

Terbit online pada laman web jurnal: <http://jurnal.iaii.or.id>

JURNAL RESTI

(Rekayasa Sistem dan Teknologi Informasi)

Vol. 5 No. 1 (2021) 31 - 38

ISSN Media Elektronik: 2580-0760

Presensi Kelas Berbasis Pola Wajah, Senyum dan Wi-Fi Terdekat dengan Deep Learning

Miftakhurrokhmat¹, Rian Adam Rajagede², Ridho Rahmadi³

^{1 2 3}Magister Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia

¹18917214@students.uui.ac.id, ²rian.adam@uui.ac.id, ³ridho.rahmadi@uui.ac.id*

Abstract

Students' attendance in class is often mandatory in education and becomes a benchmark for assessing students. Sometimes there are still fraudulent practices by students to achieve minimum attendance. From the administrative perspective, a paper-based presence system is potentially wasteful and extends the administrative stage because it requires manual recapitulation. This study aims to design a class attendance application based on facial pattern recognition, smile, and closest Wi-Fi. The method used in this research is a deep learning approach with CNN based architecture, FaceNet, to recognize faces. In addition to facial images, the system will also validate the attendance with location and time data. Location data is obtained from matching SSID from the database, and time data is taken when the user sends attendance data through API. This attendance system consists of three applications: web, mobile, and services installed on a mini-computer, which are integrated to sending attendance data to the academic system automatically. As confirmation, students are required to smile selfies to strengthen the validity of their presence. The testing model's accuracy results are 92.6%, while for live testing accuracy the model obtained 66.7%.

Keywords: presence, smiling, face recognition, convolutional neural network, deep learning

Abstrak

Kehadiran mahasiswa dalam suatu pembelajaran di kelas seringkali menjadi syarat wajib dalam dunia pendidikan, dan menjadi tolak ukur dalam menilai mahasiswa. Terkadang masih dijumpai praktik curang oleh mahasiswa dalam presensi agar mencapai kehadiran minimal. Dari sisi administrasi, presensi berbasis kertas berpotensi pemborosan dan juga memperpanjang tahapan administrasi karena membutuhkan rekapitulasi manual. Penelitian ini bertujuan untuk merancang bangun aplikasi presensi kelas berbasis pengenalan pola wajah, senyum, dan Wi-Fi terdekat. Metode yang digunakan dalam penelitian ini adalah pendekatan *Deep Learning* dengan arsitektur *CNN FaceNet* untuk mengenali wajah. Selain gambar wajah, sistem juga akan memvalidasi presensi dengan kesesuaian lokasi dan waktu. Data lokasi diperoleh dari pencocokan *SSID* dengan database, dan data waktu diambil saat mahasiswa mengirimkan data kehadiran melalui *API*. Sistem presensi ini terdiri dari tiga aplikasi yaitu web, *mobile*, dan *service* yang dipasang di komputer mini, yang saling terintegrasi untuk mengirimkan data presensi ke sistem akademik secara otomatis. Sebagai konfirmasi, siswa diwajibkan *selfie* tersenyum untuk memperkuat validitas kehadiran. Sistem terintegrasi ini masih dalam bentuk purwarupa. Hasil akurasi dari *testing* model sebesar 92,6% sedangkan untuk *testing live* akurasi sebesar 66,7%. Nilai *testing live* lebih kecil dan cukup jauh dari *testing* model dapat diartikan hasil *training* model terlalu *overfitting*.

Kata kunci: presensi, senyum, pengenalan wajah, *convolutional neural network*, *deep learning*

1. Pendahuluan

Hingga saat ini, terdapat berbagai cara yang dilakukan untuk kegiatan presensi, baik di lingkungan pendidikan,

pemerintahan, atau bidang lainnya. Sebagian menggunakan teknologi terkini seperti pengenalan sidik jari (*finger print*) [1], namun tidak sedikit pula institusi

yang masih menggunakan sistem manual seperti tanda tangan pada buku dan kemudian direkapitulasi untuk memperoleh laporan akhir dari kehadiran. Proses presensi secara manual memiliki banyak kekurangan seperti pemborosan kertas, rekapitulasi yang memakan waktu, sulit untuk diintegrasikan dengan sistem lain, dan rentan terhadap pemalsuan.

Sistem presensi yang berbasis sidik jari, seringkali dibangun pada *platform* yang *hardcoded* (tidak mengizinkan komunikasi dengan sistem atau piranti lain) sehingga menyulitkan ketika harus diintegrasikan dengan sistem utama pada sebuah institusi. Sistem presensi seperti ini juga rentan terhadap peretasan atau pemalsuan sidik jari [2,3]. Alternatif lain seperti presensi menggunakan kartu memiliki isu yang sama: dibangun pada sebuah sistem yang tidak mudah untuk diintegrasikan dengan sistem utama seperti sistem informasi akademik. Belum lagi penggunaan kartu sangatlah rentan terhadap pemalsuan karena kartu bisa dengan mudah dipindahtangankan. Lebih jauh lagi, dibutuhkan biaya yang relatif tidak sedikit untuk mencetak kartu [4,5]. Dalam hal ini, dibutuhkan sebuah solusi untuk presensi yang menjawab isu-isu permasalahan diatas.

Beberapa penelitian sebelumnya [6,7], memberikan solusi presensi dengan menghadirkan sistem pengenalan pola wajah berbasis *mobile*. Alternatif ini cukup efektif untuk diintegrasikan dengan sistem utama pada sebuah institusi, di samping penghematan penyediaan alat presensi. Namun, beberapa faktor [8] turut berpengaruh dalam keakuratan pengenalan wajah. Sehingga perlu dilakukan pengembangan lebih lanjut lagi untuk sebuah aplikasi pengenalan wajah seperti memanfaatkan CCTV dan menjadikannya berbasis *mobile*.

Deteksi wajah dapat dipandang sebagai sebuah metode klasifikasi pola dimana inputnya adalah citra dan outputnya adalah label dari citra tersebut. Pendeteksian wajah (*face detection*) adalah salah satu tahap awal yang sangat penting sebelum dilakukan proses pengenalan wajah (*face recognition*). Bidang-bidang penelitian yang berkaitan dengan pemrosesan wajah (*face processing*) diantaranya: pengenalan wajah (*face recognition*), autentikasi wajah (*face authentication*), lokalisasi wajah (*face localization*), penjejakan wajah (*face tracking*), dan pengenalan ekspresi wajah (*facial expression recognition*) [9].

Face Recognition (FR) adalah teknik biometrik yang sangat baik untuk otentikasi identitas. Teknologi FR dapat diterapkan untuk pencatatan kehadiran secara otomatis di lingkungan akademis. Terdapat beberapa keuntungan menggunakan sistem kamera, seperti menghemat waktu dan tenaga, memberikan bukti kuat untuk penjaminan mutu dan tugas manajemen sumber daya manusia, menghindari perantaraan penyakit menular [10].

Pada penelitian sebelumnya [11], dikembangkan sebuah sistem presensi mahasiswa berdasarkan pengenalan wajah menggunakan webcam dan mini PC dengan metode *Haar Cascade* untuk pendeteksian wajah, *Local Binary Pattern* dan *K-Nearest Neighbor* untuk pengenalan wajah. Nilai akurasi pengenalan wajah yang didapatkan sebesar 78,125% dengan $k=2$. Selanjutnya pada penelitian yang lain [12], digunakan metode *Haar Cascade* untuk pendeteksian wajah pada video dan *Deep Learning* untuk identifikasi. Hasil ujicoba dari penelitian tersebut mendapatkan tingkat akurasi 99,6% dengan syarat komposisi warna dan tingkat cahaya video sama dengan data *training*. Peneliti akan menggunakan metode *DLib* untuk pendeteksian wajah, dan metode *Deep Learning* dengan arsitektur berbeda yaitu *FaceNet* dengan harapan mampu memperbaiki kelemahan yang berkaitan dengan komposisi warna dan tingkat cahaya. Selain itu, penelitian ini akan mengintegrasikan beberapa komponen yaitu CCTV untuk menangkap gambar, mini komputer untuk memproses pengenalan wajah, dan *smartphone* untuk konfirmasi.

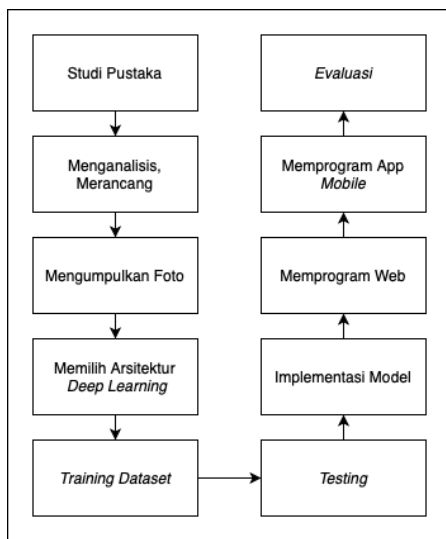
Penelitian ini bertujuan untuk merancang bangun sebuah sistem untuk melakukan presensi berbasis pola wajah dan ekspresi senyum, lokasi dan waktu yang tepat sesuai jadwal. Sebuah contoh untuk mengilustrasikan cara bagaimana aplikasi ini bekerja adalah sebagai berikut: seorang mahasiswa yang masuk kedalam ruang kelas akan mendapatkan notifikasi di *smartphone*, untuk mengkonfirmasi bahwa mahasiswa tersebut masuk ke kelas yang telah sesuai. Konfirmasi dapat dilakukan dengan tersenyum ke arah *smartphone* atau dengan menekan tombol *check-in*. Kemudian ketika kelas telah berakhir dan mahasiswa keluar, berdasarkan data posisi mahasiswa di luar kelas yang didapat dari *access point* (Wi-Fi) di ruangan tersebut, mahasiswa tersebut akan mendapatkan notifikasi untuk *check-out*. Proses *check-out* dapat dilakukan dengan tersenyum ke arah *smartphone*, yang akan dilanjutkan dengan sebuah notifikasi untuk menginformasikan jika yang bersangkutan telah presensi di kelas tersebut. Data presensi tersebut akan secara otomatis disimpan ke dalam sistem informasi akademik.

2. Metode Penelitian

Bagian ini berisi diagram alir langkah penelitian secara umum pada sistem presensi kelas berbasis pola wajah tersenyum dan Wi-Fi terdekat ditunjukkan pada Gambar 1.

Adapun secara garis besar urutan penelitian adalah sebagai berikut: Studi Pustaka dengan melakukan pencarian jurnal dan artikel yang memiliki kemiripan dengan penelitian terkait dan selanjutnya melakukan evaluasi terhadap metodologi yang dilakukan. Analisis perancangan meliputi analisis kebutuhan, yaitu *software*, *hardware*, dan teknologi yang digunakan untuk integrasi baik sisi *database*, *mobile*, web, IP CCTV maupun *network*. Pengumpulan data dilakukan

secara kolektif, mahasiswa dapat mengirimkan data berupa foto maupun video. Tahap selanjutnya adalah pemilihan arsitektur yang akan digunakan, dalam hal ini FaceNet yang merupakan salah satu arsitektur *deep learning*. *Training* dan *testing dataset* dipersiapkan dari tahapan sebelumnya yaitu pengumpulan foto. Implementasi model hasil *training dataset* diterapkan menggunakan *script* Python sebagai *background service* pada mini komputer. Aplikasi web untuk dosen dibangun diatas *framework* Laravel (PHP) dan diletakkan pada server yang sama dengan database. Aplikasi Mobile untuk mahasiswa dibangun diatas *cross platform framework* FireMonkey (Delphi). Tahap terakhir yaitu evaluasi, dilakukan dengan memastikan sistem terintegrasi dan bekerja dengan baik.



Gambar 1. Diagram Alir Penelitian

2.1. Analisis Perancangan

Dalam memenuhi perancangan dan implementasi sistem dibutuhkan perangkat keras dan lunak yang akan digunakan yaitu mini komputer dengan spesifikasi seperti pada Tabel 1 dan untuk *training dataset* dengan spesifikasi seperti pada Tabel 2.

Tabel 1. Spesifikasi *Hardware* Mini Komputer

No	Komponen	Spesifikasi
1	Processor	Intel Atom Z8350 CPU, quad core
2	Memory	4GB RAM
3	Storage	SSD 64GB
4	Sistem Operasi	Windows 10
5	Graphics	Intel HD Graphics 400

Tabel 2. Spesifikasi *Hardware* untuk Training

No	Komponen	Spesifikasi
1	Processor	2,4 GHz Intel Core i7
2	Memory	8 GB 1600 MHz DDR3
3	Storage	SSD 250GB
4	Sistem Operasi	macOS Mojave
5	Graphics	Intel HD Graphics 4000 NVIDIA GeForce GT 650M

Perancangan sangat penting dilakukan untuk mempermudah merancang bangun suatu sistem.

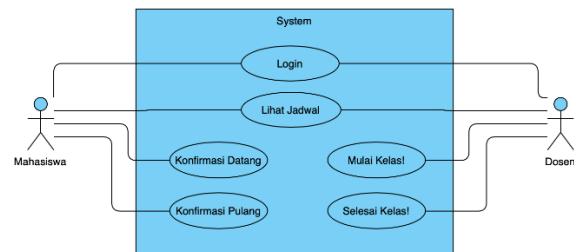
Beberapa perancangan yang dilakukan seperti perancangan basis data berupa tabel-tabel dan *use case diagram*. Gambar 2 menunjukkan contoh tabel utama untuk presensi dan notifikasi.

Total tabel yang dibentuk baik sebagai master data maupun transaksi sejumlah 13, yaitu terdiri dari tabel dosen, tabel dosen_login, tabel dosen_matkul, tabel jadwal_kuliah, tabel mahasiswa, tabel mahasiswa_login, tabel mahasiswa_matkul, tabel mahasiswa_notif, tabel mata_kuliah, tabel presensi, tabel presensi_rekap, tabel ruangan, dan tabel tahun_ajaran.

Kemudian pada Gambar 3 ditunjukkan *use case diagram* mahasiswa dan dosen untuk memberikan gambaran fitur yang ada. Dosen selaku regulator memiliki fungsionalitas dalam hal memulai kelas dan menyelesaikan kelas. Mahasiswa memiliki fungsionalitas dalam hal konfirmasi kedatangan dan konfirmasi pulang. Baik dosen maupun mahasiswa memiliki fungsionalitas yang sama yaitu login ke sistem.

adb_usmileypresensi		adb_usmileypresensi_rekap		adb_usmileymahasiswa_notif	
123 id	bigint NOT NULL	123 id	bigint NOT NULL	123 id_notif	bigint NOT NULL
123 nim	varchar(20)	123 tahun_id	int	123 mahasiswa_id	int
123 tahun_id	int	123 matkul_id	int	123 keterangan	varchar(100)
123 matkul_id	int	123 nim	varchar(20)	123 pesan	text
123 ruangan_id	int	123 total_masuk	int	123 is_read	enum('0','1')
123 dosen_id	int	123 total_jin	int	123 jadwal_id	varchar(100)
123 jadwal_id	int	123 total_sakit	int	123 action_code	varchar(100)
123 pekan	int	123 total_alpha	int	123 cdt	timestamp NOT NULL
123 tanggal	date	123 boleh_ujian	enum('0','1')	123 udt	datetime
123 hari	varchar(20)	123 udt	datetime		
123 jam_masuk	time				
123 jam_keluar	time				
123 is_senyum_masuk	enum('0','1')				
123 is_senyum_keluar	enum('0','1')				
123 is_telat	enum('0','1')				
123 toL_menit_telat	time				
123 base64_senyum_masuk	longtext				
123 base64_senyum_keluar	longtext				

Gambar 2. Tabel Utama Presensi dan Notifikasi

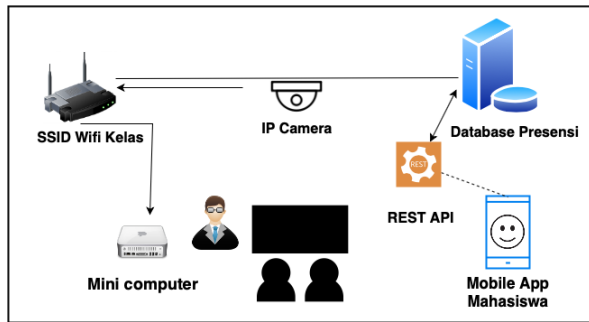


Gambar 3. *Use Case* Mahasiswa dan Dosen

2.2. Gambaran Umum Sistem

Pada Gambar 4 diperlihatkan ilustrasi topologi sistem presensi berbasis *deep learning*. Sistem ini diterapkan di dalam ruangan yang memiliki sebuah IP CCTV, sebuah mini komputer dan sebuah *wireless access point* yang terhubung dengan jaringan kampus. Pada mini komputer inilah *script* Python yang menerapkan model *deep learning* ditempatkan. Mini komputer terkoneksi dengan IP CCTV dan jaringan kampus. Proses pengiriman data presensi ke *database* pusat dilakukan melalui *Representational State Transfer Application Programming Interface* (REST API). Melalui REST API ini aplikasi *mobile* mahasiswa terhubung untuk melihat data presensi dan konfirmasi kehadiran.

Konfirmasi hanya dapat dilakukan melalui jaringan kampus.



Gambar 4. Topologi Sistem

2.3. Pengumpulan *Dataset* dan *Training*

Dataset yang digunakan pada penelitian ini berupa citra wajah yang diperoleh melalui dua cara. Cara pertama dilakukan dengan mengumpulkan foto mahasiswa. Sedangkan cara kedua dilakukan dengan mengumpulkan video mahasiswa berdurasi sekitar 1-2 menit dengan beragam pose. Video tersebut kemudian diolah dengan cara memotong *frame* per detik untuk memperoleh sekitar 60-120 foto per video. Sebelum digunakan untuk *training set* dan *testing set*, foto-foto tersebut harus dilakukan pra-pemrosesan terlebih dahulu menyesuaikan arsitektur yang dipilih yaitu FaceNet dengan dimensi 160x160 piksel dengan menggunakan *script* Python yang menerapkan *face detection* MTCNN [13].

Training dalam *Deep Learning* adalah proses melatih arsitektur *Deep Learning* menggunakan *dataset* dan menghasilkan suatu model pembelajaran. Untuk menghasilkan model tersebut dibutuhkan *hardware* dengan spesifikasi seperti pada Tabel 2. Selain itu, beberapa *library* digunakan seperti Tensorflow dan Keras ditunjukkan pada Tabel 3.

Tabel 3. *Library* yang digunakan untuk *training*

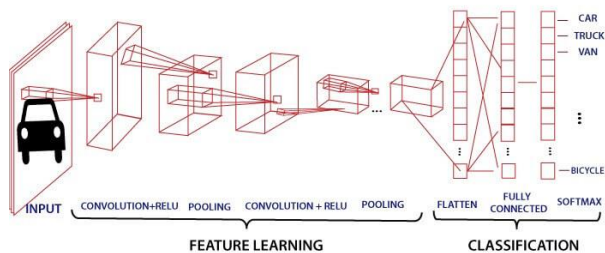
No	<i>Library</i>	Versi
1	grpcio	1.32.0
2	h5py	2.10.0
3	imageio	2.9.0
4	Keras-Preprocessing	1.1.2
5	numpy	1.16.0
6	opencv-contrib-python	4.4.0.44
7	Pillow	7.2.0
8	protobuf	3.13.0
9	scikit-learn	0.23.2
10	scipy	1.2.2
11	tensorflow	1.7.0

2.4. Convolutional Neural Network

Convolutional Neural Network (CNN) adalah salah satu arsitektur jaringan saraf tiruan yang termasuk dalam *Deep Learning* dan didesain khusus untuk menangani data berbentuk larik. CNN menerapkan operasi konvolusi pada satu atau lebih lapisannya yang terinspirasi oleh sistem saraf biologis [14]. Pada CNN setiap *neuron* dipresentasikan dalam bentuk dua

dimensi, sehingga metode ini cocok untuk pemrosesan dengan *input* berupa citra [15].

Struktur CNN dapat dilihat memiliki dua tahapan, yakni tahapan ekstraksi fitur (*feature learning*) dan proses klasifikasi seperti yang tampak pada Gambar 5 [16]. Proses ekstraksi dalam CNN dilakukan pada lapisan konvolusi dan *pooling* yang menghasilkan *feature maps*. CNN bekerja secara hierarki, sehingga *output* pada lapisan konvolusi pertama digunakan sebagai *input* pada lapisan konvolusi selanjutnya. Pada proses klasifikasi, *output* dari proses ekstraksi fitur diubah bentuknya menjadi satu dimensi lalu diinputkan ke *classifier* yang bisa berupa *fully-connected layer* seperti pada Gambar 5 atau *classifier* lain [15].



Gambar 5. Ilustrasi Convolutional Neural Network [17]

2.5. Arsitektur FaceNet

FaceNet merupakan sistem pengenalan wajah yang dikembangkan oleh peneliti Google. FaceNet mengekstrak fitur wajah menjadi vektor menggunakan arsitektur *deep Convolutional Neural Network* (*deep CNN*). Vektor nilai atau *vector embedding* yang dihasilkan dapat memetakan kemiripan wajah yang memiliki kedekatan posisi pada *embedding space*. Model *Deep CNN* yang digunakan pada FaceNet bisa berupa ZF-Net atau Inception [18].

Pada penelitian ini kami menggunakan FaceNet pada tahapan *feature learning*. Hasil dari FaceNet yang berupa vektor sebanyak 128 elemen atau *Face embedding* diklasifikasi menggunakan SVM. Model SVM mengklasifikasi identitas wajah dari *vector embedding* tersebut.

2.6. Library Tambahan

OpenCV (*Open Source Computer Vision Library*) adalah salah satu *software* pustaka yang ditujukan untuk pengolahan citra dinamis secara *real-time* [19]. OpenCV dapat diterapkan dengan menggunakan bahasa Python untuk keperluan pemrosesan citra digital dari bermacam sumber seperti video, kamera, maupun protokol lain seperti *Real Time Streaming Protocol* (RTSP). Pada penelitian ini OpenCV digunakan untuk keperluan pengumpulan *dataset* foto, mengolah *frame* dari IP CCTV, dan untuk mendeteksi pola senyuman.

DLib merupakan sebuah *library machine learning* yang ditulis dalam bahasa pemrograman C++ [20]. Pada penelitian ini DLib digunakan untuk *cropping* atau

mengambil bagian wajah dari suatu gambar yang ditangkap oleh CCTV.

3. Hasil dan Pembahasan

3.1. Dataset Citra Wajah

Pada Gambar 6 ditunjukkan sampel foto salah satu mahasiswa dari total tujuh mahasiswa. Dalam penelitian yang dilakukan, dilakukan proses seleksi *dataset* yang memiliki variasi wajah yang cukup beragam dan memenuhi jumlah minimal foto sehingga didapatkan sejumlah tujuh mahasiswa. Masing-masing mahasiswa memiliki 100 foto. Dari 100 foto tersebut dibagi menjadi tiga bagian, yaitu 10 foto digunakan sebagai *data testing*, 75 foto sebagai *data training*, dan 15 foto untuk *data validation*.



Gambar 6. Contoh Data Foto Mahasiswa

Pada Tabel 4 ditunjukkan jumlah foto satu mahasiswa, dan keseluruhan tujuh mahasiswa untuk mempermudah melihat perhitungan yang dilakukan. Perhitungan ini berdasarkan pembagian yang telah ditentukan sebelumnya menyesuaikan dengan jumlah foto yang dimiliki. Jumlah keseluruhan foto yang digunakan untuk *training* sebanyak 525 foto, untuk *testing* sebanyak 70 foto, dan untuk *validation* sebanyak 105 foto.

Tabel 4. Pembagian Data

No	Nama pembagian	Jumlah Foto Satu Mahasiswa	Jumlah Foto Tujuh Mahasiswa
1	<i>Training</i>	75	525
2	<i>Testing</i>	10	70
3	<i>Validation</i>	15	105
Total		100	700

3.2. Training

Setelah *dataset* tersedia dan pembagian data ditentukan, selanjutnya dilakukan proses *training*. Pada Tabel 4 total foto yang digunakan untuk *training* sejumlah 525 foto. Sebelum dilakukan *training*, dilakukan *alignment* atau mengubah dimensi foto menjadi 160x160 dengan menggunakan *script* Python.

Selanjutnya dilakukan *training* dengan menggunakan *script* Python dengan spesifikasi seperti pada Tabel 2. Dalam prosesnya, untuk pertama kali akan di-load *Pre-trained* model dari FaceNet. Sejumlah foto tersebut menjadi *input* untuk diekstraksi fitur-fiturnya dan melewati beberapa layer-layer yang ada di arsitektur. Hasil akhir proses *training* ini akan dihasilkan sebuah model berekstensi *pickle* sebesar 649 KB.

3.3. Testing

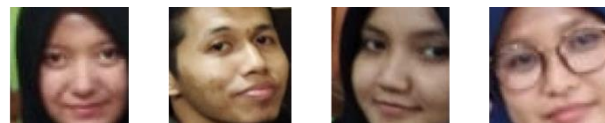
Dalam menguji suatu model *deep learning* perlu dilakukan *testing* model, dengan menggunakan *dataset*

sendiri selain dari *training* dan *validation*. Hasil *testing* ini berupa nilai akurasi. Akurasi dihitung berdasar berapa banyak model mampu mengklasifikasi dengan benar wajah yang diperoleh. Pada Tabel 4, total foto yang digunakan untuk *testing* sejumlah 70 foto. *Testing* ini mendapatkan nilai akurasi sebesar 92,6%.

Cara lain untuk melakukan *testing* adalah dengan menggunakan tangkapan layar dari IP CCTV. Sebagai eksperimen disini dilakukan pengambilan sampel sejumlah lima foto dari IP CCTV dengan situasi yang berbeda dan tiap foto minimal terdiri dari dua orang seperti ditunjukkan pada Gambar 7. Dimensi yang digunakan untuk *testing* adalah 160x160, maka foto tadi perlu dilakukan *crop* pada area wajah dengan menggunakan *library* OpenCV dan DLib. Beberapa *crop* wajah bisa dilihat seperti pada Gambar 8. Setiap *image* hasil *crop* selanjutnya dilakukan proses identifikasi. Hasil akurasi pengenalan wajah dari eksperimen ini hanya sebesar 66,7%. Hasil *testing live* lebih kecil dari hasil *testing* model dan selisihnya cukup jauh, kondisi seperti ini disebut *overfitting*. *Overfitting* adalah suatu keadaan model terlalu mengacu pada data pelatihan yang digunakan. Hasil yang *overfitting* dapat disebabkan oleh beberapa hal seperti *epoch* terlalu banyak, dan kurangnya variasi gambar data pelatihan [21].



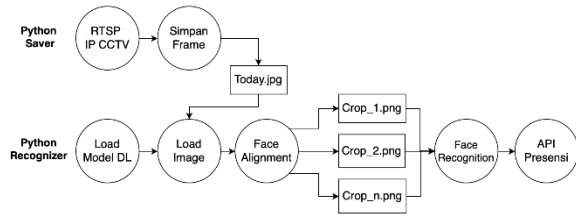
Gambar 7. Tangkapan Layar dari IP CCTV



Gambar 8. Face Detection

3.4. Implementasi Model

Dalam pengenalan wajah mahasiswa melalui IP CCTV ini, dibuat *pipeline script* Python yang mengakses CCTV menggunakan OpenCV dengan protocol RTSP. *Frame* diambil per detik. Tiap *frame* akan dilakukan pendeteksian wajah dengan konsep *face alignment* untuk selanjutnya dilakukan *face recognition*. Tahap *face recognition* membutuhkan model dari FaceNet dan model yang telah di-*training*. Gambar 9 menunjukkan bagaimana *pipeline* Python bekerja dalam mengimplementasikan model FaceNet.



Gambar 9. Implementasi Model di Pipeline Python

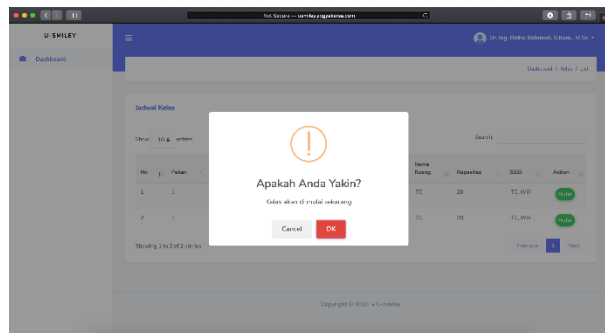
Script Python ini berjalan di mini komputer, terdiri dari dua *service* utama yaitu bagian penyimpanan *image* dan bagian pengenalan wajah. Blok bagian pertama mengakses IP CCTV melalui protokol RTSP, mendapatkan *frame image*, lalu disimpan menjadi sebuah *file image* berekstensi jpg dengan format penamaan *today.jpg*. Blok bagian kedua berjalan secara berkesinambungan memproses *file today.jpg*. Proses pertama yaitu *load model deep learning* terlebih dahulu, lalu *load image today.jpg*. *Image* yang di-load akan diproses untuk dideteksi ada berapa wajah. Setiap wajah yang dideteksi selanjutnya di-*crop* dan disimpan menjadi *file image* berekstensi jpg. Setiap *file crop* ini diolah oleh model *deep learning* yang telah di-load sebelumnya. Hasil dari identifikasi model adalah pengenalan berupa NIM dari wajah tersebut. NIM ini digunakan sebagai ID untuk presensi melalui *Applicaton Programming Interface* (API) presensi.

3.5. Aplikasi Web Dosen

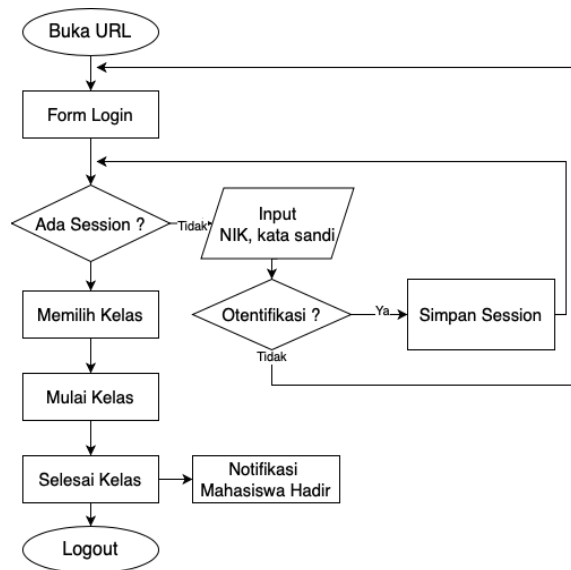
Pada saat proses kegiatan belajar mengajar, dosen berperan untuk membuka dan menutup kegiatan pembelajaran. Hal tersebut dilakukan melalui aplikasi web yang telah terintegrasi dengan sistem presensi yang ada. Pada aplikasi web ini disediakan fitur untuk melihat jadwal per hari. Dari tiap jadwal tersebut terdapat tombol Mulai dan Selesai. Tombol Mulai untuk menyatakan bahwa kelas telah dimulai dan presensi mulai dicatat seperti pada Gambar 10. Presensi masuk dilakukan secara otomatis di mini komputer yang mengakses IP CCTV. Selanjutnya dari proses pengenalan wajah tangkapan CCTV tadi didapat Nomor Induk Mahasiswa (NIM), jika sesuai dengan *database* mahasiswa yang terdaftar maka data dimasukkan ke tabel presensi. Saat pembelajaran telah selesai, dosen menekan tombol Selesai untuk menyatakan bahwa kelas boleh ditinggalkan. Alur proses bagaimana aplikasi web ini bekerja ditunjukkan pada Gambar 11.

Aplikasi web diakses melalui *Uniform Resource Locator* (URL), dan mewajibkan login terlebih dahulu menggunakan Nomor Induk Karyawan (NIK) dan kata sandi. Apabila otentifikasi berhasil, menu dapat digunakan, terutama menu Daftar Kelas. Daftar Kelas menampilkan jadwal sesuai tanggal akses. Dosen memilih kelas yang dipilih dan menekan tombol Mulai Kelas untuk memulai perkuliahan. Apabila kelas telah usai, tombol Selesai Kelas ditekan, secara otomatis

memberikan *trigger* berupa notifikasi ke *smartphone* mahasiswa untuk melakukan konfirmasi.



Gambar 10. Konfirmasi Kelas Dimulai oleh Dosen



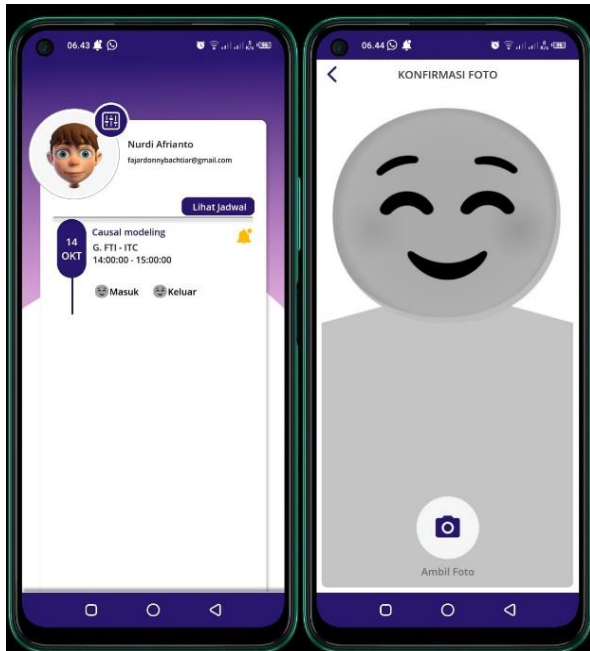
Gambar 11. Flowchart Aplikasi Web

3.6. Aplikasi Mobile Mahasiswa

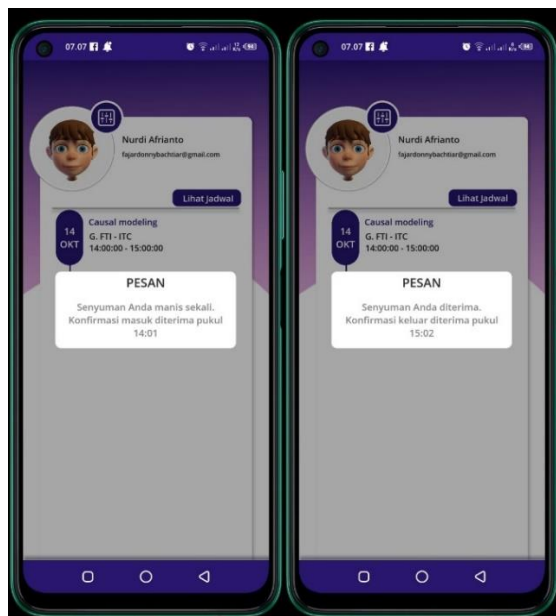
Penggunaan aplikasi *mobile* bagi mahasiswa turut mendukung jalannya sistem presensi ini. Melalui aplikasi *mobile*, mahasiswa akan mendapatkan notifikasi berupa pesan telah tercatat masuk di kelas maupun nanti saat kelas sudah selesai. Mahasiswa diwajibkan untuk melakukan konfirmasi dengan melakukan *selfie* dengan wajah tersenyum untuk presensi masuk dan presensi keluar. Dua senyuman ini dihitung sebagai satu record presensi untuk setiap mahasiswa. Ilustrasi ditunjukkan pada Gambar 12 dan Gambar 13. Sedangkan alur proses bagaimana aplikasi mobile ini bekerja ditunjukkan pada Gambar 14.

Aplikasi mobile mewajibkan *login* terlebih dahulu menggunakan Nomor Induk Mahasiswa (NIM) dan email. *One-Time Password* (OTP) dikirim melalui email untuk dicocokkan. Setelah otentifikasi berhasil maka tampilan yang pertama muncul adalah Daftar Jadwal. Ada *background service* di *mobile* yang memunculkan notifikasi apabila mahasiswa yang bersangkutan diidentifikasi memasuki kelas, dan dosen menyatakan kelas selesai. Apabila CCTV berhasil mengidentifikasi

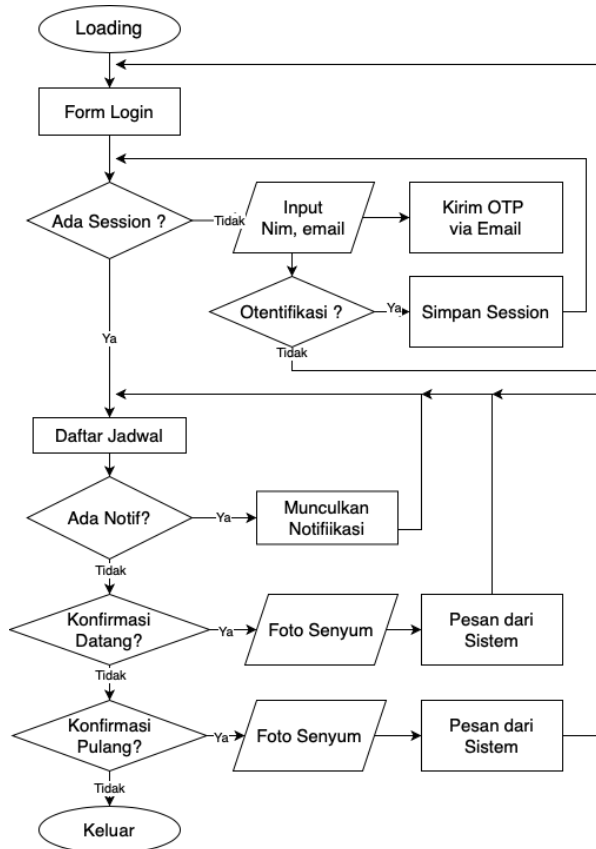
mahasiswa saat hadir di kelas, sebuah ikon lonceng dimunculkan. Mahasiswa melakukan konfirmasi dengan *selfie* tersenyum. Apabila dosen menyatakan kelas telah selesai, ikon lonceng muncul juga di daftar jadwal yang sesuai. Mahasiswa sekali lagi melakukan konfirmasi dengan *selfie* tersenyum. Pengenalan pola senyuman ini menggunakan API Python dan *library* OpenCV yang mengenali senyuman dari *selfie image* yang dikirimkan.



Gambar 12. Notifikasi dan Konfirmasi Presensi di *Mobile App*



Gambar 13. Pesan Balasan dari Sistem Setelah Konfirmasi di *Mobile App*



Gambar 14. Flowchart Aplikasi *Mobile* Mahasiswa

4. Kesimpulan

Dari penelitian yang dilakukan didapatkan hasil terbentuknya sistem presensi terintegrasi web, *mobile*, dan *service* yang menerapkan arsitektur *deep learning* berbasis CNN yaitu FaceNet. FaceNet digunakan sebagai *feature extractor* yang dikombinasikan dengan SVM sebagai *classifier*. Selain itu, validasi diperkuat dengan kondisi pengecekan lokasi berdasarkan SSID yang dicocokkan *database*, dan waktu berdasarkan kapan waktu pengidentifikasian wajah di *server*. Sebagai konfirmasi, mahasiswa juga diwajibkan *selfie* tersenyum sebagai penguat keabsahan presensi. Sistem terintegrasi ini siap diujicoba secara nyata, tetapi mengingat masih masa pandemi Covid-19 saat ini masih dalam bentuk purwarupa.

Hasil pengujian dari model *deep learning* dilakukan dengan dua cara yaitu *testing* secara model dan *testing* secara *live* dengan sampel. Hasil akurasi dari *testing* model sebesar 92,6% sedangkan untuk *testing live* akurasi hanya sebesar 66,7%. Nilai *testing live* lebih kecil dan cukup jauh dari *testing* model dapat diartikan hasil *training* model terlalu *overfitting*, disebabkan kurang beragamnya citra wajah yang dikumpulkan tiap mahasiswa pada penelitian ini. Untuk studi selanjutnya kami menyarankan untuk lebih beragamnya pengumpulan *dataset* citra wajah tiap mahasiswa sebagai salah satu cara meningkatkan akurasi.

Ucapan Terimakasih

Penelitian ini didukung oleh Hibah Penelitian Tesis Magister dari KEMENRISTEKDIKTI dengan Nomor 081/SP2H/AMD/LT/DRPM/2020.

Daftar Rujukan

- [1] A. Fakhri, I. K. Raharjana, and B. Zaman, "Pemanfaatan Teknologi Fingerprint Authentication untuk Otomatisasi Presensi Perkuliahan," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 1, no. 2, p. 41, Nov. 2015, doi: 10.20473/jisebi.1.2.41-48.
- [2] T. Matsumoto and H. Matsumoto, "Impact of artificial gummy fingers on fingerprint systems," *Proc.*, 2002.
- [3] Samirso, "ZKTeco - How to Enter the Device Without Admin Affirming." <https://www.instructables.com/id/ZKTeco-How-to-Enter-the-Device-Without-Admin-Affir/> (accessed Aug. 17, 2019).
- [4] William and A. P. U. Sembiring, "Implementasi Identifikasi Sidik Jari Pada Sistem Informasi Penjualan," *Semin. Nas. Sist. Inf. Indones.*, vol. 2013, p. 351, 2013.
- [5] U. Syafitri, "Efektivitas Penerapan Absensi Finger Print Pada Lembaga Pendidikan Perkebunan Medan," *Univ. Medan Area*, pp. 23–24, 2018.
- [6] C. Suhery and I. Ruslianto, "Identifikasi Wajah Manusia untuk Sistem Monitoring Kehadiran Perkuliahan menggunakan Ekstraksi Fitur Principal Component Analysis (PCA)," *J. Edukasi dan Penelit. Inform.*, vol. 3, no. 1, p. 9, Apr. 2017, doi: 10.26418/jp.v3i1.19792.
- [7] Z. Fachmi, M. Sudarma, and L. Jasa, "Sistem Monitoring Kehadiran Perkuliahan Menggunakan Face Detection Dengan Algoritma Viola Jones," *Maj. Ilm. Teknol. Elektro*, 2019, doi: 10.24843/mite.2019.v18i01.p18.
- [8] D. Derisma, "Sistem Pengenalan Wajah Secara Realtime Berbasis Android Menggunakan Metode Eigenface Pada OpenCV," *J. Komput. Terap.*, vol. 2, no. 2, pp. 127–136, 2016.
- [9] A. Pradana and E. Paulus, "Aplikasi Deteksi Wajah pada Sekumpulan Orang dengan Membandingkan Metode Viola-Jones dan KLT," in *SENAPATI 2016*, 2016, pp. 230–231.
- [10] N. T. Son *et al.*, "Implementing CCTV-based attendance taking support system using deep face recognition: A case study at FPT polytechnic college," *Symmetry (Basel)*, 2020, doi: 10.3390/sym12020307.
- [11] M. A. Prasanty and F. Utamingrum, "Sistem Presensi Mahasiswa Berdasarkan Pengenalan Wajah Menggunakan Metode LBP dan K-Nearest Neighbor Berbasis Mini PC," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. Vol. 4, no. April 2020, pp. 1168–1171, 2020.
- [12] G. Q. O. Pratamasunu, O. I. R. Farisi, and M. Jannah, "Pengenalan Wajah Mahasiswa Universitas Nurul Jadid Pada Video Menggunakan Metode Haar Cascade Dan Deep Learning," *Core-IT J. Komputasi dan Teknol. Inf.*, vol. Vol. 1, no. November 2020, pp. 25–34, 2020.
- [13] T. Stat, "Face Recognition Using," vol. 4, no. 2. pp. 2–5, 2003, [Online]. Available: <https://github.com/davidsandberg/facenet>.
- [14] F. Hu, G. S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sens.*, 2015, doi: 10.3390/rs71114680.
- [15] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Fully convolutional neural networks for remote sensing image classification," 2016, doi: 10.1109/IGARSS.2016.7730322.
- [16] J. Yosinki, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," 2014.
- [17] JavaTpoint, "Convolutional Neural Network In PyTorch." <https://www.javatpoint.com/pytorch-convolutional-neural-network>.
- [18] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," 2015, doi: 10.1109/CVPR.2015.7298682.
- [19] A. Zein, "Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OpenCV dan DLIB Python," *Sainstech*, 2018.
- [20] N. Wahyudiana and S. Budi, "Perbandingan Performa Pre-Trained Classifier dLib dan HAAR Cascade (OpenCV) Untuk Mendeteksi Wajah," *J. Strateg.*, vol. 1, p. 376, 2019.
- [21] N. Ramadhani, J. Hendryli, and D. E. Herwindianti, "Pencarian Objek Wisata Bersejarah Di Pulau Jawa Menggunakan Convolutional Neural Network," *J. Ilmu Komput. dan Sist. Inf.*, vol. 7, 2019.