

BAB 4

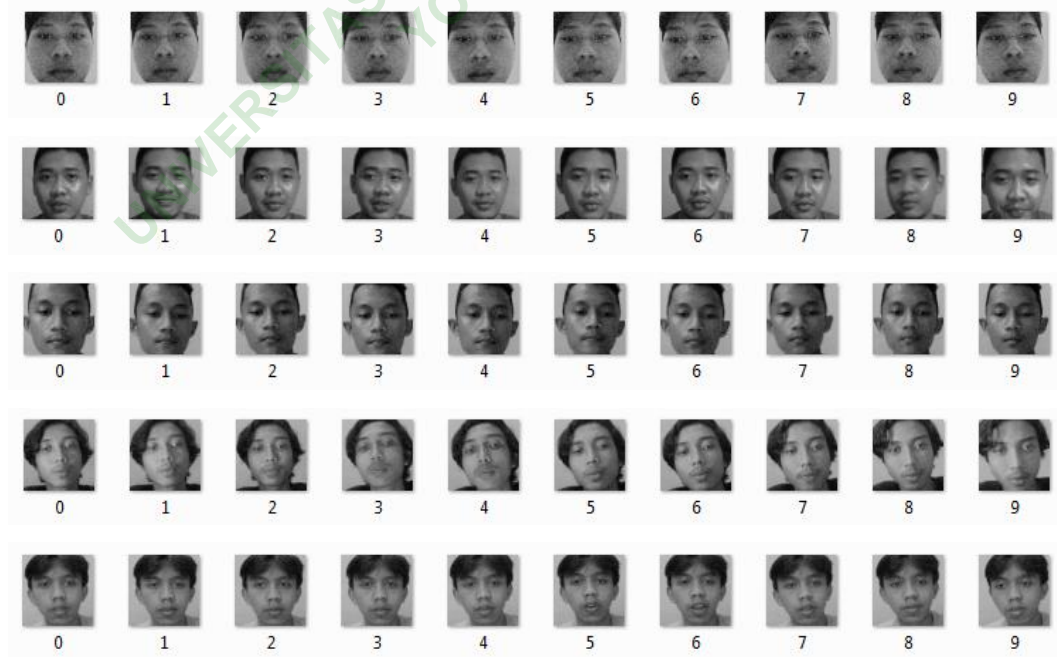
HASIL PENELITIAN

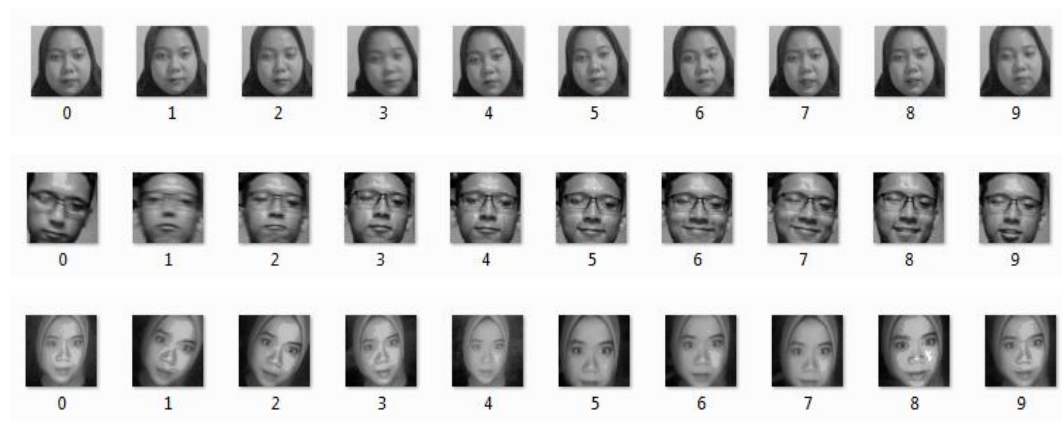
4.1 RINGKASAN HASIL PENELITIAN

Penelitian aplikasi presensi *online face recognition* menggunakan OpenCV pada studi kasus Dinas Komunikasi dan Informatika Kabupaten Mempawah berbasis web dengan menggunakan *framework* django dan bahasa pemrograman Python serta menggunakan *Database Management System* (DBMS) MySQL dalam mengelola data pengguna dan mengelola data presensi pada aplikasi presensi.

4.1.1 Analisis Kebutuhan

Dalam analisis kebutuhan ini menyediakan beberapa *dataset* untuk pengenalan wajah dimana pengumpulan *dataset* ini mengumpulkan sekitar 5 sampai 10 orang dan mengambil sampel wajah sebanyak 10 wajah perorang. Pembuatan *dataset* ini, kamera yang sudah siap untuk melakukan merekam wajah, foto yang telah diambil akan tersimpan kedalam folder *dataset*.





Gambar 4.1 *Dataset*

Setelah sampel wajah tersimpan di *dataset* yang dapat dilihat pada Gambar 4.1. Pada tahapan analisis kebutuhan proses pengenalan citra wajah dapat dideteksi secara *real time* dan ada tahapan antarmuka sistem yang akan ditampilkan. Peralatan penelitian disini akan di jelaskan aturan cara aplikasi web berjalan pada laptop dari tahapan *enrollment*, pembuatan *dataset* dan testing sistem.

4.1.2 Hasil Pengumpulan Data

Dari hasil pengumpulan data yang telah dilaksanakan penelitian pada tanggal 27-30 November 2021. Dalam penelitian ini mengumpulkan data terkait kebutuhan sistem presensi yang diperoleh dengan cara peninjauan di lingkungan kantor yang diteliti secara langsung di Diskominfo Kabupaten Mempawah. Beberapa teknik yang digunakan dalam penelitian ini yaitu melakukan pengamatan (*observasi*) dan wawancara secara langsung dalam upaya pengumpulan data dan pemahaman terhadap sistem yang berjalan sehingga dapat dilihat dan dipahami cara kerja sistem yang berjalan. Dalam melakukan proses *observasi* dan wawancara dilakukan dalam waktu 4 hari selama proses penelitian berlangsung. Adapun hasil pengumpulan data yang dapat dilihat pada lampiran 4.

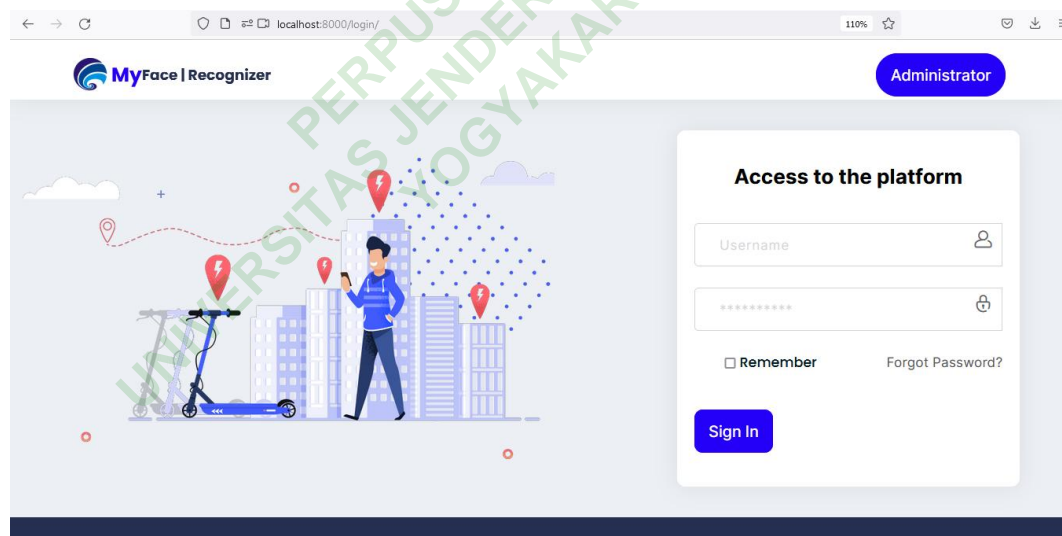
4.2 IMPLEMENTASI

Implementasi aplikasi merupakan hasil data yang sudah diperoleh pada rancangan yang telah dibuat sebelumnya. Dalam pengembangan aplikasi

presensi *online* berbasis web ini, bahasa pemrograman yang digunakan adalah Python dengan memanfaatkan *framework* django dan pengolahan datanya yaitu menggunakan basis data MySQL. Tampilan *interface* pada aplikasi memiliki 3 *role* hak akses yaitu staff pegawai, operator dan administrator. Staff pegawai akan berinteraksi dalam melakukan presensi kehadiran maupun ketidakhadiran, melihat data, melakukan penambahan data dan perubahan data. Operator akan berinteraksi sebagai otoritas konfirmasi kehadiran pegawai. sedangkan administrator merupakan pengguna yang memiliki hak akses penuh.

4.2.1 Form Login

Halaman *login* merupakan halaman yang digunakan oleh pengguna sebelum masuk ke dalam halaman *dashboard*. Dalam halaman *login*, staff pegawai atau operator harus menginputkan *username* dan *password* terlebih dahulu agar dapat masuk ke halaman *dashboard* sesuai *role* yang telah dibuat oleh pihak administrator. Adapun tampilan form *login* ditunjukkan pada Gambar 4.2



Gambar 4. 2 Tampilan *Form Login*

```

1. def login_view(request):
2.     form = LoginForm(request.POST or None)
3.     msg = None
4.     context_object_name = 'data_login'
5.     if request.method == 'POST':
6.         username = request.POST.get('username')
7.         password = request.POST.get('password')

```

```

8.     user = authenticate(request, username=username,
9.     password=password)
10.    if user is not None:
11.        if user is not None and user.is_superuser:
12.            messages.info(request, 'special form is staf!')
13.            return redirect('/login/')
14.        else:
15.            pass
16.
17.    if user is not None and user.is_staff:
18.        try:
19.            akun_operator=User.objects.get(pk=user.id)
20.            auth_login(request, user)
21.            request.session['id']=akun_operator.id
22.            request.session['is_staff']=akun_operator.is_staff
23.            request.session['username']=request.POST['username']
24.            request.session['password']=request.POST['password']
25.            return HttpResponseRedirect(reverse_lazy('Dashboard
26.            Operator'))
27.        except:
28.            messages.add_message(request, messages.INFO('akun
29.            yang anda masukkan salah!'))
30.    if user is not None and user.is_active:
31.        try:
32.            akun_pegawai = User.objects.get(pk=user.id)
33.            auth_login(request, user)
34.            request.session['id']=akun_pegawai.id
35.            request.session['is_active']=akun_pegawai.is_active
36.            request.session['username']=request.POST['username']
37.            request.session['password']=request.POST['password']
38.            return HttpResponseRedirect(reverse_lazy('Dashboard
39.            Pegawai'))
40.        except:
41.            messages.add_message(request, messages.INFO('akun
42.            yang anda masukkan salah!'))
43.        else:
44.            return HttpResponseRedirect("Akun tidak aktif")
45.    else:
46.        print("username: {} and password {}".format(username,
47.        password))
48.        messages.info(request, 'pengguna & kata sandi salah!')
49.    else:
50.        return render(request, 'login_view.html', {'name':
51.        request.user.username, 'form':form, 'msg':msg})

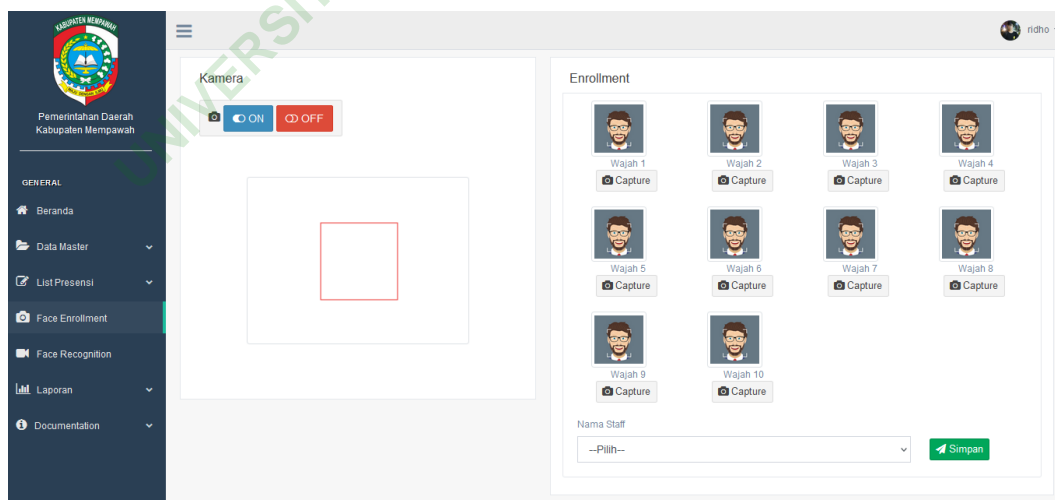
```

Dari potongan kode diatas merupakan kode untuk pembuatan *login* masing-masing *role* untuk mengakses masuk ke dalam sistem. Dalam kode ini dibuat 3 hak akses yaitu administrator, staff pegawai, dan operator. kode baris 5 berfungsi untuk mengirim data atau nilai langsung ke dalam *action* untuk ditampung, tanpa menampilkan di URL. Kode baris 10 akan melakukan

pengecekan apakah ada *user* yang dikirimkan, kode baris 11 sampai 13 menjelaskan jika *user* tidak ada dan *user* adalah *superuser* maka akan menampilkan *messages* "form ini khusus akun staff" dan akan meredirect ke halaman *login*. Kode baris 17 sampai 29 merupakan kode untuk mencoba *login* sebagai operator yang dimana jika *user* tidak ada dan *user* adalah operator maka *user* mencoba melakukan *authenticate login* dan merequest *session id* operator. Kode baris 30-42 merupakan kode untuk *login* sebagai pegawai yang dimana jika *user* adalah pegawai maka *user* mencoba melakukan *auth_login* dan merequest *session id* pegawai, jika data yang di masukkan *user valid* maka sistem akan melakukan *HttpResponseRedirect* ke halaman *dashboard*. Kode baris 48 merupakan kode yang menampilkan *messages* "*username* dan *password* yang dimasukkan tidak *valid*".

4.2.2 Face Enrollment

Halaman *face enrollment* merupakan halaman yang menampilkan fitur pendaftaran wajah. Fungsi dari fitur *face enrollment* adalah untuk mendaftarkan data wajah agar wajah dapat dikenali oleh sistem. Dalam halaman ini terdapat tombol *on/off* untuk mengaktifkan *camera* dan tombol *capture* untuk mengambil sampel wajah. Adapun tampilan *face enrollment* ditunjukkan pada Gambar 4.3



Gambar 4.3 Face Enrollment

Berikut ini kode untuk mengambil sampel wajah dalam fitur *face enrollment* dalam *script* Python dan HTML

```

1. <div class="box-body">
2.   <div class="btn-group">
3.     <span class="btn btn-default disabled"><span class="
4.       fa fa-camera"></span>
5.     <button class="btn btn-primary camera-on"><i class="
6.       fa fa-toggle-on"></i> ON</button>
7.     <button class="btn btn-danger camera-off"><i class="
8.       fa fa-toggle-off"></i> OFF</button>
9.   </div>
10. <div class="camera-container center-block img-thumbnail">
11.   <video id="video" width="320" height="240" preload
12.     autoplay loop muted></video>
13.   <canvas id="canvas" width="320" height="240"></canvas>
14. </div>
15.</div>

```

Potongan kode di atas merupakan kode untuk membuat tombol *on* dan *off* pada *camera* web.

```

1. @login_required(login_url=settings.LOGIN_URL)
2. @csrf_exempt
3. def enroll(request):
4.     simpan = False
5.     if request.method == 'POST':
6.         faces = [str(request.POST['face_0']).encode('ascii'),
7.                 str(request.POST['face_1']).encode('ascii'),
8.                 str(request.POST['face_2']).encode('ascii'),
9.                 str(request.POST['face_3']).encode('ascii'),
10.                str(request.POST['face_4']).encode('ascii'),
11.                str(request.POST['face_5']).encode('ascii'),
12.                str(request.POST['face_6']).encode('ascii'),
13.                str(request.POST['face_7']).encode('ascii'),
14.                str(request.POST['face_8']).encode('ascii'),
15.                str(request.POST['face_9']).encode('ascii')]
16.         id_pegawai = request.POST['id_pegawai']
17.         save_faces(id_pegawai, faces)
18.         simpan = True
19.         return render(request, 'enrollment.html', {'active': 'enrol',
20.             'simpan': simpan})
21.
22. def save_faces(id_pegawai, faces):
23.     idx = 0
24.     dirname = os.path.join(FOTO_PATH, "{}_wajah".format(
25.         id_pegawai))
26.     if not os.path.exists(dirname):
27.         os.makedirs(dirname)
28.         clear_dir(dirname)
29.
30.     for face in faces:
31.         with open(os.path.join(dirname, "{}.png".format(idx)), 'wb')
32.             fh.write(base64.decodebytes(face))
33.         idx += 1

```

```

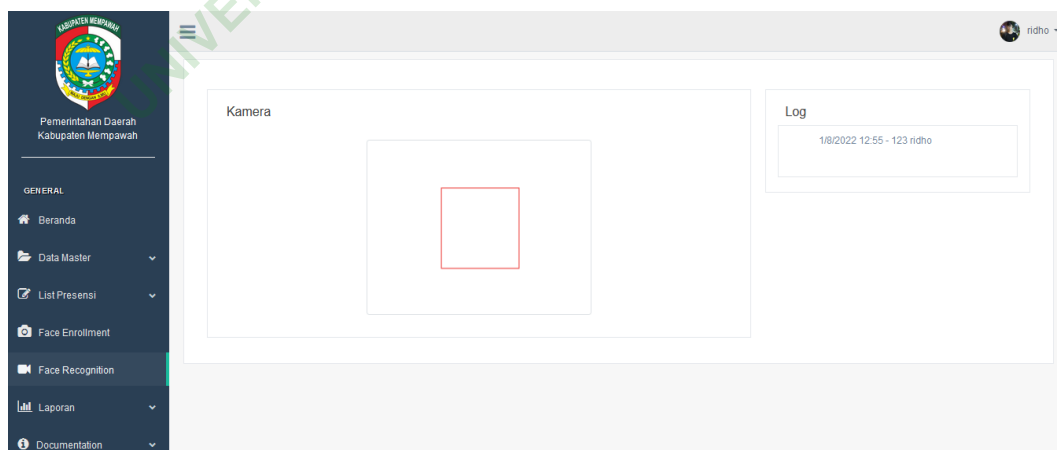
34. return dirname
35.
36. def save_face(dirname, name, face):
37.     if not os.path.exists(dirname):
38.         os.makedirs(dirname)
39.         clear_dir(dirname)
40.         with open(os.path.join(dirname, "{}.png".format(name)), 'wb')
41.             as fh:
42.             fh.write(base64.decodebytes(face))

```

Dari potongan kode di atas merupakan kode untuk mengambil *dataset* wajah dengan tujuan agar sistem dapat mengenali wajah setiap *user*. Pada baris kode 1 merupakan *decorator* yang berfungsi untuk melakukan proses autentikasi terlebih dahulu. Kode baris 5 merequest *method post*, dan pada kode baris 6 sampai 20 merupakan suatu proses untuk mengubah karakter yang ada pada *character set*. Kode baris 17 akan menyimpan data id pegawai dan hasil *capture image* kedalam folder *dataset*.

4.2.3 Face Recognition

Halaman *face recognition* merupakan halaman untuk mendeteksi wajah. Dalam halaman ini sistem akan mengenali wajah pengguna yang sebelumnya telah didaftarkan pada tahap *face enrollment*, kemudian sistem akan membaca identitas pengguna yang akan tampil pada log disamping. Halaman *face recognition* dapat dilihat pada Gambar 4.4.



Gambar 4. 4 *Face Recognition*

```

1. FOTO_PATH = os.path.join(BASE_DIR, "static\dataset")
2. pcaRecognizer = cv2.face.LBPHFaceRecognizer_create()
3. TEMP_DIR = os.path.join(BASE_DIR, "temporary")
4.
5. @login_required(login_url=settings.LOGIN_URL)
6. @csrf_exempt
7. def absen(request):
8.     imgs, lbl = prepare_training_data(FOTO_PATH)
9.     print(pcaRecognizer.train(imgs, np.array(lbl)))
10.    current_user = request.user
11.    current_account = Staff_Pegawai.objects.get(
12.        user_id=current_user.id)
13.    emp_id = current_account.id
14.
15.    if request.method == 'POST':
16.        face = str(request.POST['face']).encode('ascii')
17.        clear_dir(TEMP_DIR)
18.        save_face(TEMP_DIR, 'tmp', face)
19.        data = cv2.imread(os.path.join(TEMP_DIR, 'tmp.png'))
20.        data = cv2.cvtColor(data, cv2.COLOR_BGR2GRAY)
21.        pg, predict_label = predict(data, emp_id)
22.        print("tes", predict_label)
23.        print("tes", emp_id)
24.        if predict_label[0] == emp_id:
25.            log = Facerecognition.objects.filter(pegawai_id=pg.id,
26.                tanggal=date.today())
27.            if len(log):
28.                log = Facerecognition.objects.get(pk=log[0].id)
29.                lg.waktu = datetime.now()
30.                lg.save(force_update=True)
31.            else:
32.                Facerecognition(keterangan="Wajah dikenal sebagai {}".
33.                    format(pg.user.username),
34.                    pegawai = pg,
35.                    pegawai_id = emp_id,
36.                    status = 'Hadir'
37.                    user = request.user,
38.                    tanggal = datetime.now(),
39.                    waktu = datetime.now()).save()
40.        else:
41.            Facerecognition(keterangan="Wajah tidak dikenal",
42.                status='Tanpa_Keterangan',
43.                pegawai_id=emp_id,
44.                user=request.user,
45.                tanggal=datetime.now(),
46.                waktu=datetime.now()).save()
47.    logs = Facerecognition.objects.filter(tanggal=date.today())
48.    return render(request, get_presensi_facerecognition.html,
49.        {'active': 'absen', 'logs':logs})

```

Dari potongan kode diatas merupakan kode untuk pembuatan fitur presensi *face recognition* menggunakan *library* OpenCV dari Python. Kode baris 2

berfungsi untuk mendefinisikan suatu fungsi perulangan untuk melatih sistem mengenali gambar. Oleh karena itu dibutuhkan sebuah algoritma *recognizer* yang telah disediakan oleh OpenCV yaitu *local binary patterns histogram*. Kode baris 5 merupakan *decorator* untuk melakukan autentikasi *user* terlebih dahulu, Kode baris 8 sampai 9 akan melakukan *prepare training* data yang berada pada *path* folder *dataset* dan merubah label menjadi nilai *array*. Selain proses pengenalan wajah juga ada proses untuk *convert image* menjadi *grayscale* pada baris 20 menggunakan perintah *cv2.cvtColor* dengan parameter *cv2.COLOR_BGR2GRAY*. Kode baris 24 sampai 46 proses untuk memprediksi label nilai data berdasarkan model yang dilatih, jika prediksi data *valid* maka wajah dikenali sebagai *user* jika tidak maka wajah tidak dikenali.

4.2.4 List Presensi Face Recognition

Halaman *list* presensi *face recognition* merupakan halaman yang menampilkan informasi kehadiran maupun ketidakhadiran. Pada halaman ini staff pegawai dapat melihat informasi data presensi pegawai dan dapat menghapus data presensi. Data keterangan yang ditampilkan pada halaman *list* presensi *face recognition* yaitu 'wajah dikenali sebagai *user*' atau 'wajah tidak dikenali'. Adapun tampilan *list* presensi *face recognition* yang ditunjukkan pada Gambar 4.5.

<input type="checkbox"/>	Pegawai	Keterangan	Status	Waktu masuk	Waktu keluar	Action
<input type="checkbox"/>	ridho	Wajah dikenal sebagai ridho	Hadir	29 Juli 2022, 22.30	29 Juli 2022, 22.30	
<input type="checkbox"/>	ridho	Wajah dikenal sebagai ridho	Hadir	1 Agu 2022, 12.06	1 Agu 2022, 12.06	

Gambar 4.5 List Presensi Face Recognition

Berikut ini merupakan kode untuk menampilkan data presensi *face recognition* dalam bentuk *script* Python dan HTML

```

1. @login_required(login_url=settings.LOGIN_URL)
2. @csrf_exempt
3. def faceRecognizer_list(request):
4.     context = {
5.         'daftar_faceRecognizer': Facerecognition.objects.all()
6.         .filter(user=request.user)
7.     }
8.     return render(request, 'faceRecognizerlist.html', context)

```

Kode di atas merupakan kode untuk mengambil semua objek data yang berada di dalam *database* dan melakukan *filter* data *user* untuk ditampilkan ke *template*.

```

1. <table id="example" class="table is-striped table-hover">
2.     <thead>
3.         <tr class="headings">
4.             <th><input type="checkbox" id="check-all"></th>
5.             <th class="dtr-control">Detail</th>
6.             <td>Pegawai</td>
7.             <td>Keterangan</td>
8.             <td>Status</td>
9.             <td>Tanggal</td>
10.            <td>Waktu</td>
11.            <td>Action</td>
12.        </tr>
13.    </thead>
14.    <tbody>
15.        {% for face in daftar_faceRecognizer %}
16.            <tr class="even pointer">
17.                <td><input type="checkbox" id="check-all"></td>
18.                <td>{{face.pegawai}}</td>
19.                <td>{{face.keterangan}}</td>
20.                <td>{{face.status}}</td>
21.                <td>{{face.tanggal}}</td>
22.                <td>{{face.waktu}}</td>
23.                <td>
24.                    {% endfor %}
25.            </tbody>
26. </table>

```

Dari potongan kode diatas merupakan kode untuk menampilkan informasi data presensi dalam tabel *face recognition*. Kode baris 1 merupakan *script* pembuka untuk membuat tampilan tabel. Baris kode 3 sampai 12 untuk membuat baris dan kolom sesuai dengan data yang ada pada kolom tabel. Kode pada baris 15 sampai 24 berfungsi untuk mengambil data dari perulangan hasil *parsing* data dari kode Python yang berfungsi untuk menampilkan data pegawai, keterangan, status, tanggal, dan waktu.

4.3 DATABASE

Dalam pembuatan basis data dan tabel adalah langkah awal dalam mengimplementasikan sebuah sistem. pada aplikasi presensi *online* ini menggunakan *database* MySQL pada bab sebelumnya telah dijelaskan rancangan *database* yang akan digunakan. Berikut ini implementasi *database* MySQL yang diberi nama *myface_db* yang didalamnya terdapat 18 tabel *database*.

Pada penelitian ini *database* MySQL digunakan untuk menyimpan data diri staff pegawai, operator, dan data presensi pegawai. Pada *database* terdapat beberapa tabel, yaitu tabel *face recognition*, tabel presensi manual, tabel operator, tabel staff pegawai, tabel golongan pegawai, tabel jabatan pegawai, dan tabel pensiun pegawai. Adapun tampilan implementasi *database* MySQL ditunjukkan pada Gambar 4.6.

Tabel	Tindakan	Baris	Jenis	Penyortiran	Ukuran	Beban
auth_group	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_general_ci	32.0 KB	-
auth_group_permissions	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_general_ci	48.0 KB	-
auth_permission	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	56	InnoDB	utf8mb4_general_ci	32.0 KB	-
auth_user	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	4	InnoDB	utf8mb4_general_ci	32.0 KB	-
auth_user_groups	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	0	InnoDB	utf8mb4_general_ci	48.0 KB	-
auth_user_user_permissions	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	168	InnoDB	utf8mb4_general_ci	48.0 KB	-
django_admin_log	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	11	InnoDB	utf8mb4_general_ci	48.0 KB	-
django_content_type	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	14	InnoDB	utf8mb4_general_ci	48.0 KB	-
django_migrations	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	41	InnoDB	utf8mb4_general_ci	16.0 KB	-
django_session	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	3	InnoDB	utf8mb4_general_ci	32.0 KB	-
face_recognition_facerecognition	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	24	InnoDB	utf8mb4_general_ci	48.0 KB	-
presensi_manual_presensi_manual	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	10	InnoDB	utf8mb4_general_ci	48.0 KB	-
staff_operator_staff_operator	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	1	InnoDB	utf8mb4_general_ci	32.0 KB	-
staff_pegawai_golongan_pegawai	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	1	InnoDB	utf8mb4_general_ci	16.0 KB	-
staff_pegawai_jabatan_pegawai	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	2	InnoDB	utf8mb4_general_ci	16.0 KB	-
staff_pegawai_pegawai_pensiun	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	1	InnoDB	utf8mb4_general_ci	32.0 KB	-
staff_pegawai_staff_pegawai	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	2	InnoDB	utf8mb4_general_ci	80.0 KB	-
thumbnail_kvstore	☆ Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	9	InnoDB	utf8mb4_general_ci	16.0 KB	-
18 tabel	Jumlah	347	InnoDB	utf8mb4_general_ci	672.0 KB	0 B

Gambar 4. 6 Implementasi *database* MySQL

4.4 PENGUJIAN SISTEM

Pengujian sistem dilakukan dengan *blackbox testing* dengan menjalankan dan mengeksekusi fungsi, kemudian diamati apakah hasil dari unit tersebut sesuai dengan proses bisnis yang diinginkan.

4.4.1 Hasil Pengujian Fungsi *Login* Staff Pegawai

Pengujian pada halaman *login* staff pegawai merupakan proses pengujian halaman web yang dibuat. Hasil pengujian pada *form login* terdiri dari fungsi, skenario, hasil yang diharapkan dan keterangan. Hasil pengujian pada halaman *login* pegawai dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Pengujian Fungsi *Login*

No	Fungsi	Skenario	Hasil yang diharapkan	Keterangan
1	<i>Login</i>	Melakukan <i>login</i> dengan <i>username</i> dan <i>password</i> yang <i>valid</i>	Berhasil masuk kedalam <i>dashboard</i> pegawai	Berhasil
2	<i>Login</i>	Melakukan <i>login</i> dengan <i>username</i> dan <i>password</i> yang tidak <i>valid</i>	Tidak dapat masuk kedalam sistem dan menampilkan gagal <i>login</i>	Berhasil
3	<i>Log Out</i>	Mengklik <i>Log Out</i>	Berhasil melakukan <i>logout</i>	Berhasil

4.4.2 Hasil Pengujian Fungsi *Login* Operator

Pengujian pada halaman *login* operator merupakan proses pengujian halaman web yang dibuat. Hasil pengujian pada *form login* operator terdiri dari fungsi, skenario, hasil yang diharapkan, dan keterangan. Hasil pengujian pada halaman *login* operator dapat dilihat pada Tabel 4.2.

Tabel 4. 2 Pengujian Fungsi *Login* Operator

No	Fungsi	Skenario	Hasil yang diharapkan	Keterangan
1	<i>Login</i>	Melakukan <i>login</i> dengan <i>username</i> dan <i>password</i> yang <i>valid</i>	Berhasil masuk kedalam dashboard operator	Berhasil
2	<i>Login</i>	Melakukan <i>login</i> dengan <i>username</i> dan <i>password</i> yang tidak <i>valid</i>	Tidak dapat masuk kedalam sistem dan menampilkan gagal <i>login</i>	Berhasil
3	<i>Log Out</i>	Mengklik <i>logout</i>	Berhasil melakukan <i>logout</i>	Berhasil

4.4.3 Hasil Pengujian Menu Staff Pegawai

Pengujian menu staff pegawai merupakan proses pengujian halaman web yang dibuat. Hasil pengujian pada menu staff pegawai terdiri dari fungsi, skenario, hasil yang diharapkan, dan keterangan. Tabel hasil pengujian dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Pengujian Menu Staff Pegawai

No	Fungsi	Skenario	Hasil yang diharapkan	Keterangan
1	Data staff pegawai	Klik menu data master dan klik staff pegawai	Menampilkan seluruh data-data pegawai	Berhasil

4.4.4 Hasil Pengujian Menu Jabatan Pegawai

Pengujian menu jabatan pegawai merupakan proses pengujian halaman web yang dibuat. Hasil pengujian pada menu jabatan pegawai terdiri dari fungsi, skenario, hasil yang diharapkan, dan keterangan. Hasil pengujian pada menu jabatan pegawai dapat dilihat pada Tabel 4.4.

Tabel 4. 4 Pengujian Menu Jabatan Pegawai

No	Fungsi	Skenario	Hasil yang diharapkan	Keterangan
1	Jabatan pegawai	Klik menu data jabatan pegawai	Menampilkan data-data jabatan pegawai	Berhasil
2	Tambah data jabatan	Klik tombol tambah dan <i>input</i> data jabatan baru	Berhasil menambah data jabatan pegawai	Berhasil
3	Ubah data jabatan	Klik ubah dan mengubah data jabatan	Berhasil mengubah data jabatan	Berhasil
4	Hapus data jabatan	Klik tombol hapus	Berhasil hapus data	Berhasil
5	Cari data jabatan	Masukkan pencarian dikolom pencarian data	Menampilkan data kata kunci yang sesuai	Berhasil

4.4.5 Hasil Pengujian Menu Golongan Pegawai

Pengujian menu golongan pegawai merupakan proses pengujian halaman web yang dibuat. Hasil pengujian pada menu golongan pegawai terdiri dari fungsi, skenario, hasil yang diharapkan, dan keterangan. Hasil pengujian pada menu golongan pegawai dapat dilihat pada Tabel 4.5.

Tabel 4. 5 Pengujian Menu Golongan Pegawai

No	Fungsi	Skenario	Hasil yang diharapkan	Keterangan
1	Golongan pegawai	Klik menu golongan pegawai	Menampilkan data golongan pegawai	Berhasil
2	Tambah data golongan	Klik tambah golongan dan <i>input</i> data golongan baru	Berhasil menyimpan data golongan	Berhasil
3	Ubah data golongan	Klik ubah data golongan dan mengubah data golongan	Berhasil mengubah data golongan pegawai	Berhasil
4	Hapus data	Klik tombol hapus	Berhasil hapus data	Berhasil

	golongan			
5	Cari data golongan	Masukkan pencarian dikolom pencarian data	Menampilkan data kata kunci yang sesuai	Berhasil

4.4.6 Hasil Pengujian Menu Pensiun

Pengujian menu pensiun merupakan proses pengujian halaman web yang dibuat. Hasil pengujian pada menu pensiun terdiri dari fungsi, skenario, hasil yang diharapkan, dan keterangan. Hasil pengujian pada menu pensiun dapat dilihat pada Tabel 4.6.

Tabel 4. 6 Pengujian Menu Pensiun

No	Fungsi	Skenario	Hasil yang diharapkan	Keterangan
1	Pegawai pensiun	Klik menu pensiun	Menampilkan data pegawai pensiun	Berhasil
2	Tambah data pensiun	Klik tambah pensiun dan <i>input</i> data pensiun baru	Berhasil menyimpan data pensiun	Berhasil
3	Ubah data pensiun	Klik ubah data pensiun dan mengubah data pensiun	Berhasil mengubah data pensiun	Berhasil
4	Hapus data pensiun	Klik tombol hapus	Berhasil hapus data pensiun	Berhasil
5	Cari data pegawai pensiun	Masukkan pencarian dikolom pencarian data	Menampilkan data kata kunci yang sesuai	Berhasil

4.4.7 Hasil Pengujian Presensi Manual

Pengujian menu presensi manual merupakan proses pengujian halaman web yang dibuat. Hasil pengujian pada menu presensi manual terdiri dari fungsi, skenario, hasil yang diharapkan, dan keterangan. Hasil pengujian pada menu presensi manual dapat dilihat pada Tabel 4.7.



Tabel 4. 7 Pengujian Presensi Manual




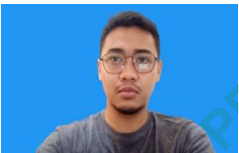



No	Fungsi	Skenario	Hasil yang diharapkan	Keterangan
1	Presensi manual	Klik menu data master lalu klik presensi manual	Menampilkan list data presensi pegawai	Berhasil
2	Tambah presensi manual	Klik tombol tambah presensi dan <i>input</i> data presensi	Berhasil menyimpan data presensi	Berhasil
3	Ubah data presensi	Klik ubah data presensi dan mengubah data presensi	Berhasil merubah data	Berhasil
4	Hapus data presensi	Klik tombol hapus	Berhasil hapus data presensi	Berhasil
5	Cari data presensi	Masukkan pencarian dikolom pencarian data	Menampilkan data yang dicari	Berhasil







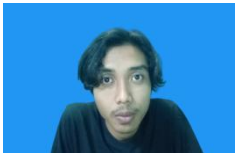

4.4.8 Hasil Pengujian Presensi *Face Recognition*








Pengujian presensi *face recognition* merupakan proses pengujian halaman web yang dibuat. Hasil pengujian presensi *face recognition* terdiri dari hasil *capture image*, kondisi saat presensi, wajah dikenali, dan wajah tidak dikenali. Hasil pengujian pada presensi *face recognition* dapat dilihat pada Tabel 4.8.





Tabel 4. 8 Pengujian Presensi *Face Recognition*

No	Citra wajah	Hasil <i>capture image</i>	Kondisi saat presensi	Wajah dikenali	Wajah tidak dikenali
1			Kondisi pengguna saat presensi menguji satu individu dalam kumpulan <i>dataset</i>	✓	

			Kondisi pengguna saat presensi pengujian terhadap satu individu yang tidak ada dalam kumpulan <i>dataset</i>		✓
			Kondisi pengguna saat presensi menggunakan penutup masker	✓	
			Kondisi pengguna saat presensi menggunakan aksesoris kaca mata	✓	
2			Kondisi pengguna saat presensi menguji satu individu dalam kumpulan <i>dataset</i>	✓	
			Kondisi pengguna saat presensi pengujian terhadap satu individu yang tidak ada dalam kumpulan <i>dataset</i>		✓
			Kondisi pengguna saat presensi menggunakan penutup masker	✓	

			Kondisi pengguna saat presensi menggunakan aksesoris kacamata	✓	
3			Kondisi pengguna saat presensi menguji satu individu dalam kumpulan <i>dataset</i>	✓	
			Kondisi pengguna saat presensi pengujian terhadap satu individu yang tidak ada dalam kumpulan <i>dataset</i>	✓	
			Kondisi pengguna saat presensi menggunakan penutup masker		✓
			Kondisi pengguna saat presensi menggunakan aksesoris kacamata	✓	
4			Kondisi pengguna saat presensi menguji satu individu dalam kumpulan <i>dataset</i>	✓	

			Kondisi pengguna saat presensi pengujian terhadap satu individu yang tidak ada dalam kumpulan <i>dataset</i>		✓
			Kondisi pengguna saat presensi menggunakan penutup masker	✓	
			Kondisi pengguna saat presensi menggunakan aksesoris kacamata	✓	
5			Kondisi pengguna saat presensi menguji satu individu dalam kumpulan <i>dataset</i>	✓	
			Kondisi pengguna saat presensi pengujian terhadap satu individu yang tidak ada dalam kumpulan <i>dataset</i>		✓
			Kondisi pengguna saat presensi menggunakan penutup masker		✓

			Kondisi pengguna saat presensi menggunakan aksesoris kacamata	✓	
6			Kondisi pengguna saat presensi menguji satu individu dalam kumpulan <i>dataset</i>	✓	
			Kondisi pengguna saat presensi pengujian terhadap satu individu yang tidak ada dalam kumpulan <i>dataset</i>		✓
			Kondisi pengguna saat presensi menggunakan penutup masker	✓	
			Kondisi pengguna saat presensi menggunakan aksesoris kacamata	✓	

4.4.9 Hasil Pengujian Akurasi

Pengujian untuk mencari akurasi pengenalan wajah dengan membagi hasil *test* atau sampel wajah yang berhasil dengan jumlah sampel percobaan dikali dengan 100%. Berikut adalah rumus yang digunakan untuk menghitung akurasi dapat dilihat pada rumus (1).

$$Akurasi (\varphi) = \frac{\text{Test yang berhasil}}{\text{Jumlah Percobaan}} \times 100\% \quad (1)$$

False acceptance rate (FAR) merupakan tingkat kesalahan penerimaan dalam mengenali identitas gambar masukkan. Berikut adalah rumus yang digunakan untuk menghitung FAR dapat dilihat pada rumus (2).

$$FAR = \frac{\text{Banyak FAR}}{\text{Jumlah Percobaan}} \times 100\% \quad (2)$$

False rejection rate (FRR) adalah kesalahan dalam menolak gambar masukkan. Sebuah gambar masukkan yang seharusnya dapat dikenali berubah menjadi tidak dikenali. Berikut merupakan rumus yang digunakan untuk menghitung FRR dapat dilihat pada rumus (3).

$$FRR = \frac{\text{Banyak FRR}}{\text{Jumlah Percobaan}} \times 100\% \quad (3)$$

Pengujian akurasi merupakan proses pengujian pada fitur *face recognition* yang dibuat. Hasil pengujian akurasi terdiri dari akurasi, *false rejection rate*, dan *false acceptance rate*. Tabel hasil pengujian akurasi dapat dilihat pada Tabel 4.9.

Tabel 4. 9 Pengujian Akurasi

Percobaan ke	LBPH		
	Akurasi (%)	FRR (%)	FAR (%)
Pengujian 1	100.00	0	0
Pengujian 2	100.00	0	0
Pengujian 3	80.00	20.00	10.00
Pengujian 4	100.00	0	0
Pengujian 5	80.00	20.00	10.00
Pengujian 6	90.00	10.00	10.00

Pengujian 7	90.00	10.00	10.00
Pengujian 8	80.00	20.00	10.00
Pengujian 9	100.00	0	0
Pengujian 10	100.00	0	0
Rata-rata	92	8	5

4.4.10 Hasil Pengujian *Face Enrollment*

Pengujian *face enrollment* merupakan proses pengujian halaman web yang dibuat. Hasil pengujian *face enrollment* terdiri dari fungsi, skenario, hasil yang diharapkan, dan keterangan. Tabel hasil pengujian *face enrollment* dapat dilihat pada Tabel 4.10.

Tabel 4. 10 Pengujian *Face Enrollment*

No	Fungsi	Skenario	Hasil yang diharapkan	Keterangan
1	<i>Face enrollment</i>	Klik menu <i>face enrollment</i>	Menampilkan <i>fitur face enrollment</i>	Berhasil
2	<i>On camera</i>	Klik <i>on camera</i>	Mengaktifkan <i>camera</i>	Berhasil
3	<i>Off camera</i>	Klik tombol <i>off camera</i>	Menonaktifkan <i>camera</i>	Berhasil
4	Tambah gambar wajah	Klik tombol <i>on</i> dan <i>capture</i> wajah	Mengambil sampel wajah	Berhasil
5	Simpan gambar wajah	Pilih nama pegawai lalu klik simpan	Menyimpan data wajah ke dalam <i>dataset</i>	Berhasil

4.4.11 Hasil Pengujian Menu *Report*

Pengujian menu *report* merupakan proses pengujian halaman web yang dibuat. Hasil pengujian menu *report* terdiri dari fungsi, skenario, hasil yang diharapkan, dan keterangan. Tabel hasil pengujian menu *report* dapat dilihat pada Tabel 4.11.

Tabel 4. 11 Pengujian Menu *Report*

No	Fungsi	Skenario	Hasil yang diharapkan	Keterangan
1	Menu <i>report</i>	Klik menu laporan lalu klik menu <i>report</i>	Menampilkan data kehadiran dan ketidakhadiran pegawai	Berhasil
2	Cari data pegawai	Masukkan pencarian di kolom pencarian data	Menampilkan data kata kunci yang sesuai	Berhasil
3	<i>Export file pdf, excel, dan print</i>	Pilih salah satu <i>export file</i>	Menampilkan data dalam bentuk <i>softfile</i> dan mendownload hasil <i>export</i>	Berhasil
4	<i>Filter tanggal</i>	Pilih dari tanggal, sampai tanggal yang sesuai	Menampilkan data kehadiran maupun ketidakhadiran berdasarkan kata kunci tanggal yang sesuai	berhasil

4.4.12 Hasil Pengujian Edit Profil Pegawai

Pengujian menu edit profil pegawai merupakan proses pengujian halaman web yang dibuat. Hasil pengujian pada halaman edit profil pegawai terdiri dari fungsi, skenario, hasil yang diharapkan, dan keterangan. Tabel hasil pengujian dapat dilihat pada Tabel 4.12.

Tabel 4. 12 Pengujian Edit Profil Pegawai

No	Fungsi	Skenario	Hasil yang diharapkan	Keterangan
1	Profil pegawai	Klik foto profil di pojok kanan atas lalu klik menu profil	Menampilkan data profil pegawai	Berhasil
2	Edit profil	Klik tombol edit profil dan mengubah data profil	Berhasil merubah data	Berhasil

4.4.13 Hasil Pengujian Ubah *Password* Pegawai

Pengujian menu ubah *password* pegawai merupakan proses pengujian halaman web yang dibuat. Hasil pengujian pada halaman ubah *password* pegawai terdiri dari fungsi, skenario, hasil yang diharapkan, dan keterangan. Tabel hasil pengujian dapat dilihat pada Tabel 4.13.

Tabel 4. 13 Pengujian Ubah *Password* Pegawai

No	Fungsi	Skenario	Hasil yang diharapkan	Keterangan
1	Ubah <i>password</i>	Klik foto profil di pojok kanan atas lalu klik menu ganti <i>password</i>	Menampilkan <i>form</i> data <i>password</i>	Berhasil
2	Ubah <i>password</i>	Memasukkan sandi lama dan sandi baru	<i>Password</i> berhasil di ubah	Berhasil

4.4.14 Hasil Pengujian Menu Rekap Presensi Operator

Pengujian menu rekap presensi operator merupakan proses pengujian halaman web yang dibuat. Hasil pengujian pada halaman rekap presensi operator terdiri dari fungsi, skenario, hasil yang diharapkan, dan keterangan. Hasil pengujian pada halaman rekap presensi operator dapat dilihat pada Tabel 4.14.

Tabel 4. 14 Pengujian Menu Rekap Presensi Operator

No	Fungsi	Skenario	Hasil yang diharapkan	Keterangan
1	Menu rekap presensi	Klik menu rekap presensi lalu klik menu daftar presensi pegawai	Menampilkan keseluruhan data presensi pegawai	Berhasil
2	Konfirmasi presensi	Klik tombol <i>confirm</i> lalu ubah data konfirmasi	Berhasil merubah data konfirmasi	Berhasil

4.4.15 Hasil Pengujian Menu Tambah Pegawai

Pengujian menu tambah pegawai merupakan proses pengujian halaman web yang dibuat. Hasil pengujian pada halaman tambah pegawai terdiri dari fungsi, skenario, hasil yang diharapkan, dan keterangan. Hasil pengujian pada menu tambah pegawai dapat dilihat pada Tabel 4.15.

Tabel 4. 15 Pengujian Menu Tambah Pegawai

No	Fungsi	Skenario	Hasil yang diharapkan	Keterangan
1	Menu tambah pegawai	Klik menu tambah pegawai	Menampilkan keseluruhan data pegawai	Berhasil
2	Tambah data pegawai	Klik tombol tambah data staff dan <i>input</i> data pegawai	Berhasil menambahkan data pegawai	Berhasil
3	Hapus data pegawai	Klik tombol hapus	Berhasil hapus data	Berhasil
4	<i>Read</i> data pegawai	Klik <i>icon read</i>	Berhasil menampilkan data dengan modal	Berhasil

4.4.16 Hasil Pengujian Edit Profil Operator

Pengujian edit profil operator merupakan proses pengujian halaman web yang dibuat. Hasil pengujian pada halaman edit profil operator terdiri dari fungsi, skenario, hasil yang diharapkan, dan keterangan. Hasil pengujian pada menu edit profil operator dapat dilihat pada Tabel 4.16.

Tabel 4. 16 Pengujian Edit Profil Operator

No	Fungsi	Skenario	Hasil yang diharapkan	Keterangan
1	Profil operator	Klik foto profil di pojok kanan atas lalu klik menu profil	Menampilkan data profil operator	Berhasil
2	Edit profil operator	Klik tombol edit profil dan mengubah data profil	Berhasil merubah data	Berhasil

4.5 FITUR-FITUR SISTEM

Sistem presensi *online face recognition* pada studi kasus Dinas Komunikasi dan Informatika Kabupaten Mempawah memiliki fitur-fitur sebagai berikut:

Staff Pegawai:

1. Menampilkan data pegawai dengan *datatable*
2. Mengedit data profil pegawai
3. Menampilkan, mengubah, menambah, dan menghapus data jabatan pegawai
4. Menampilkan, mengubah, menambah, dan menghapus data golongan pegawai
5. Menampilkan, mengubah, menambah, dan menghapus data pensiun pegawai
6. Fitur pencarian data
7. Menampilkan, mengubah, menambah, dan menghapus data presensi manual
8. Fitur *face enrollment*
9. Fitur *face recognition*
10. Menampilkan dan menghapus data presensi *face recognition*
11. Visualisasi data kehadiran dan ketidakhadiran pegawai
12. *Export file* pdf, excel, dan *print*

Operator:

1. Menampilkan data operator dengan *datatable*
2. Mengedit profil data operator
3. konfirmasi data kehadiran pegawai
4. visualisasi data status kehadiran pegawai
5. Menampilkan, mengubah, menambah, dan menghapus data pegawai

4.6 PEMBAHASAN

Aplikasi presensi *online* berbasis web pada studi kasus Diskominfo Kabupaten Mempawah dibangun dengan bahasa pemrograman Python yaitu *framework* django berbasis web. Aplikasi ini dapat membantu pihak instansi khususnya bagian pengelolaan data presensi pegawai. Selain membantu pihak instansi aplikasi ini juga membantu staff pegawai dalam melihat data presensi, melihat data keseluruhan kehadiran maupun ketidakhadiran pegawai dan melakukan pelaporan kepada pihak instansi.

Secara garis besar, bab ini akan menjelaskan deskripsi penelitian yaitu kelebihan sistem presensi yaitu mudah digunakan, minim kontak fisik, biaya pengembangan sistem relatif kecil, meminimalkan tindak kecurangan pada saat presensi. Sedangkan kekurangan sistem yaitu masih ada beberapa fitur yang belum dilengkapi seperti *notification* pada operator.

PERPUSTAKAAN
JENDERAL ACHMAD YANI
UNIVERSITAS YOGYAKARTA