



Metode *Face Recognition* untuk Identifikasi Personil Berdasar Citra Wajah bagi Kebutuhan Presensi *Online* Universitas Negeri Semarang

Luthfi Maslichul Kurniawan

BPTIK Universitas Negeri Semarang

Email: emka@mail.unnes.ac.id

Abstrak

Salah satu hasil evaluasi dari Sistem Presensi *Online* Pegawai Universitas Negeri Semarang adalah adanya proses-proses curang di dalam sistem presensi dengan menitipkan presensi atau melakukan foto presensi kosong. Sistem Presensi *Online* Pegawai yang dikembangkan dengan bahasa pemrograman *PHP*, basis data *MySQL* dan pustaka *JavaScript JPEGCam* itu tidak cukup mampu memberikan gambaran tentang kedisiplinan dari pegawai di lingkungan Universitas Negeri Semarang. Untuk itulah, perlu dibangun sebuah sistem yang mampu mengolah data-data foto yang ditangkap dari proses presensi *online* tersebut untuk dianalisis apakah ada wajah manusia yang terdeteksi. Maka, dikembangkanlah sebuah sistem *face recognition* yang dirancang menggunakan bahasa pemrograman *Python* dan pustaka *OpenCV*. Hasil dari rancang bangun ini adalah sistem *face recognition* yang mampu berjalan secara otomatis di komputer *server* untuk membaca basis data presensi, mengolah foto-foto yang tersimpan pada basis data tersebut, mendeteksi wajah pada foto-foto yang diolah kemudian menampilkan hasilnya pada tabel basis data presensi untuk diolah dalam bentuk skor deteksi wajah yang tampil di rekapitulasi presensi *online* pegawai.

Kata Kunci: *Face recognition, Image processing, OpenCV, Python, Sistem informasi kepegawaian*

1. PENDAHULUAN

Sebagai salah satu Perguruan Tinggi yang masuk dalam kategori Badan Layanan Umum (BLU), Universitas Negeri Semarang dituntut untuk selalu melakukan pelayanan publik yang memuaskan. Hal itu bisa dilakukan bila sumber daya manusia, dalam hal ini seluruh tenaga pendidik (dosen) dan tenaga kependidikan Universitas Negeri Semarang yang profesional, diantaranya memiliki pengetahuan teknis dan ketrampilan yang baik, mempunyai loyalitas dan tanggungjawab yang besar dan memiliki kedisiplinan yang tinggi [1].

Untuk membentuk karakter sumber daya manusia yang memiliki profesionalitas dan kedisiplinan tinggi, Universitas Negeri Semarang sejak tahun 2009 telah mengembangkan sistem presensi pegawai secara *online* yang bisa diakses pada jaringan *intranet* kampus. Sistem presensi pegawai yang dikembangkan berbasis *web* ini memungkinkan setiap pegawai Universitas Negeri Semarang untuk melakukan presensi di loket-loket presensi yang disediakan di masing-masing unitnya.

Metode Face Recognition untuk Identifikasi Personil Berdasar Citra Wajah bagi Kebutuhan Presensi Online Universitas Negeri Semarang

Sistem Presensi *Online* yang dikembangkan sebagai aplikasi berbasis *web* itu ditulis dalam bahasa pemrograman *PHP* dan memanfaatkan basis data berbasis *MySQL*. Selain itu, sistem ini memberdayakan pustaka berbasis bahasa pemrograman *JavaScript* yaitu *JPEGCam* untuk menangkap foto dari *Webcam* yang terpasang pada komputer *client* dan mengirimkannya ke komputer *server* untuk diproses.

Proses presensi pegawai dapat dilakukan dengan sangat mudah, cukup dengan menyesuaikan posisi diri di depan kamera pada loket yang disediakan, kemudian mengetikkan kode identitas pegawai, baik itu NIP maupun NRP dan menekan tombol Enter. Berikut tampilan Presensi *Online* Pegawai seperti ditunjukkan pada Gambar 1.



Gambar 1. Tampilan Presensi Online Pegawai Universitas Negeri Semarang

Kemudahan proses presensi ini ditunjang pula dengan sistem rekapitulasi yang lengkap dan terintegrasi di Sistem Informasi Kepegawaian Universitas Negeri Semarang (Simpeg) seperti ditunjukkan pada Gambar 2 berikut.

Q Rekap Kehadiran

No./Nama/NIP	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
1.	M	P	M	P	M	P	M	P	M	P	M	P	M	P	M	P	M	P	M	P	M	P	M	P	M	P	M	P	M	P	M	P
2.	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
3.	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
4.	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
5.	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
6.	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
7.	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
8.	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
9.	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
10.	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V

Gambar 2. Tampilan rekap presensi pegawai di Sistem Informasi Kepegawaian

Meskipun telah menerapkan sistem presensi pegawai secara *online*, Universitas

Negeri Semarang tetap memberlakukan proses pembinaan kedisiplinan pegawai dengan tetap memberikan wewenang kepada para pejabat di masing-masing unit untuk melakukan pengendalian dan pengawasan presensi pegawai di lingkungannya masing-masing. Proses pengendalian ini dapat dilakukan oleh para pejabat pada modul rekap presensi yang telah disediakan pada Sistem Informasi Kepegawaian (Simpeg).

Dari serangkaian evaluasi yang dilakukan oleh Biro Administrasi Umum dan Kepegawaian Universitas Negeri Semarang, didapati sejumlah temuan bahwa masih banyak para pegawai yang tidak melakukan presensi sebagaimana mestinya. Beberapa kasus klasik seperti titip absen masih kerap ditemui. Untuk itulah perlu dikembangkan lagi satu mekanisme yang dapat mengurangi kasus-kasus seperti ini dikemudian hari.

Salah satu mekanisme yang dapat diterapkan dalam mengatasi kasus diatas adalah melakukan penguatan kontrol dan pengawasan presensi pegawai yang selama ini dilakukan secara manual oleh para pejabat masing-masing unit. Penguatan yang dimaksud adalah dengan memberikan rekapitulasi yang lebih lengkap dari data-data foto presensi pegawai yang telah tersimpan pada *database*.

Teknologi yang dapat diterapkan pada mekanisme diatas adalah teknologi *face recognition* atau teknologi pendektasian wajah. Dengan teknologi ini, sistem akan melakukan analisa-analisa citra dari foto-foto presensi yang telah tersimpan, apakah benar-benar ada 'wajah manusia' yang ada foto-foto yang dimaksud. Jika ada, data-data ini dapat disajikan dalam rekapitulasi presensi pegawai dalam bentuk statistik hasil analisis. Para pejabat yang melakukan proses kendali dan pengawasan presensi pegawai-pegawainya akan secara mudah melihat siapa saja yang memiliki skor analisis *face recognition* yang rendah, kemudian melakukan langkah-langkah yang perlu dilakukan untuk pegawai tersebut.

Teknologi *face recognition* saat ini sudah sangat canggih dan diterapkan di hampir semua sistem mutakhir yang mengelola citra dan foto, contohnya situs-situs media sosial seperti *Facebook* dan *Google Plus* yang mampu mengenali wajah manusia dan bahkan memprediksi nama dari pemilik wajah manusia tersebut. Teknologi canggih itu dikembangkan dari beberapa pustaka-pustaka yang telah beredar secara umum. Contohnya *OpenCV (Open-source Computer Vision)*.

Menurut [2], *OpenCV* atau *Open-source Computer Vision* adalah pustaka pemrograman yang digunakan untuk pemrosesan citra (*image processing*) secara *real-time*. Pustaka ini pertama kali dikembangkan di Intel Corporation tahun 2000 yang lalu sampai sekarang. Pustaka yang ditulis dalam bahasa pemrograman C++ ini telah tersedia dalam hampir semua bahasa pemrograman besar, seperti *Python*, *Java* dan *MATLAB*.

OpenCV menggunakan algoritma *Haar* untuk melakukan analisis citra. Algoritma *Haar* sendiri merupakan salah satu dari sekian banyak algoritma matematis untuk menghitung sebuah bentuk dari suatu bagian citra yang diproses. Algoritma *Haar* sendiri berkembang dari teknologi *face recognition* pertama yang disebut *Haar*

wavelets yang hanya mendeteksi lengkungan-lengkungan badan dan muka, menjadi *Haar-like features* yang mampu mendeteksi bagian-bagian dari citra yang menyerupai bagian dari muka, misalnya bibir, mata, hidung, dll. Data-data *Haar-like features* sendiri saat ini terus dikembangkan oleh sejumlah peneliti dan dapat diunduh dan dipakai secara bebas [3-4].

Bahasa pemrograman yang paling matang dalam menyediakan pustaka *OpenCV* diatas adalah bahasa pemrograman *Python*. *Python* adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Bahasa pemrograman ini dipakai sebagai *backend* di banyak sistem informasi skala besar karena kemampuannya yang dapat berjalan cepat dan berjalan pada *multiplatform* sistem operasi [5-6].

Dengan membangun sistem *face recognition* pada Presensi *Online* Pegawai Universitas Negeri Semarang, diharapkan proses pengendalian dan pengawasan presensi pegawai secara *online* dapat dilakukan lebih mudah dan akurat. Data-data yang dihasilkan dari sistem ini kemudian dapat digunakan untuk mengembangkan sistem informasi kepegawaian yang lebih canggih lagi, sehingga proses pengelolaan sumber daya manusia pada Universitas Negeri Semarang dapat lebih efektif dan efisien.

2. METODOLOGI

Perancangan sistem *face recognition* pada Presensi *Online* Pegawai Universitas Negeri Semarang ini dilakukan dalam beberapa tahap yang secara umum mengikuti tahapan-tahapan siklus *SDLC* (*software development life cycle*) dari mulai proses pengambilan data awal, penganalisan kebutuhan sistem, pembuatan desain sistem, kemudian pembuatan program dari sistem itu sendiri. Hasil dari rancang bangun dari sistem *face recognition* ini akan dijelaskan dalam deskripsi yang mewakili tiap bagiannya.

Keberhasilan dari sistem ini akan diuji dengan tahap pengujian yang dilakukan dalam dua lingkungan yang berbeda, Lingkungan Pengembangan (*development environment*) dan Lingkungan Produksi (*production environment*). Sistem akan diuji dulu dalam lingkungan pengembangan sampai dirasa siap untuk kemudian diuji dan digunakan dalam lingkungan produksi Sistem Informasi Kepegawaian Universitas Negeri Semarang.

3. HASIL DAN PEMBAHASAN

3.1. Sistem Face Recognition Presensi Online Pegawai

Sistem *face recognition* Presensi *Online* Pegawai berfungsi untuk melakukan deteksi wajah manusia pada citra foto presensi yang dilakukan secara *online* oleh pegawai

Universitas Negeri Semarang. Dengan pengolahan citra yang dimaksud, hasil deteksi akan disimpan dalam basis data presensi pegawai dan basis data tampungan yang dapat digunakan dikemudian hari untuk membuat algoritma yang lebih cerdas. Sistem *face recognition* akan memberikan keputusan apakah sebuah citra foto benar-benar terdapat wajah manusia atau tidak. Sistem juga akan memberikan koordinat dari wajah yang terdeteksi dan menyimpannya untuk digunakan dikemudian hari.

3.1.1. Pembacaan Data Presensi Pegawai

Sistem *face recognition* ini sepenuhnya dikembangkan dalam bahasa *Python*. Diperlukan pustaka *Python-MySQL* seperti ditunjukkan pada Gambar 3 untuk dapat membaca dan mengolah data yang tersimpan dalam basis data presensi yang menggunakan server *MySQL*.

```
import cv2
import sys
import MySQLdb as mydb
import os

# koneksi database
koneksi = mydb.connect('localhost', namauser, '',
tabel'presensi');
tabelPresensi = "foto_pr_" + bln + "_" + thn

with koneksi:

    # konek, select data identitas, tanggal dan nama file foto
    query = koneksi.cursor()
    query.execute("SELECT identitas, tgl, name_file FROM " +
tabelPresensi + " WHERE sesi = '1' AND tgl = '" + tgl + "'
ORDER BY tgl_isi DESC")

    for i in range(query.rowcount):

        row = query.fetchone()

        #definisikan variabel umum
        r_identitas = row[0]
        r_tgl       = str(row[1])
        r_file      = row[2]
        .....
```

Gambar 3. Pustaka *Python-MySQL*

Bagian dari program diatas kemudian dilanjutkan dengan proses pembacaan citra foto dan *face recognition* dengan menggunakan *OpenCV*.

3.1.2. Face Recognition

Algoritma proses *face recognition* dapat dilakukan dalam beberapa tahap sebagai berikut. (a) Proses pembacaan *file* foto, (b) Proses konversi *file* foto ke dalam format *Grayscale*, (c) Penyiapan data wajah dalam *array* tertentu, (d) *Looping* untuk tiap wajah yang dikenali dalam foto yang diolah, (e) Membingkai wajah yang dideteksi dengan kotak dan menyimpan ke dalam *file* baru untuk penelitian lebih lanjut dan (f) Melakukan *update* data presensi yang diolah tersebut, apakah mendeteksi wajah atau tidak.

Adapun sintaks dari bagian sistem yang berkaitan dengan *face recognition* seperti ditunjukkan pada Gambar 4 berikut.

```
# path file foto
    imagePath = pathFoto + r_tgl + "/" + r_file

# jika file foto exist,
if os.path.exists(imagePath):

    # baca file foto tersebut
    image = cv2.imread(imagePath)

    # ubah foto ke modus Grayscale
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # dapatkan data wajah dengan algoritma Haar, taruh
ke array faces
    faces =
faceCascade.detectMultiScale(gray, scaleFactor=1.3778, minN
ighbors=5, minSize=(24,
                    24), flags =
cv2.cv.CV_HAAR_SCALE_IMAGE)

    # jika, ada wajah yang terdeteksi
    if len(faces) > 0:

        # hanya looping array
        image_faces = []
        for (x, y, w, h) in faces:
            image_faces.append(image[y:(y+h),
x:(x+w)])

        # proses setiap wajah yang terdeteksi pada satu
file foto
        for i, face in enumerate(image_faces):

            # path file untuk bingkai wajah yang
terdeteksi
            filename = pathPatern + str(r_identitas) +
```

```
"/" + str(i) + ".jpg"

        # deteksi folder untuk identitas tertentu,
        if not
os.path.exists(os.path.dirname(filename)):
        # buat folder khusus identitas tersebut
jika belum ada

        os.makedirs(os.path.dirname(filename))

        # tulis file bingkai wajah yang terdeteksi
        if not os.path.exists( filename ):
            cv2.imwrite( filename, face)
        else:

            # jika sudah ada file bingkai wajah,
            ganti dg nama lain y + 1
            y = i + 1
            new_filename = pathPatern +
str(r_identitas) + "/" + str(y) + ".jpg"
            cv2.imwrite( new_filename, face)

            # insert pattern ke DB
            current = koneksi.cursor()
            current.execute("INSERT INTO pattern
VALUES (NULL, %s, %s)", (r_identitas, face, ))

            # update data Siremun
            coordinate = str(x) + "," + str(y) + "," + str(w)
+ "," + str(h)
            with koneksiSiremun:
                cur = koneksiSiremun.cursor()
                cur.execute("UPDATE " + tabelSiremun + "
SET face_detected = '1', face_coord = %s WHERE nip = %s AND
tanggal = %s",
                ( coordinate, r_identitas,
tanggalFix ) )

            print r_identitas, r_file, str(x) + "," +
str(y) + "," + str(w) + "," + str(h)
            else:
                with koneksiSiremun:
                    cur = koneksiSiremun.cursor()
                    cur.execute("UPDATE " + tabelSiremun + "
SET face_detected = '0' WHERE nip = %s AND tanggal = %s",
                    ( r_identitas, tanggalFix ) )
                    print r_identitas, r_file, "tidak ketemu
wajah"
```

Gambar 4. Sintaks sistem yang berkaitan dengan *face recognition*

3.1.3. Eksekusi Sistem Secara Otomatis

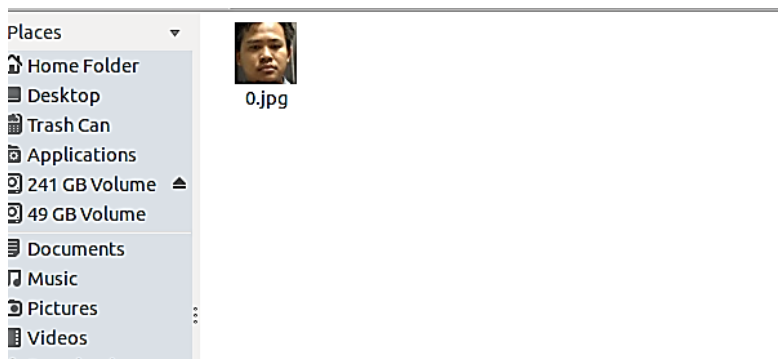
Sistem ini dirancang sebagai aplikasi *backend* yang berjalan secara otomatis di sistem operasi *Linux* pada komputer *server* yang terkoneksi dengan basis data presensi pegawai. Proses eksekusi otomatis dilakukan dengan bantuan *Linux Cron*. Eksekusi aplikasi dijalankan pada jam 9 pagi, atau 2 jam setelah batas waktu presensi pegawai dianggap telat dan pada pukul 23 malam dimana seluruh pegawai sudah diasumsikan selesai bekerja pada hari tersebut. Berikut adalah contoh tampilan hasil eksekusi dari sistem diatas ditunjukkan oleh Gambar 5.

```
131803128 1_131803128_1389568065.jpeg tidak ketemu wajah
131568902 1_131568902_1389567987.jpeg tidak ketemu wajah
196803142008101001 1_196803142008101001_1389567905.jpeg tidak ketemu wajah
82012710091243 1_82012710091243_1389567852.jpeg tidak ketemu wajah
132297708 1_132297708_1389567835.jpeg tidak ketemu wajah
131813667 1_131813667_1389567805.jpeg tidak ketemu wajah
132320172 1_132320172_1389567785.jpeg tidak ketemu wajah
70662208011076 1_70662208011076_1389567731.jpeg tidak ketemu wajah
130604216 1_130604216_1389567705.jpeg tidak ketemu wajah
132158718 1_132158718_1389567698.jpeg tidak ketemu wajah
130812917 1_130812917_1389567657.jpeg 43,126,63,63
132320167 1_132320167_1389567640.jpeg tidak ketemu wajah
131474079 1_131474079_1389567626.jpeg tidak ketemu wajah
131658885 1_131658885_1389567605.jpeg tidak ketemu wajah
132321250 1_132321250_1389567529.jpeg tidak ketemu wajah
81091508021096 1_81091508021096_1389567455.jpeg 58,8,86,86
197911102009101001 1_197911102009101001_1389567453.jpeg tidak ketemu wajah
198708292011031026 1_198708292011031026_1389567360.jpeg tidak ketemu wajah
65070608011092 1_65070608011092_1389567305.jpeg 39,66,63,63
197505042008101003 1_197505042008101003_1389567117.jpeg tidak ketemu wajah
80040808021127 1_80040808021127_1389567095.jpeg tidak ketemu wajah
80031908021120 1_80031908021120_1389567083.jpeg tidak ketemu wajah
83052408021115 1_83052408021115_1389567073.jpeg 41,80,72,72
87022408021119 1_87022408021119_1389567030.jpeg tidak ketemu wajah
```

Gambar 5. Tampilan eksekusi sistem dalam terminal.

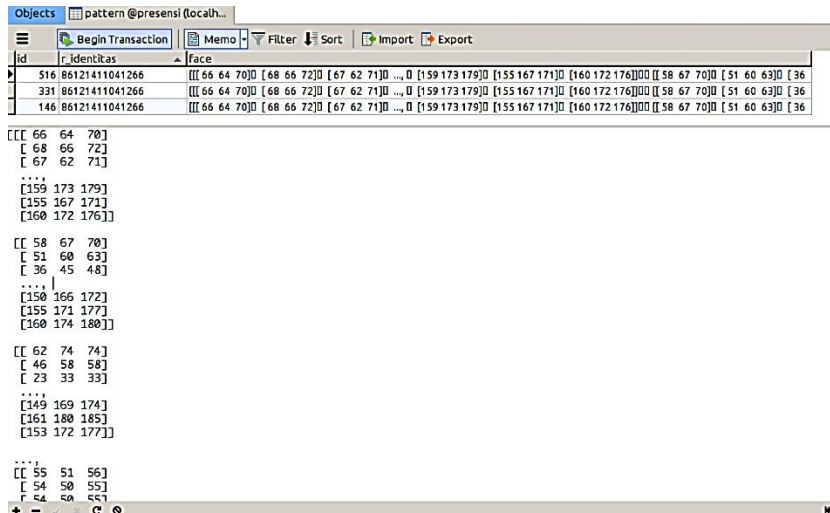
3.1.4. Hasil Deteksi

Hasil deteksi proses *face recognition* selain disimpan dalam basis data presensi pegawai juga dicadangkan dalam *file* baru. *File* baru tersebut memuat citra wajah yang terdeteksi dan disimpan dalam format *file* *JPG*. *File-file* ini akan dipersiapkan untuk algoritma *face detection* yang lebih canggih lagi, dimana sistem akan mampu mendeteksi apakah benar foto si Pegawai A yang tersimpan dalam basis data adalah benar-benar Pegawai A. Contoh hasil dari *file* hasil deteksi sistem ini seperti terlihat pada Gambar 6.



Gambar 6. Tampilan file patern wajah hasil deteksi sistem

Tampilan data *patern* wajah yang disimpan ke dalam basis data seperti ditunjukkan pada Gambar 7.



id	r_identitas	face
516	86121411041266	[[[66 64 70] [68 66 72] [67 62 71] ... 0 [159 173 179] [155 167 171] [160 172 176]] [58 67 70] [51 60 63] [36
331	86121411041266	[[[66 64 70] [68 66 72] [67 62 71] ... 0 [159 173 179] [155 167 171] [160 172 176]] [58 67 70] [51 60 63] [36
146	86121411041266	[[[66 64 70] [68 66 72] [67 62 71] ... 0 [159 173 179] [155 167 171] [160 172 176]] [58 67 70] [51 60 63] [36

```
[[[ 66 64 70]
 [ 68 66 72]
 [ 67 62 71]
 ...
 [ 159 173 179]
 [ 155 167 171]
 [ 160 172 176]]
 [ 58 67 70]
 [ 51 60 63]
 [ 36 45 48]
 ...
 [ 150 166 172]
 [ 155 171 177]
 [ 160 174 180]]
 [ 62 74 74]
 [ 46 58 58]
 [ 23 33 33]
 ...
 [ 149 169 174]
 [ 161 180 185]
 [ 153 172 177]]
 ...
 [ 55 51 56]
 [ 54 58 55]
 [ 54 58 55]]
```

Gambar 5. Tampilan data *patern* wajah yang terdeteksi dan disimpan dalam basis data

Program ini tidak dikembangkan untuk lingkup deteksi canggih seperti itu, namun proses penyiapan data untuk pendeteksian yang lebih canggih sudah disiapkan.

3.2. Pembahasan

Sistem *face recognition* Presensi *Online* Pegawai Universitas Negeri Semarang ini dikembangkan dengan bahasa pemrograman *Python*. Basis data yang digunakan dan diolah pada sistem ini adalah basis data *MySQL*. Sistem ini menggunakan pustaka *python-opencv* yang menyajikan pustaka *OpenCV* (*Open-source Computer Vision*) di bahasa pemrograman *Python*.

Selain itu, untuk keperluan data pembanding deteksi wajah, sistem ini menggunakan data wajah-wajah yang telah dikoleksi dalam bentuk *Haar-like Features* pada sebuah *file XML*. Data koordinat-koordinat bentuk-bentuk bagian wajah tersebut kemudian dibaca oleh pustaka *OpenCV* dan dijadikan pembanding untuk melakukan proses penghitungan *face recognition*.

Dalam proses penggunaan algoritma *Haar-like Features* ini, harus dilakukan uji coba berkali-kali untuk mendapatkan formula *Scale Factor* atau faktor pengali yang cocok. Proses uji coba dapat dilakukan dengan memodifikasi *scale factor* dalam sintaks program kemudian menjalankan program tersebut dan menganalisis hasil yang didapati.

Dari sejumlah proses percobaan yang dilakukan, *Scale Factor* dari algoritma *Haar-like Feature* yang digunakan oleh *OpenCV* pada sistem ini cocok pada angka

1,3778. Angka yang terlalu tinggi dapat menyebabkan proses *face recognition* menjadi tidak akurat karena terlalu peka. Sementara angka yang terlalu rendah akan menghasilkan proses *face recognition* yang sia-sia karena semua bentuk wajah tidak dapat dikenali.

Adapun keunggulan-keunggulan dari sistem *face recognition* ini sebagai berikut (a) Mampu dijalankan secara otomatis untuk membaca dan menganalisis presensi pegawai. (b) Mampu mendeteksi bentuk wajah dari foto presensi yang tersimpan dalam basis data, (c) Melakukan pencadangan hasil deteksi bentuk wajah yang ditemukan untuk digunakan penelitian lanjutan dan (d) Mampu menuliskan hasil deteksi yang dilakukan baik secara *verbose* saat dijalankan maupun dituliskan dalam nilai *true-false* yang dimasukkan dalam basis data presensi yang diolah.

Kendati memiliki sejumlah keunggulan diatas, sistem ini memiliki beberapa kekurangan yang masih dapat diperbaiki dan dikembangkan pada penelitian lanjutan yang sangat memungkinkan dilakukan. Adapun kekurangan-kekurangan itu sebagai berikut (a) Ketidakmampuan sistem untuk memberikan deteksi yang akurat dari foto kosong tanpa wajah yang menghasilkan *false positive* karena adanya pendar sinar lampu yang tertangkap sehingga seolah-olah berupa wajah, (b) Ketidakmampuan sistem untuk memberikan deteksi yang akurat dari foto wajah yang tidak sempurna, dalam hal ini sedang memakai aksesoris kepala, atau adanya objek yang menghalangi sehingga bentuk wajah menjadi tidak sempurna dan (c) Kemampuan mendeteksi apakah benar foto pegawai A yang diproses benar-benar didapati foto si A.

4. SIMPULAN

Rancang bangun sistem *face recognition* Presensi *Online* Pegawai Universitas Negeri Semarang dengan *Python* dan *OpenCV* dapat terealisasi. Proses rancang bangun dari sistem ini dikembangkan menurut siklus pengembangan perangkat lunak *SDLC* (*Software Development Life Cycle*) yang meliputi tahap pengambilan data awal, penganalisan kebutuhan sistem, pembuatan desain sistem, kemudian pembuatan program dari sistem itu sendiri serta proses pengujian dalam lingkungan pengembangan (*development environment*) dan lingkungan produksi (*production environment*). Sistem ini memberdayakan data wajah yang telah dihitung menggunakan algoritma *Haar-like Features* dan dihitung dalam faktor pengali (*scale factor*) 1,3778. Sistem *face recognition* Presensi *Online* Pegawai ini telah mampu melengkapi dan mempermudah proses pengendalian dan pengawasan presensi pegawai oleh pejabat masing-masing unit karena menambahkan skor prediksi *face recognition* pada rekapitulasi presensi pegawai masing-masing unit. Dengan adanya skor yang dihitung dari tiap foto presensi harian yang dilakukan pegawai, pejabat dapat memberikan kebijakan tersendiri terkait pegawai tersebut maupun terkait infrastruktur sistem presensi yang ada di unitnya masing-masing jika ditemui banyak hasil *false-positive*.

5. REFERENSI

- [1] Abdurrohkman. 2014. *Kiat Menjadi Staf Profesional*. <http://www.banyumaskab.go.id/read/8098/kiat-menjadi-staf-profesional>, diakses 20 Mei 2015.
- [2] Brahmhatt, Samarth. 2013. *Practical OpenCV*. Appres, New York.
- [3] M. Hassaballah, Kenji Murakami, Shun Ido. 2011. Face Detection evaluation: a new approach based on the golden ratio. *Springer-Verlag London 2011*. Published 8 July.
- [4] Wu Lifang, Shen Lansun. 2003. Face Recognition from Front-view Face. *Journal of Electronics*. Vol. 20(1): 45-50.
- [5] Hetland, Magnus Lie. 2006. *Beginning Python: from Novice to Professional*. Apress, New York.
- [6] Python Software Foundation. 2008. "What is Python Good For?". General Python FAQ. <http://docs.python.org/faq/general.html#what-is-python-good-for>, diakses 20 Mei 2015.