

Aplikasi Smart Presensi Menggunakan *Multiple Face Recognition*

Nazruddin Safaat H.¹, Novri Kurniawan²

Teknik Informatika UIN Sultan Syarif Kasim Riau
JI HR. Soebrantas Panam Km. 15, Riau 28293
e-mail: ¹nazruddin.safaat@uin-suska.ac.id

Abstrak

Universitas Islam Negeri Sultan Syarif Kasim Riau memiliki Fakultas Sains dan Teknologi yang telah menggunakan Aplikasi Smart Presensi untuk mengisi kehadiran pada beberapa mata kuliah. Terdapat beberapa kekurangan dari Aplikasi ini yaitu tidak semua mahasiswa memiliki Android dan koneksi internet, *Multiple face Recognition* dapat menutupi kekurangan tersebut. Seluruh wajah mahasiswa pada satu kelas disimpan ke server menggunakan aplikasi dan dosen membuat data latih pada awal pertemuan mata kuliah. Dosen mengisi daftar kehadiran secara otomatis dengan cara mengambil foto seluruh wajah mahasiswa. Foto tersebut dikirim ke server untuk diproses menggunakan *face recognition*. Hasil dari proses tersebut adalah daftar nama untuk mengisi kehadiran. *OpenCV* adalah library open source yang digunakan untuk pemrosesan gambar. *Histogram Oriented Gradients* adalah metode yang digunakan untuk deteksi wajah atau *face detection*. *Face Recognition* adalah package python yang digunakan untuk pengenalan wajah. Aplikasi diuji menggunakan 2 skenario serta 5 parameter dengan durasi prediksi 40 sampai 80 detik. Skenario cukup cahaya dengan tingkat pencahayaan adalah 83 lux dan maksimal 390 lux menghasilkan akurasi 87,5% sampai 100%. Skenario kurang cahaya dengan tingkat pencahayaan adalah 23 lux dan maksimal 151 lux menghasilkan akurasi 87,5% sampai 93,7.

Kata kunci: Aplikasi Smart Presensi, Multi-face Recognition, OpenCV, Face Recognition, Histogram Oriented Gradients

Abstract

Sultan Syarif Kasim Riau Islamic State University has a Faculty of Science and Technology which has used the Smart Presensi Application to fill attendance in several courses. There are several shortcomings of this application, namely not all students have Android and internet connection, *Multiple face recognition* can cover these shortcomings. All faces of students in one class are saved to the server using the application and the lecturer creates training data at the beginning of the course meeting. The lecturer fills in the attendance list automatically by taking photos of all students' faces. The photo is sent to the server for processing using *face recognition*. The result of the process is a list of names to fill attendance. *OpenCV* is an open source library used for image processing. *Histogram Oriented Gradients* is a method used for *face detection* or *face detection*. *Face Recognition* is a python package used for facial recognition. The application was tested using 2 scenarios and 5 parameters with a prediction duration of 40 to 80 seconds. A sufficient light scenario with an illumination level of 83 lux and a maximum of 390 lux yields an accuracy of 87.5% to 100%. The low light scenario with an illumination level of 23 lux and a maximum of 151 lux yields an accuracy of 87.5% to 93.7%.

Keywords: Smart Presensi Applications, Multi-face Recognition, OpenCV, Face Recognition, Histogram Oriented Gradients

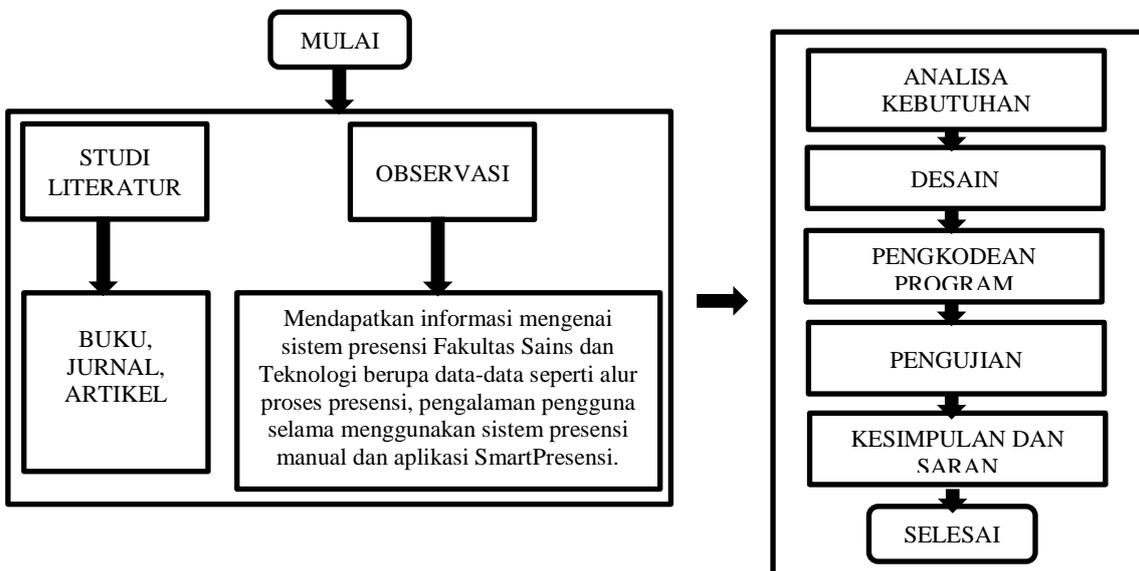
1. Pendahuluan

Aplikasi *Smart Presensi* yang memanfaatkan *QR Code Scanning* dan *Location Based Service* telah digunakan oleh beberapa dosen Teknik Informatika [2]. Secara singkat cara kerja Aplikasi ini adalah mahasiswa menjalankan aplikasi di smartphone dan lokasi dari smartphone harus berada di dalam kelas sehingga dapat mengisi kehadiran dengan cara *Scanning QR Code*. Dari cara kerja tersebut, terdapat beberapa kekurangan. Kekurangan yang pertama adalah aplikasi tersebut hanya dapat dijalankan pada smartphone Android, sehingga mahasiswa yang mempunyai smartphone selain Android tidak bisa menjalankan Aplikasi tersebut atau mengisi kehadiran secara digital. Kekurangan yang kedua adalah tidak semua mahasiswa memiliki koneksi internet, sehingga aplikasi tidak bisa mengisi kehadiran bagi mahasiswa yang tidak memiliki koneksi internet.

Multi-Face Recognition dapat dimanfaatkan untuk menutupi kekurangan dari Aplikasi *Smart Presensi* yang telah digunakan oleh beberapa dosen Teknik Informatika. Penerapan *Multi-Face Recognition* dibantu dengan menggunakan *library OpenCV* dan *Face Recognition* untuk membantu dalam proses pengenalan wajah. *Open Source Computer Vision* yang disingkat *OpenCV* adalah *library* dalam bidang keilmuan computer vision dan machine learning yang secara bebas digunakan untuk mengembangkan aplikasi dalam dua bidang tersebut [3]. *Face Recognition* adalah *package python* untuk pemrosesan wajah yang dikembangkan menggunakan toolkit Dlib [4]. Cara kerja dengan menerapkan *Multi-Face Recognition* di Aplikasi *Smart Presence*, yaitu: Semua gambar wajah mahasiswa yang berada pada satu kelas dilatih dan disimpan dalam basis data, kemudian dosen mengambil foto seluruh mahasiswa di kelas sampai seluruh wajah dideteksi, maka *OpenCV* dan *Face Recognition* membantu dalam mengenali semua wajah pada gambar tersebut dan secara otomatis mengisi kehadiran. Dari cara kerja tersebut, untuk mengisi kehadiran adalah seorang dosen dengan cara mengambil video seluruh mahasiswa dan secara otomatis mengisi kehadiran. Sebelum kehadiran disimpan, dosen membuka aplikasi untuk validasi kehadiran dengan cara membandingkan data wajah yang dideteksi sama dengan wajah yang ada di basis data. Terkait dengan penelitian *Face Recognition*, beberapa penelitian telah dilakukan dengan menerapkan *Face Recognition* untuk mengisi kehadiran [5][6].

2. Metode Penelitian

Metode penelitian menjelaskan rencana kerja yang akan dilakukan dalam penelitian agar hasil yang didapatkan sesuai dengan yang diharapkan. Tahapan dari metode penelitian ini adalah identifikasi masalah, studi pustaka, pengumpulan data, analisa kebutuhan, desain, pengodean program, pengujian, kesimpulan dan saran.



Gambar 1. Tahapan Penelitian

2.1. Identifikasi Masalah

Identifikasi masalah merupakan suatu tahapan awal yang harus ada pada sebuah penelitian. Pada tahap ini, dilakukan kegiatan penelitian untuk mendapatkan inti/gambaran permasalahan dan mencari solusi dari permasalahan tersebut. Masalah yang diangkat pada penelitian ini adalah Bagaimana menerapkan *Multiple face recognition* pada aplikasi *Smart Presensi* untuk mengisi kehadiran menggunakan presensi wajah.

2.2. Studi Pustaka

Studi pustaka merupakan proses pencarian referensi teori yang berhubungan dengan penelitian yang sedang dilakukan. Studi pustaka ini dilakukan untuk mengetahui teori, metode dan konsep sesuai dengan apa yang sedang diteliti. Studi pustaka dapat bersumber dari buku, jurnal dan artikel.

2.3. Pengumpulan Data

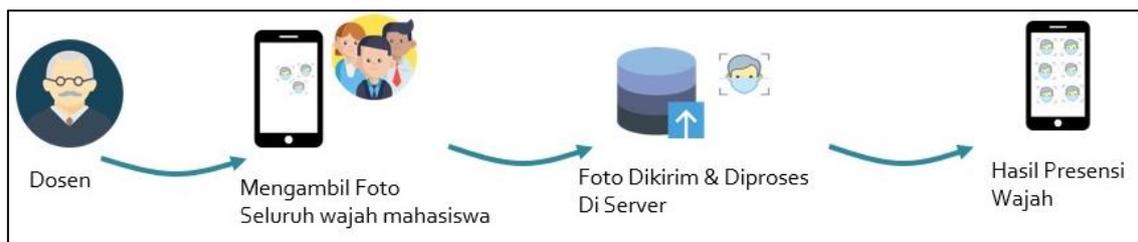
Langkah awal yang harus dilakukan untuk menentukan data apa saja yang dibutuhkan pada penelitian adalah dengan mengumpulkan data. Pada penelitian ini pengumpulan data dilakukan dengan teknik studi literatur dan observasi. Studi literatur dilakukan dengan membaca teori dan penelitian terkait. Observasi dilakukan dengan cara analisa sistem lama dan fitur baru yang akan digunakan pada sistem lama tersebut. Sehingga akan diperoleh bagian sistem mana yang dianggap baik dan dianggap kurang baik.

2.4. Analisa Kebutuhan

Tahap pengumpulan kebutuhan pengguna akan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar pengguna paham dengan perangkat lunak yang dibutuhkan. Data yang didapat pada penelitian ini adalah hasil observasi, data akan diolah untuk mendapatkan solusi dari masalah yang ada. Spesifikasi kebutuhan perangkat lunak pada tahap analisa akan didokumentasikan. Analisa pembangunan sistem menggunakan metode UML diantaranya: *Usecase Diagram*, *Usecase Scenario*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*.

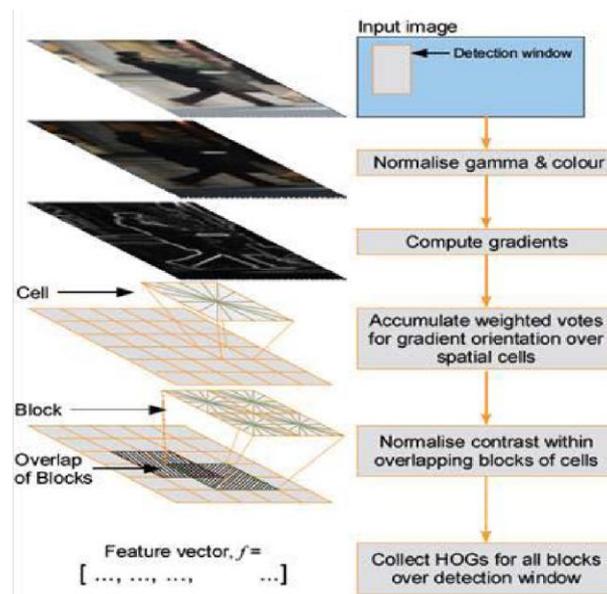
2.5. Desain

Pada tahap desain akan dilakukan proses pembuatan desain program perangkat lunak berdasarkan kebutuhan perangkat lunak dari tahap analisis. Sehingga tahap desain sistem akan menghasilkan representasi desain dari tahap analisis kebutuhan. Desain perangkat lunak yang dihasilkan pada tahap ini akan didokumentasikan. Metode pendekatan sistem yang adalah pendekatan dengan *Object Oriented* dengan menggunakan *Object Oriented Analysis (OOA)* dan *Object Oriented Design (OOD)* yang akan divisualisasikan dengan *Unified Modeling Language (UML)*. Model desain UML yang akan digunakan adalah *Usecase Diagram*, *Usecase Scenario*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram* dan *Deployment Diagram*. Berikut alur aplikasi yang akan dibangun, yaitu:



Gambar 2. Alur Presensi Menggunakan *Multiple face recognition*

Pada penelitian ini, *face detection* menggunakan metode *histogram oriented gradients* yang merupakan metode deteksi pada gambar menggunakan arah intensitas gradien dan arah sudut *pixel*. Berikut adalah tahapan dari metode HOG.



Gambar 3. Flowchart HOG [7]

2.6. Pengodean Program

Pengkodean program dilakukan dengan mentranslasikan desain ke dalam program perangkat lunak. Sehingga akan diperoleh hasil program komputer sesuai dengan rancangan pada tahap desain yang sudah dilakukan. Tahap pengkodean program (*coding*) akan dilakukan pada aplikasi *mobile* dan *web*, aplikasi *mobile* dalam kasus ini adalah Android yang akan diprogram menggunakan bahasa Java dan *web* yang akan diprogram menggunakan bahasa PHP dan Python, *database* MySQL sebagai sistem manajemen basis data.

2.7. Pengujian

Pada tahap pengujian dilakukan pengujian terhadap perangkat lunak guna memastikan fungsi-fungsi yang ada pada perangkat lunak berjalan dengan semestinya dan diujikan sebagaimana kondisi aslinya. Pengujian akan dilakukan terhadap perangkat lunak dan pengguna, yakni dengan cara *Black Box* dan *User Acceptance Test* (UAT). Sehingga akan mengurangi kesalahan dan akan diperoleh hasil yang sesuai diinginkan. Skenario pengujian dilakukan dalam ruangan terhadap beberapa parameter yaitu jumlah wajah dan status lampu dalam keadaan hidup dan mati. Pengujian menggunakan data yang diambil dari kelas laboratorium dengan jumlah 24 wajah dan 13 wajah dalam satu foto. Tahap Pengujian menggunakan Skenario dan parameter menghasilkan akurasi yang merupakan satuan ukur untuk menghitung hasil dari pengujian. Persamaan yang digunakan adalah persentase jumlah pengujian benar dibagi total data uji [8].

$$Akurasi = \frac{\text{jumlah pengujian benar}}{\text{total data uji}} \times 100\% \quad (2.1)$$

2.8. Kesimpulan dan Saran

Tahapan kesimpulan dan saran ini merupakan langkah terakhir dari penelitian ini. Akan diperoleh kesimpulan dari penelitian yang dilakukan ini dan akan disebutkan saran-saran yang membangun terhadap penelitian ini agar menimbulkan penelitian baru yang dapat memperbaiki penelitian sebelumnya.

3. Hasil dan Pembahasan

3.1. Implementasi

Setelah tahap analisa dan perancangan selesai dilakukan maka dilanjutkan ke tahap implementasi dan pengujian. Dalam tahap ini dilakukan pengujian terhadap fitur-fitur yang tersedia pada aplikasi, kemudian dilakukan pengamatan hasil pengujian tersebut sehingga diketahui fitur-fitur yang masih memiliki kekurangan dan dapat dilakukan pengambilan

kesimpulan. Pengujian aplikasi menggunakan perangkat PC dan Android. Sebelum melakukan proses presensi wajah, mahasiswa harus mengunggah foto wajah sebanyak 4 kali yaitu wajah lurus ke kamera dengan ekspresi datar dan senyum, serong kanan dan kiri.



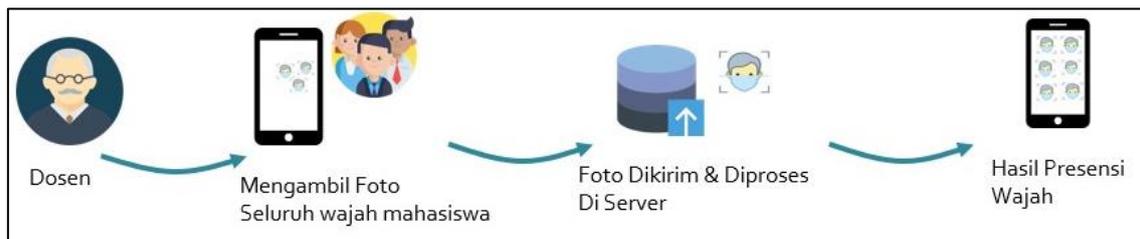
Gambar 4. Alur Simpan Data Wajah Mahasiswa

Setelah data wajah mahasiswa diunggah ke server, maka server akan melakukan cron job untuk membuat data latih. Pada penelitian ini, data latih akan dibuat oleh dosen dengan cara membuka halaman web utk data latih.



Gambar 5. Alur Membuat Data Latih Wajah

Data latih berhasil dibuat, maka dosen dapat melakukan proses presensi menggunakan face recognition dengan cara mengirim foto ke server untuk diproses. Jika terdapat banyak mahasiswa di kelas, maka dosen dapat mengirim 2 foto ke server untuk diproses.

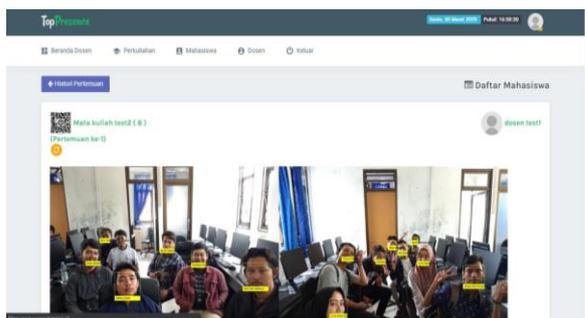


Gambar 6. Alur Proses Presensi Wajah



Gambar 7. Tampilan Foto Kelas

Setelah selesai diproses di server, maka hasil dari proses *face recognition* dapat dilihat di *android* dan *web*. Berikut hasil implementasi aplikasi *web* dan *android*.



Gambar 8. Tampilan Halaman Presensi Detail



Gambar 9. Tampilan Hasil Presensi Wajah

Halaman presensi detail merupakan halaman yang menampilkan detail dari presensi kehadiran sebuah matakuliah, yaitu menampilkan mahasiswa yang hadir dan tidak hadir. Halaman hasil presensi wajah adalah halaman yang menampilkan hasil dari prediksi menggunakan *face recognition* terhadap foto yang diproses. Terdapat tombol validasi untuk menghapus data prediksi yang tidak akurat.

3.2. Pengujian

Pengujian aplikasi menggunakan skenario digunakan untuk mengukur akurasi aplikasi, ketika aplikasi akan digunakan untuk mengisi kehadiran di kelas menggunakan beberapa parameter atau situasi yang berbeda. Aplikasi *Smart Presensi Multiple face recognition* terdapat fitur baru menggunakan *face recognition* untuk mengisi presensi, pada penelitian ini menggunakan skenario kualitas cahaya dan parameter jarak kamera, posisi tempat duduk, atribut wajah, gaussian blur dan data latih dalam foto untuk mengukur tingkat akurasi dari fitur *face recognition*. Pada penelitian ini, data uji yang digunakan adalah foto mahasiswa dalam kelas laboratorium dan ruang kelas gedung baru dengan jumlah maksimal 24 orang dan 13 orang dalam satu foto. Ukuran foto wajah tidak ditentukan karena aplikasi akan otomatis melakukan pengecilan ukuran pada foto yang akan dilatih dan diuji. Pengukuran akurasi dilakukan dengan menghitung jumlah deteksi dan prediksi wajah yang benar.

Tabel 1. Hasil Skenario Cukup Cahaya

No	Parameter	Akurasi				
		2 meter	3 meter	4 meter	5 meter	
1	Jarak kamera	93.7%	87.5%	94,1%	75%	
		90%				
2	Posisi tempat duduk	90%				
		Kacamata		Kumis dan Jenggot		
3	Atribut wajah	93.7%		100%		
		Skala 1	Skala 2	Skala 3	Skala 4	Skala 5
4	Gaussian blur	85.1%	70.3%	40.7%	22.2%	11.1%
		85%				
5	Data latih	85%				

Pengujian menggunakan 5 parameter yaitu jarak kamera, posisi tempat duduk, atribut wajah, gaussian blur dan data latih menghasilkan tingkat akurasi yang berbeda. Jarak kamera yang memiliki tingkat akurasi tinggi adalah jarak 2 sampai 4 meter, sehingga jika lebih dari 5 meter maka aplikasi memiliki tingkat akurasi yang rendah. Posisi tempat duduk memiliki tingkat akurasi yang tinggi karena mengganti posisi tempat duduk tidak mempengaruhi tingkat akurasi. Atribut wajah memiliki tingkat akurasi yang tinggi, membuktikan bahwa penambahan atribut seperti kacamata dan kumis jenggot tidak mempengaruhi akurasi aplikasi. *Gaussian blur* memiliki tingkat akurasi yang berbeda, tapi aplikasi hanya memiliki tingkat akurasi yang tinggi pada skala 1 dan 2. Aplikasi tidak dapat mendeteksi dan prediksi wajah pada skala 3 sampai 5. Pada parameter data latih, jumlah data latih kurang hingga setengah total data latih. Aplikasi dapat mengenali wajah yang telah dilatih dan wajah yang belum dilatih. Aplikasi bisa mendeteksi dan

mengenali wajah pada skenario cukup cahaya menggunakan 5 parameter dengan rentang waktu 40 detik sampai dengan 1 menit 20 detik. Perbedaan waktu saat aplikasi mendeteksi dan mengenali wajah, disebabkan oleh jumlah wajah dalam foto yang diproses.

Tabel 2. Hasil Skenario Kurang Cahaya

No	Parameter	Akurasi				
		2 meter	3 meter	4 meter	5 meter	
1	Jarak kamera	93.7%	87.5%	93,7%	75%	
		90%				
2	Posisi tempat duduk	90%				
		Kacamata		Kumis dan Jenggot		
3	Atribut wajah	93.7%		87,5%		
		Skala 1		Skala 2	Skala 3	Skala 4
4	Gaussian blur	88,89%	66,67%	37,03%	18,5%	0,37%
		80%				
5	Data latih	80%				

Pengujian menggunakan skenario kurang cahaya 5 parameter yaitu jarak kamera, posisi tempat duduk, atribut wajah, gaussian blur dan data latih menghasilkan tingkat akurasi yang berbeda. Jarak kamera yang memiliki tingkat akurasi tinggi adalah jarak 2 sampai 4 meter, sehingga jika lebih dari 5 meter maka aplikasi memiliki tingkat akurasi yang rendah. Posisi tempat duduk memiliki tingkat akurasi yang tinggi karena mengganti posisi tempat duduk tidak mempengaruhi tingkat akurasi. Atribut wajah memiliki tingkat akurasi yang tinggi, membuktikan bahwa penambahan atribut seperti kacamata dan kumis jenggot tidak mempengaruhi akurasi aplikasi. Gaussian blur memiliki tingkat akurasi yang berbeda, tapi aplikasi hanya memiliki tingkat akurasi yang tinggi pada skala 1. Aplikasi tidak dapat mendeteksi dan prediksi wajah pada skala 2 sampai 5. Pada parameter data latih, jumlah data latih kurang hingga setengah total data latih. Aplikasi dapat mengenali wajah yang telah dilatih dan wajah yang belum dilatih. Aplikasi bisa mendeteksi dan mengenali wajah pada skenario kurang cahaya menggunakan 5 parameter dengan rentang waktu 40 detik sampai dengan 1 menit 20 detik. Perbedaan waktu saat aplikasi mendeteksi dan mengenali wajah, disebabkan oleh jumlah wajah dalam foto yang diproses. Pada skenario kurang cahaya waktu proses tidak memiliki perbedaan dengan skenario cukup cahaya.

4. Kesimpulan

Kesimpulan yang diperoleh bahwa implementasi *face recognition* pada aplikasi *Smart Presensi* untuk mengisi kehadiran telah berhasil dilakukan. Aplikasi sudah diuji menggunakan metode *black-box testing* dan 2 skenario pengujian yang memiliki hasil sesuai dengan yang dirancang dan diharapkan. *Face recognition* membantu dosen dalam mengisi kehadiran mahasiswa, sehingga kekurangan aplikasi yang hanya bisa di perangkat android dan memerlukan koneksi internet dapat ditutupi.

Daftar Pustaka

- [1] B. A. A. dan Kemahasiswaan, "Panduan dan Informasi Akademik," 2015, p. 200.
- [2] R. D. Novendra, "Aplikasi Smart Presence Menggunakan QR Code dan LBS," Islamic State University Sultan Syarif Kasim, 2018.
- [3] D. L. Baggio and dkk, *Mastering OpenCV with Practical Computer Vision Projects*. Birmingham B3 2PB, UK: Packt Publishing Ltd., 2012.
- [4] A. Geitgey, "face_recognition," *Github.com*, 2020. [Online]. Available: https://github.com/ageitgey/face_recognition. [Accessed: 16-Dec-2019].
- [5] Y. Mao, H. Li, and Z. Yin, "Who Missed The Class ? - Unifying Multi-Face Detection, Tracking, and Recognition in Videos Department of Computer Science Missouri University of Science and Technology, USA," 2014.
- [6] T. Mantoro and M. A. Ayu, "Multi-Faces Recognition Process Using Haar Cascades and Eigenface Methods," *2018 6th Int. Conf. Multimed. Comput. Syst.*, pp. 1–5, 2018.
- [7] S. Kapur and N. Thakkar, *Mastering OpenCV Android Application Programming*. Birmingham B3 2PB, UK: Packt Publishing Ltd., 2015.
- [8] A. Dinda, B. Sudirman, Y. A. Sari, and F. Utamingrum, "Pengenalan Wajah dengan Pose Unik menggunakan Metode Learning Vector Quantization," vol. 3, no. 1, pp. 884–891, 2019.