kamera.
- Pose atau Orientasi Wajah.
- Penggunaan Masker dan Kacamata.

Pengujian dilakukan di ruang Sidang Utama Jurusan Informatika dengan pencahayaan normal (± 500 lux), menggunakan kamera Xiaomi Poco M4 Pro beresolusi 852×480 piksel. Partisipan dalam pengujian ini adalah Pramana Afriandy dan Doni Winata.

e. Pengujian Akurasi Pencatatan Aktivitas

Pengujian ini bertujuan untuk mengukur tingkat kesesuaian data aktivitas mahasiswa yang dicatat oleh aplikasi identifikasi (*desktop*) dengan kondisi sebenarnya di lapangan. Perhitungan akurasi mengacu pada rumus akurasi (lihat Gambar 13), dimana “V1” merupakan durasi sebenarnya, dan “V2” adalah durasi yang dicatat oleh aplikasi.

$$Akurasi (\%) = (100 - Perbedaan) \times 100$$

$$Perbedaan (\%) = \frac{|V_1 - V_2|}{\frac{(V_1 + V_2)}{2}}$$

Gambar 13. Rumus Akurasi Pencatatan Aktivitas

Sumber masukan (input) pengujian adalah Video Pengujian 1 yang menampilkan Mahasiswa Test G, A, H, dan E, serta Video Pengujian 2 yang menampilkan Mahasiswa Test D, C, F, dan B. Video Pengujian 1 dan 2 dapat diakses di <https://drive.google.com/drive/folders/10EPI6sqFWv86BQO9PquMH8PQUEjAPaKt?usp=sharing>, dengan durasi kehadiran tiap orang diatur berbeda. Tabel VII memuat perbandingan durasi nyata mahasiswa berada di kampus dengan durasi yang tercatat oleh sistem. Kolom “Realita” berisi durasi yang telah ditetapkan (sebenarnya terjadi), “Hasil” berisi durasi hasil identifikasi wajah, dan “Perbedaan (%)” berisi selisih sesuai perhitungan pada Rumus Akurasi Pencatatan Aktivitas.

Tabel 3 Rencana Pengujian Akurasi Pencatatan Aktivitas

Identitas Partisipan	Durasi Berada di Kampus			Durasi Menghadiri Kegiatan		
	Realita (J:M:S)	Hasil (J:M:S)	Pebedaan (%)	Realita (J:M:S)	Hasil (J:M:S)	Pebedaan (%)
Mahasiswa Test G	0:30:00			1:55:00		
Mahasiswa Test A	0:10:00			1:50:00		
Mahasiswa Test H	0:04:00			1:00:00		

Mahasiswa Test E	0:25:00			0:50:00		
Mahasiswa Test D	0:26:00			1:15:00		
Mahasiswa Test C	0:20:00			1:25:00		
Mahasiswa Test F	1:00:00			1:30:00		
Mahasiswa Test B	0:05:00			1:40:00		
	Rata-rata			Rata-rata		

7. Deployment Sistem

Deployment untuk aplikasi web dilakukan melalui hosting pada sebuah VPS dengan spesifikasi 4 GB RAM, 2 vCPU, 60 GB penyimpanan, dan sistem operasi Ubuntu 22.04. Adapun *deployment* aplikasi *desktop* dilakukan dengan mengompilasi kode Python menjadi *file executable* (.exe) menggunakan *compiler* Nuitka.

HASIL DAN PEMBAHASAN

1. Tampilan Antarmuka Aplikasi Web

a. Halaman Pendaftaran

Halaman Pendaftaran Mahasiswa (Gambar 13) menyediakan formulir bagi calon pengguna untuk mendaftar sebagai mahasiswa dengan isian nama, NIM, email, username, password, dan konfirmasi password.

Gambar 14. Halaman Pendaftaran Mahasiswa

Halaman Pendaftaran Orang Tua (Gambar 14) menyediakan formulir bagi calon pengguna untuk mendaftar sebagai orang tua dengan isian nama, akun anak, email, *username*, *password*, dan konfirmasi *password*.

Gambar 14. Halaman Pendaftaran Orang Tua

Halaman Pendaftaran Dosen (Gambar 15) menyediakan formulir bagi calon pengguna untuk mendaftar sebagai dosen dengan isian nama, NIK, email, *username*, *password*, dan konfirmasi *password*.

Gambar 15. Halaman Pendaftaran Dosen

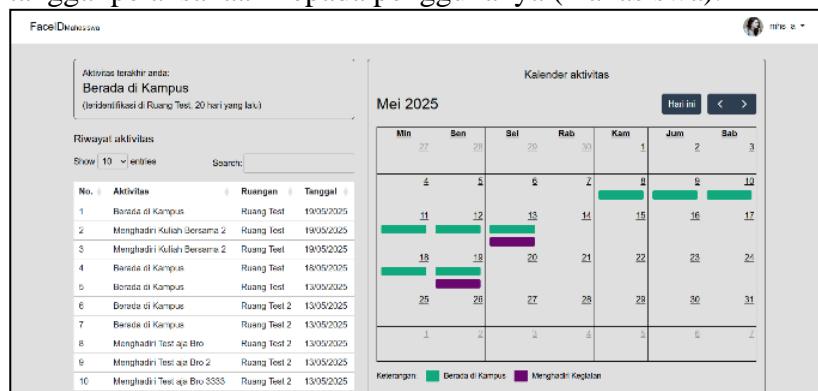
b. Halaman *Login*

Halaman *Login* (Gambar 16) menyediakan formulir bagi pengguna untuk masuk ke aplikasi dengan isian nama, akun anak, *username*, *password*, dan konfirmasi *password*.

Gambar 16. Halaman *Login*

c. Halaman Mahasiswa

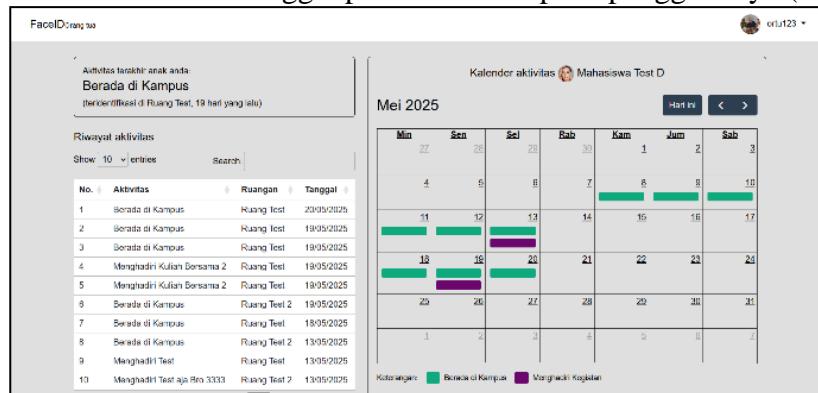
Halaman Beranda Mahasiswa (Gambar 17) menampilkan informasi aktivitas dalam tiga format: aktivitas terakhir, tabel riwayat seluruh aktivitas, dan kalender aktivitas berdasarkan tanggal pelaksanaan kepada penggunanya (mahasiswa).



Gambar 17. Halaman Beranda Mahasiswa

d. Halaman Orang Tua

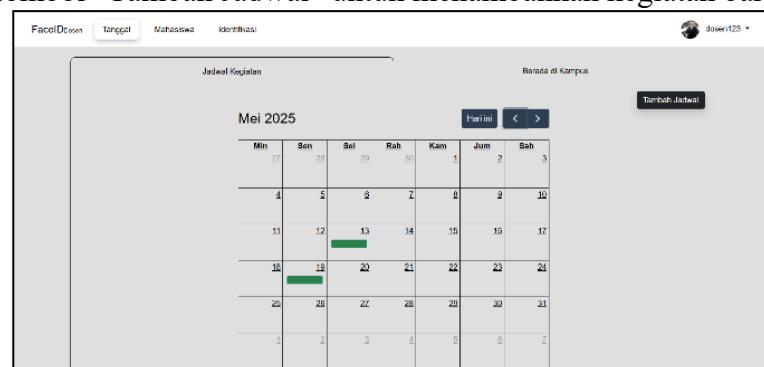
Halaman Beranda Orang Tua (Gambar 18) menampilkan informasi aktivitas mahasiswa (anak pemilik akun) dalam tiga format: aktivitas terakhir, tabel riwayat, dan kalender aktivitas berdasarkan tanggal pelaksanaan kepada penggunanya (orang tua).



Gambar 18. Halaman Beranda Orang Tua

e. Halaman Dosen

Halaman Beranda Dosen (Gambar 19) menampilkan jadwal kegiatan dalam bentuk kalender. Tanggal dengan kegiatan ditandai kotak hijau yang menaut ke daftar kegiatan, serta tersedia tombol “Tambah Jadwal” untuk menambahkan kegiatan baru.



Gambar 19. Halaman Beranda Dosen

f. Halaman Aktivitas

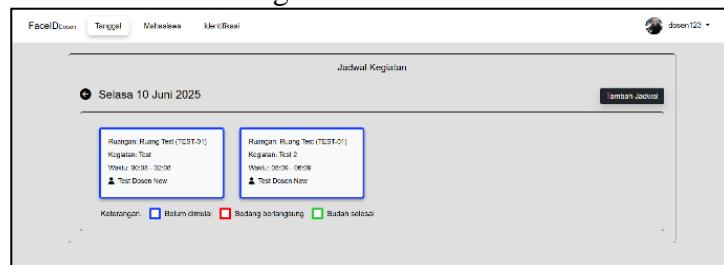
Formulir Jadwal Kegiatan Baru (Gambar 20) pada Halaman Beranda Dosen digunakan untuk menambahkan jadwal kegiatan dengan isian ruangan, nama kegiatan, waktu mulai, waktu selesai, dan keterangan. Waktu mulai dan selesai berformat datetime mencakup tanggal, bulan, tahun, jam, dan menit.

The figure shows a screenshot of the "Tambah Jadwal Kegiatan" (Add Activity Schedule) form. It includes fields for Room (Ruang), Activity Name (Nama Kegiatan), Start Time (Waktu Mulai), End Time (Waktu Selesai), and Description (Keterangan). A "Simpan" (Save) button is visible.

Gambar 20. Formulir Tambah Jadwal Kegiatan

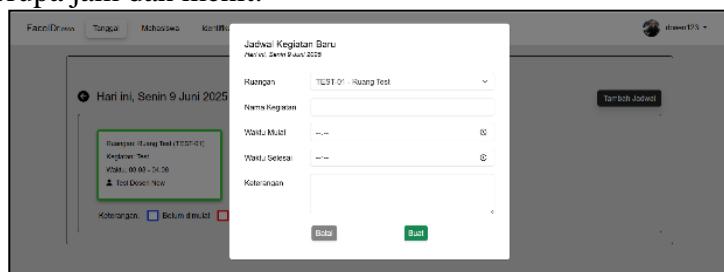
Halaman Jadwal Kegiatan Harian (Gambar 21) menampilkan seluruh kegiatan pada hari atau tanggal yang dipilih di Beranda Dosen. Setiap jadwal ditandai border biru (belum

mulai), merah (berlangsung), atau hijau (selesai). Tersedia tombol “Tambah Jadwal Tanggal Ini” untuk menambahkan kegiatan baru.



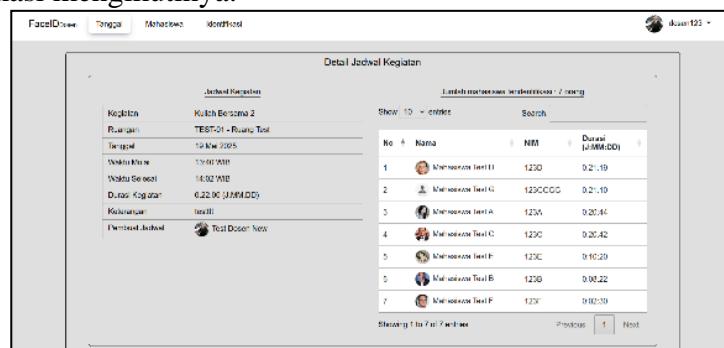
Gambar 21. Halaman Jadwal Kegiatan Harian (per tanggal)

Formulir Jadwal Kegiatan Baru Harian (Gambar 22) pada halaman Jadwal Kegiatan Harian digunakan dosen untuk menambahkan kegiatan pada tanggal terpilih. Formulir ini berisi isian ruangan, nama kegiatan, waktu mulai, waktu selesai, dan keterangan, dengan format waktu berupa jam dan menit.



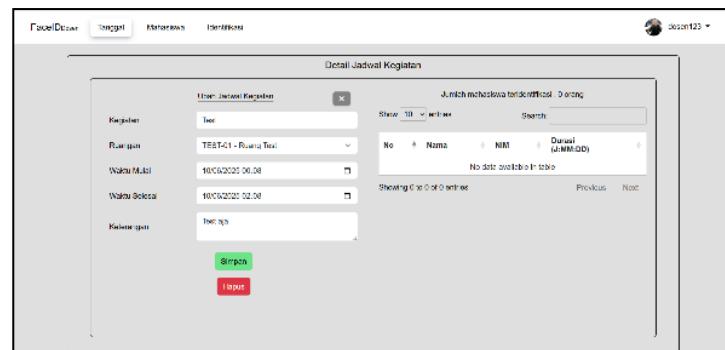
Gambar 22. Formulir Tambah Jadwal Kegiatan Harian (per tanggal)

Halaman Detail Jadwal Kegiatan (Gambar 23) terbuka saat salah satu jadwal pada Jadwal Kegiatan Harian dipilih, menampilkan *detail* kegiatan beserta daftar mahasiswa yang teridentifikasi mengikutinya.



Gambar 23. Halaman *Detail* Jadwal Kegiatan

Formulir Ubah Jadwal Kegiatan (Gambar 24) pada Halaman Detail Jadwal Kegiatan dapat diakses dosen jika jadwal tersebut buatannya sendiri dan belum selesai. Data yang dapat diubah meliputi nama kegiatan, ruangan, waktu mulai, waktu selesai, dan keterangan.



Gambar 24. Halaman Ubah Jadwal Kegiatan

Dosen dapat melihat data aktivitas mahasiswa yang teridentifikasi berada di kampus (tanpa mengikuti kegiatan) melalui Halaman Berada di Kampus (Gambar 4.13). Halaman ini menampilkan daftar nama, NIM, lokasi, dan durasi aktivitas pada hari akses, serta memungkinkan penggantian tanggal untuk melihat data pada hari tertentu.

Detail Jadwal Kegiatan					
Daftar Mahasiswa					
Senin 19 Mei 2025					10:00:00
Search					Cancel
No.	Nama	NIM	Tempat Teridentifikasi (Ruang)	Waktu Terakhir (JJ/MM/DD)	Durasi Aktifitas (J:MM:SS)
1	Pramanika Afandy	1234567890123	TEST41 - Ruang Tes1	2025-05-19 10:00:00	00:02:00
2	Mahasiswa Tes1 C	1234567890124	TEST41 - Ruang Tes1	2025-05-19 10:00:01	00:00:00
3	Mahasiswa Tes1 D	1234567890125	TEST41 - Ruang Tes1	2025-05-19 10:00:31	00:00:57
4	Mahasiswa Tes1 G	1234567890126	TEST41 - Ruang Tes1	2025-05-19 10:00:33	00:00:00
5	Mahasiswa Tes1 A	1234567890127	TEST41 - Ruang Tes1	2025-05-19 10:00:34	00:00:44
6	Mahasiswa Tes1 C	1234567890128	TEST40 - Ruang Tes1	2025-05-19 10:00:37	00:00:02
7	Mahasiswa Tes1 D	1234567890129	TEST40 - Ruang Tes1	2025-05-19 10:00:37	00:00:02
8	Mahasiswa Tes1 E	1234567890130	TEST40 - Ruang Tes1	2025-05-19 10:00:37	00:00:02

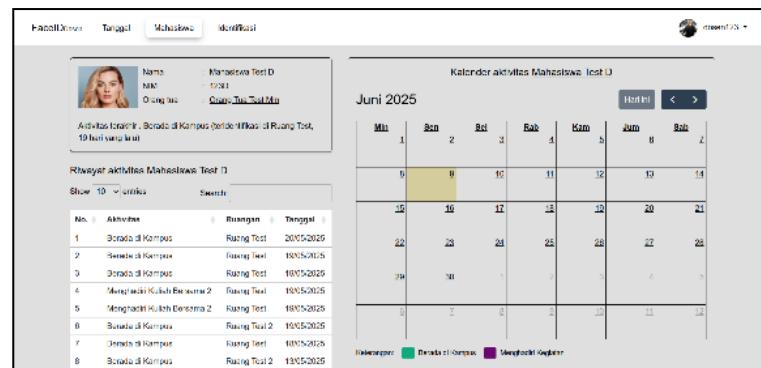
Gambar 25. Halaman Aktivitas Berada di Kampus

Selain berdasarkan tanggal, dosen dapat mengakses aktivitas per mahasiswa melalui Halaman Aktivitas Terakhir Mahasiswa (Gambar 4.14), yang menampilkan tabel berisi nama, NIM, aktivitas terakhir, lokasi, tanggal, dan waktu teridentifikasi.

Aktivitas Terakhir Mahasiswa					
Search					
No.	Nama	NIM	Aktivitas Terakhir	Tempat Teridentifikasi (Ruang)	Tanggal (DD/MM/YYYY) Waktu (HH:MM:SS)
1	Pramanika Afandy	1234567890123	Berada di Kampus	TEST41 - Ruang Tes1	2025/05/20 10:00:00
2	Rulan	1234567890124	Berada di Kampus	TEST41 - Ruang Tes1	2025/05/20 10:00:00
3	Vigo	1234567890125	Berada di Kampus	TEST41 - Ruang Tes1	2025/05/20 10:00:00
4	Uvan Wicaks	1234567890126	Berada di Kampus	TEST41 - Ruang Tes1	2025/05/20 10:00:00
5	Mahasiswa Tes1 D	1234567890127	Berada di Kampus	TEST41 - Ruang Tes1	2025/05/20 10:00:00
6	Mahasiswa Tes1 B	1234567890128	Berada di Kampus	TEST41 - Ruang Tes1	2025/05/20 10:00:00
7	Mahasiswa Tes1 C	1234567890129	Berada di Kampus	TEST41 - Ruang Tes1	2025/05/20 10:00:00
8	Fandi	1234567890130	Berada di Kampus	TEST41 - Ruang Tes1	2025/05/20 10:00:00
9	Mahasiswa Tes1 E	1234567890131	Berada di Kampus	TEST41 - Ruang Tes1	2025/05/20 10:00:00

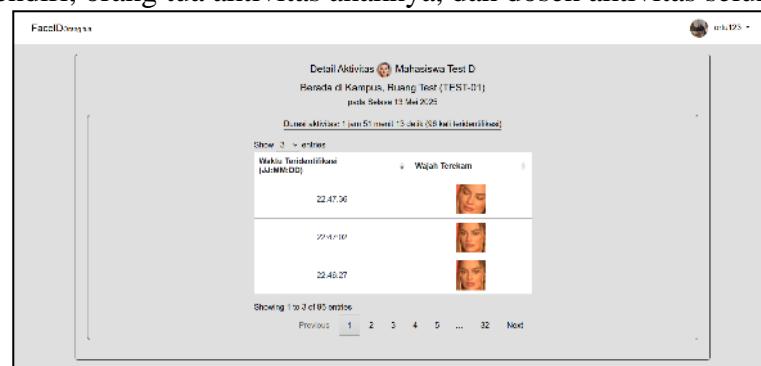
Gambar 26. Halaman Aktivitas Terakhir Mahasiswa

Pada Halaman Aktivitas Terakhir Mahasiswa, memilih salah satu nama akan membuka Halaman *Detail Mahasiswa* (Dosen) (Gambar 27) menampilkan kepada dosen seluruh aktivitas yang dimiliki seorang mahasiswa.

Gambar 27. Halaman *Detail* Mahasiswa (untuk Dosen)

g. Halaman Detail Aktivitas

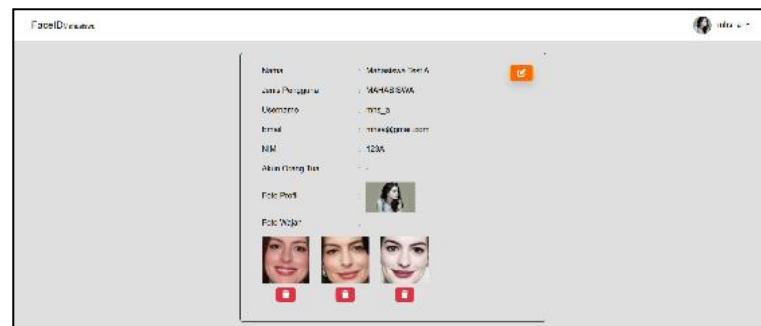
Pada Halaman Detail Aktivitas (Gambar 28), mahasiswa hanya dapat melihat aktivitasnya sendiri, orang tua aktivitas anaknya, dan dosen aktivitas seluruh mahasiswa.



Gambar 28. Halaman Detail Aktivitas

h. Halaman Akun

Halaman Akun Mahasiswa (Gambar 29) hanya dapat diakses pemilik akun untuk melihat, mengubah data (nama, email, NIM, foto profil), serta menambah atau menghapus foto wajah.



Gambar 29. Halaman Akun Mahasiswa

Halaman Akun Orang Tua (Gambar 30) hanya dapat diakses pemilik akun untuk melihat dan mengubah data (nama, email, akun anak, dan foto profil).



Halaman Akun Dosen (Gambar 31) hanya dapat diakses pemilik akun untuk melihat dan mengubah data (nama, email, NIP, dan foto profil).

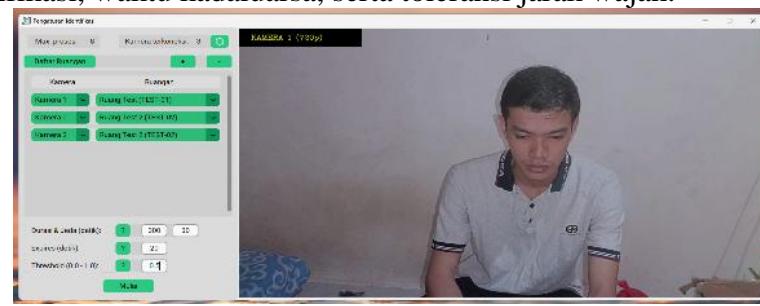


Gambar 31. Halaman Akun Dosen

2. Tampilan Antarmuka Aplikasi *Desktop*

a. Halaman Pengaturan Identifikasi

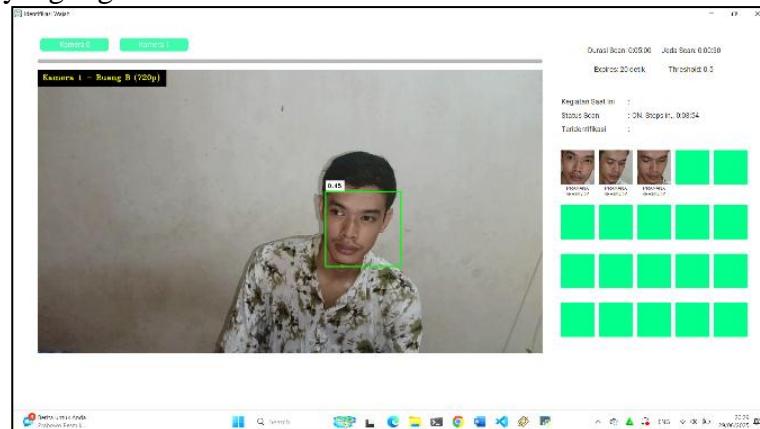
Halaman Pengaturan Identifikasi: Halaman ini (Gambar 32) digunakan untuk memulai identifikasi dengan mengatur jumlah proses, pasangan ruangan & kamera, durasi dan jeda identifikasi, waktu kadaluarsa, serta toleransi jarak wajah.



Gambar 32. Halaman Pengaturan Identifikasi

b. Halaman Identifikasi

Halaman ini (Gambar 4.28) muncul setelah Pengaturan Identifikasi, menampilkan proses identifikasi yang berjalan, serta foto dan nama mahasiswa yang terdeteksi pada setiap kamera yang digunakan.



Gambar 33. Halaman Monitor Identifikasi

3. Hasil Pengujian Aplikasi

a. Hasil Pengujian *Black-Box*

Pada pengujian *Black-Box*, aplikasi berhasil menjalankan seluruh fungsinya. Skenario pengujian mulai dari BB001 hingga BB110 telah dilaksanakan dan menghasilkan validitas sebesar 100%.

b. Hasil Pengujian Akurasi Pengenalan Wajah

Pengujian ini (Gambar 4.29) dilakukan pada 10 Februari 2025 selama 20 menit dengan *threshold* pengenalan wajah sebesar 0,55.



Gambar 34. Kondisi Ruangan Pengujian Akurasi Pengenalan Wajah

Setiap wajah yang terdeteksi disimpan dengan label dikenali/tidak, identitas terdekat, dan jarak perbandingan. Label ditentukan dari toleransi jarak. Selama 20 menit pengujian, sistem melakukan 949 identifikasi (Tabel 4.2).

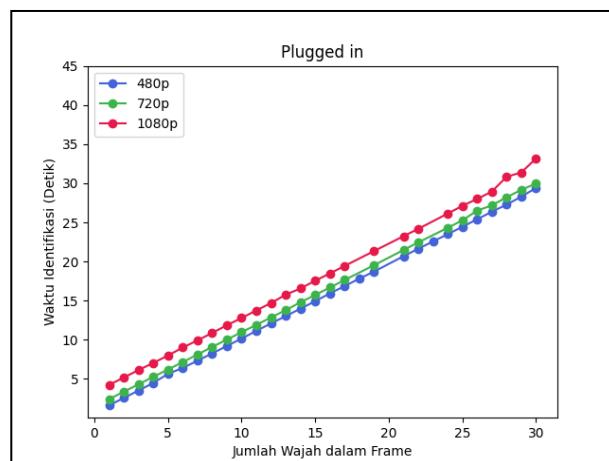
Tabel 4 Hasil Pengujian Akurasi Pengenalan Wajah

Jumlah Identifikasi	True Positives	True Negatives	False Positives	False Negatives
949	769	7	95	78

Nilai TP, TN, FP, dan FN dari pengujian dihitung dengan rumus akurasi (Persamaan 3.1), menghasilkan akurasi 81,77%.

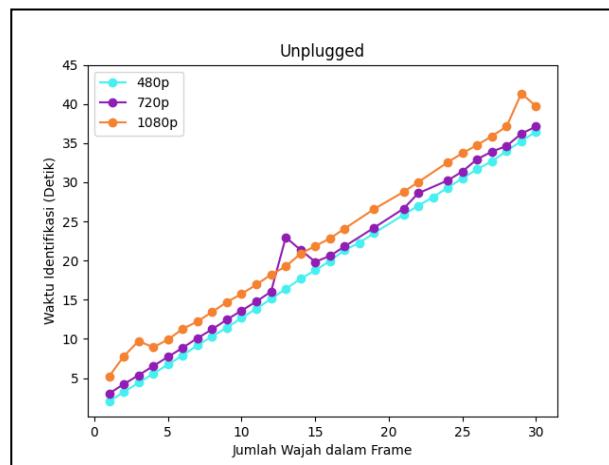
c. Hasil Pengujian Kecepatan Pengenalan Wajah

Data hasil pengujian tersedia di tautan Google Drive (<https://drive.google.com/drive/folders/10EP16sqFWv86BQO9PquMH8PQUEjAPaKt?usp=sharing>) dan divisualisasikan dalam plot diagram. Gambar 4.30 menunjukkan hubungan jumlah wajah per frame dengan waktu identifikasi pada kondisi *plugged-in* (laptop sedang terhubung ke *adaptor* atau *charger*) untuk resolusi 480p (biru), 720p (hijau), dan 1080p (merah).



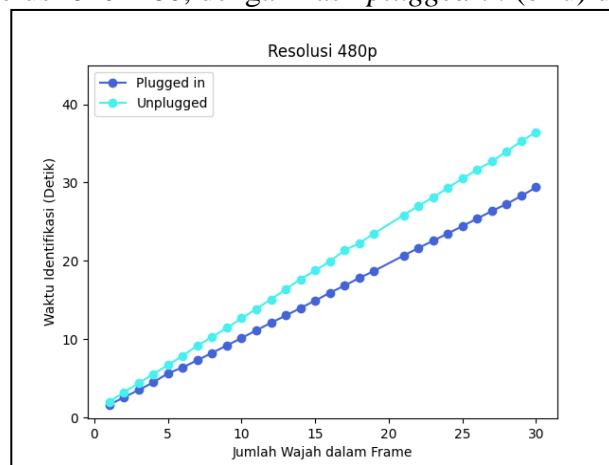
Gambar 35. Hasil Pengujian Kecepatan dalam Kondisi *Plugged-in*

Gambar 4.31 menunjukkan hubungan jumlah wajah per frame dengan waktu identifikasi pada kondisi *unplugged* untuk resolusi 480p (sian), 720p (ungu), dan 1080p (orange).



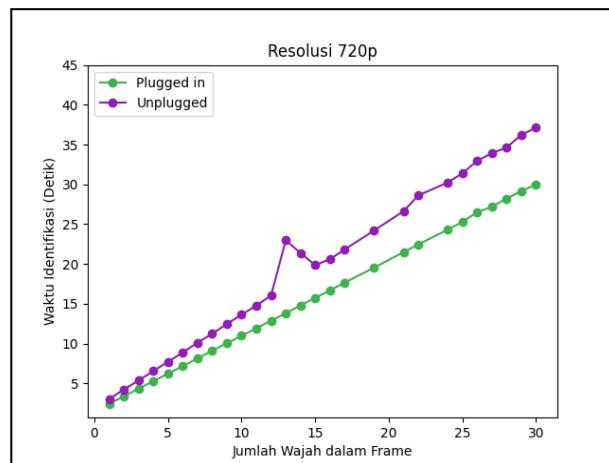
Gambar 36. Hasil Pengujian Kecepatan dalam Kondisi *Unplugged*

Gambar 4.32 menunjukkan hubungan jumlah wajah per frame dengan waktu identifikasi pada resolusi 640×480 , dengan hasil *plugged-in* (biru) dan *unplugged* (sian).



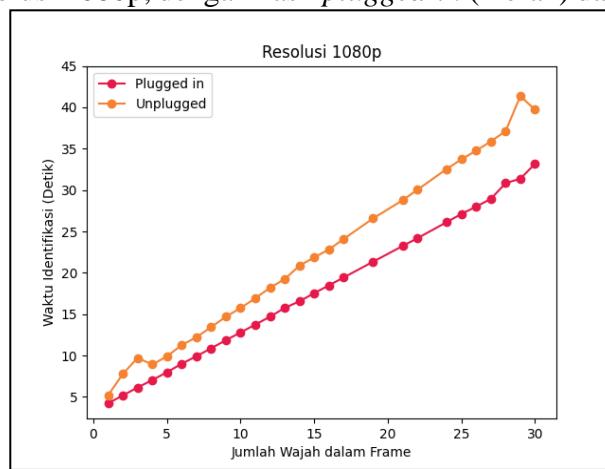
Gambar 37. Hasil Pengujian Kecepatan dalam Resolusi 480p

Gambar 4.33 menunjukkan hubungan jumlah wajah per frame dengan waktu identifikasi pada resolusi 1280×720 , dengan hasil *plugged-in* (hijau) dan *unplugged* (ungu).



Gambar 38. Hasil Pengujian Kecepatan dalam Resolusi 720p

Gambar 4.34 menunjukkan hubungan jumlah wajah per *frame* dengan waktu identifikasi pada resolusi 1080p, dengan hasil *plugged-in* (merah) dan *unplugged* (orange).



Gambar 39. Hasil Pengujian Kecepatan dalam Resolusi 1080p

Data menunjukkan lama waktu sistem mengenali satu wajah per frame pada resolusi 480p, 720p, dan 1080p. Hasilnya disajikan sebagai rata-rata kecepatan pengenalan wajah (Tabel 4.3).

Tabel 5 Hasil Pengujian Kecepatan Pengenalan Wajah

No.	Resolusi Frame	Kondisi Daya	Kecepatan Rata-rata
1	480p (854x480)	<i>Plugged-in</i>	1,04 detik per wajah
2	720p (1280x720)	<i>Plugged-in</i>	1,15 detik per wajah
3	1080p (1920x1080)	<i>Plugged-in</i>	1,39 detik per wajah
4	480p (854x480)	<i>Unplugged</i>	1,3 detik per wajah
5	720p (1280x720)	<i>Unplugged</i>	1,45 detik per wajah
6	1080p (1920x1080)	<i>Unplugged</i>	1,77 detik per wajah

d. Hasil Pengujian Ketangguhan Pengenalan Wajah

Tabel 3.20 menyajikan hasil uji ketangguhan pengenalan wajah pada lima jarak (1,0–5,0 m) dengan tiga percobaan tiap jarak. Hasil uji menunjukkan sistem konsisten mengenali wajah hingga 3,0 m, mulai gagal pada 4,0 m, dan sepenuhnya gagal pada 5,0 m.

Tabel 6 Hasil Pengujian Ketangguhan (Kondisi Jarak)

Identitas Partisipan	Jarak (Meter)	Percobaan ke-1	Percobaan ke-2	Percobaan ke-3
Pramana Afriandy	1,0	TP	TP	TP
	2,0	TP	TP	TP
	3,0	TP	TP	TP
	4,0	FN	TP	FN
	5,0	FN	FN	FN
Doni Winata	1,0	TP	TP	TP
	2,0	TP	TP	TP
	3,0	TP	TP	TP
	4,0	FN	FN	TP
	5,0	FN	FN	FN

Tabel 3.21 menampilkan hasil uji ketangguhan pengenalan wajah terhadap variasi pose (*yaw*, *roll*, *pitch*), masing-masing diuji tiga kali pada dua partisipan. Hasil TP menunjukkan keberhasilan pengenalan, sedangkan FN menunjukkan kegagalan. Sistem baik pada $yaw \leq 60^\circ$ dan sedikit *roll*, namun menurun pada sudut ekstrem, terutama *pitch* menunduk/menengadah dan *yaw* 90° .

Tabel 7 Hasil Pengujian Ketangguhan (Kondisi Pose)

Identitas Partisipan	Pose (Orientasi)	Percobaan ke-1	Percobaan ke-2	Percobaan ke-3
Pramana Afriandy	<i>Yaw</i> 30°	TP	TP	TP
	<i>Yaw</i> 45°	TP	TP	TP
	<i>Yaw</i> 60°	TP	TP	TP
	<i>Yaw</i> 90°	FN	FN	FN
	<i>Roll</i> 1/2	TP	TP	TP
	<i>Roll</i> 2/2	FN	FN	FN
	<i>Pitch</i> Menunduk 1/2	FN	FN	TP
	<i>Pitch</i> Menengadah 2/2	FN	FN	FN
	<i>Pitch</i> Menengadah 1/2	TP	TP	TP
Doni Winata	<i>Pitch</i> Menengadah 2/2	FN	FN	FN
	<i>Yaw</i> 30°	TP	TP	TP
	<i>Yaw</i> 45°	TP	TP	TP
	<i>Yaw</i> 60°	FN	FN	FN
	<i>Yaw</i> 90°	FN	FN	FN
	<i>Roll</i> 1/2	TP	TP	TP
	<i>Roll</i> 2/2	FN	FN	FN
	<i>Pitch</i> Menunduk 1/2	FN	FN	FN
	<i>Pitch</i> Menengadah 2/2	FN	FN	FN

Tabel 3.22 memperlihatkan hasil uji ketangguhan pengenalan wajah dengan atribut masker, kacamata, dan kombinasinya, diuji pada dua partisipan dengan tiga percobaan tiap kondisi. Hasil menunjukkan masker, terutama dengan kacamata, menurunkan akurasi sistem, sedangkan penggunaan kacamata saja masih memungkinkan pengenalan baik, seperti pada partisipan Doni Winata.

Tabel 8 Hasil Pengujian Ketangguhan (Kondisi Masker & Kacamata)

Partisipan	Kondisi	Percobaan 1	Percobaan 2	Percobaan 3
Pramana Afriandy	Masker	FN	FN	FN
	Kacamata	FP	FN	FN
	Masker & kacamata	FN	FN	FN
Doni Winata	Masker	FN	FN	FN
	Kacamata	TP	TP	TP
	Masker & kacamata	FN	FN	FN

e. Hasil Pengujian Akurasi Pencatatan Durasi Aktivitas

Identifikasi dilakukan selama 30 menit pada Ruang A dan Ruang B. Dari pengujian yang dilakukan, (lihat Tabel 4.4) didapati dua jenis aktivitas yang dicatat untuk masing-masing partisipan (kecuali Jessica Chastain), yaitu Berada di Kampus dan Menghadiri Kegiatan.

Tabel 7 Hasil Pengujian Akurasi Pencatatan Aktivitas

Identitas Partisipan	Durasi Berada di Kampus			Durasi Menghadiri Kegiatan		
	Realita (J:M:S)	Hasil (J:M:S)	Pebeda-an (%)	Realita (J:M:S)	Hasil (J:M:S)	Pebeda-an (%)
Mahasiswa Test G	0:30:00	0:29:05	3,10296	1:55:00	1:59:24	3,75427
Mahasiswa Test A	0:10:00	0:08:02	21,8115	1:50:00	2:00:19	8,95868
Mahasiswa Test H	0:04:00	0:03:57	1,25786	1:00:00	0:57:29	4,2843
Mahasiswa Test E	0:25:00	0:25:00	0	0:50:00	1:00:17	18,6489
Mahasiswa Test D	0:26:00	0:40:07	42,7023	1:15:00	1:25:24	12,9676
Mahasiswa Test C	0:20:00	0:24:11	18,9363	1:25:00	1:30:03	5,76978
Mahasiswa Test F	1:00:00	1:00:14	0,38813	1:30:00	1:30:03	0,05554
Mahasiswa Test B	0:05:00	0:19:58	119,893	1:40:00	1:30:28	10,0105

Identitas Partisipan	Durasi Berada di Kampus			Durasi Menghadiri Kegiatan		
	Realita (J:M:S)	Hasil (J:M:S)	Pebeda-an (%)	Realita (J:M:S)	Hasil (J:M:S)	Pebeda-an (%)
	Rata-rata		26,0115	Rata-rata		8,05619

Rata-rata selisih aktivitas Berada di Kampus sebesar 26,01%, sedangkan Menghadiri Kegiatan 8,05%. Berdasarkan Persamaan 3.2, akurasi pencatatan aktivitas mencapai 82,96%.

KESIMPULAN

Aplikasi *desktop* yang dibangun dengan Dlib dan OpenCV mampu mengidentifikasi wajah mahasiswa secara otomatis dengan akurasi dan ketangguhan terukur, serta mendukung pencatatan aktivitas secara *real-time* pada beberapa kamera. Sistem pencatatan aktivitas yang dikembangkan tidak hanya mencatat kehadiran, tetapi juga menghitung durasi berdasarkan waktu masuk dan keluar ruangan, serta mengintegrasikan data kehadiran dengan jadwal kegiatan untuk mengklasifikasikan aktivitas mahasiswa sebagai “menghadiri kegiatan” atau “berada di kampus”. Selain itu, aplikasi web yang dibangun dapat menyajikan informasi aktivitas secara spesifik kepada mahasiswa, orang tua, dan dosen sesuai hak akses masing-masing.

DAFTAR PUSTAKA

- Adam, F. K., Pasaribu, A. F. O., & Wahyudi, A. D. (2023). Aplikasi Monitoring Absensi Karyawan Ditlantas Dengan Penerapan Teknologi GPS (Studi Kasus: Ditlantas Polda Lampung). *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, 4(1), 1–9. <https://doi.org/10.33365/jatika.v4i1.723>
- Mochamad Nashrullah, O., Okvi Maharani, Sp., Abdul Rohman, Sp., Eni Fariyatul Fahyuni, Sp., Nurdyansyah, I., & Sri Untari MPd, R. (2023). *METODOLOGI PENELITIAN PENDIDIKAN (Prosedur Penelitian, Subjek Penelitian, dan Pengembangan Teknik Pengumpulan Data)* Diterbitkan oleh UMSIDA PRESS.
- Mukhlis, I. R., & Santoso, R. (2023). Perancangan Basis Data Perpustakaan Universitas Menggunakan MySQL dengan Physical Data Model dan Entity Relationship Diagram. *Journal of Technology and Informatics (JoTI)*, 4(2), 81–87. <https://doi.org/10.37802/joti.v4i2.330>
- Ni Nyoman Emang Smrti, Andisana I Putu Gd Sukenada, Dwi Trisna Rahayu Ni Kadek, Adnan Adnan, & Julianara Pande Putu Ode. (2023). Flowgorithm Sebagai Penunjang Pembelajaran Algoritma dan Pemrograman. *Jurnal Bangkit Indonesia*, 12(1), 56–64.
- Nur Ichsanudin, M., Yusuf, M., Jurusan Rekayasa Sistem Komputer, S., Teknik Industri, J., AKPRIND Yogyakarta, I., & Artikel, R. (2022). *PENGUJIAN FUNGSIONAL PERANGKAT LUNAK SISTEM INFORMASI PERPUSTAKAAN DENGAN METODE BLACK BOX TESTING BAGI PEMULA INFO ARTIKEL ABSTRAK*. 1(2), 1–8. <https://doi.org/10.55123>
- Pressman, R. S. (2002). *Rekayasa Perangkat Lunak: Pendekatan Praktisi (Buku I)*. Penerbit ANDI.
- Saepudin, S., Pudarwati, E., Warman, C., Sihabudin, S., & Giri, G. (2022). Perancangan Arsitektur Sistem Pemesanan Tiket Wisata Online Menggunakan Framework Zachman. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 11(2), 162–171. <https://doi.org/10.32736/sisfokom.v11i2.1415>

- Susanto, R., & Andriana, A. D. (2016). PERBANDINGAN MODEL WATERFALL DAN PROTOTYPING UNTUK PENGEMBANGAN SISTEM INFORMASI. In *Majalah Ilmiah UNIKOM* (Vol. 14, Issue 1).
- Teoh, K. H., Ismail, R. C., Naziri, S. Z. M., Hussin, R., Isa, M. N. M., & Basir, M. S. S. M. (2021). Face Recognition and Identification using Deep Learning Approach. *Journal of Physics: Conference Series*, 1755(1). <https://doi.org/10.1088/1742-6596/1755/1/012006>
- Tinto, Vincent. (2012). *Completing college : rethinking institutional action*. The University of Chicago Press.