

BAB 4

HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

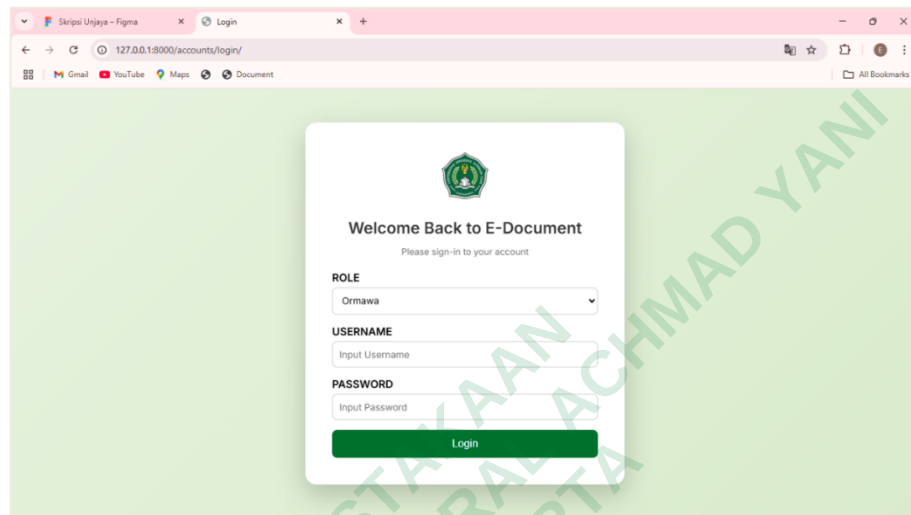
Sistem informasi manajemen dokumen ormawa berbasis web merupakan sistem yang digunakan untuk mengatasi berbagai permasalahan dalam pengelolaan surat di lingkungan Ormawa Universitas Jenderal Achmad Yani Yogyakarta (Unjaya). Sistem yang dibangun berhasil mengotomatisasi penomoran dokumen guna menghindari kesalahan dan duplikasi, menyediakan fitur pencarian dan pelacakan dokumen yang efisien, serta meningkatkan keamanan melalui mekanisme autentikasi dan kontrol akses *user*. Selain itu, sistem juga dilengkapi dengan fitur pengiriman surat digital antar Ormawa dan *dashboard* laporan bulanan. Berdasarkan hasil pengujian User Acceptance Test (UAT) yang dilakukan terhadap 25 responden dengan 20 pertanyaan, diperoleh persentase tingkat kepuasan sebesar 90.66% pengguna puas terhadap semua fitur yang disediakan oleh sistem. Hasil ini menunjukkan bahwa sistem telah memenuhi kebutuhan pengguna dengan sangat baik dan layak untuk diimplementasikan secara penuh di lingkungan Ormawa Unjaya.

4.2 IMPLEMENTASI DESAIN ANTARMUKA

Interface atau antarmuka adalah bagian dari sistem yang menjadi penghubung antara *user* dengan aplikasi. *Interface* dirancang untuk memudahkan interaksi *user* dalam mengakses, mengelola, dan memahami fungsi-fungsi yang ada di dalam sistem. Pada sistem informasi manajemen dokumen Ormawa berbasis web yang dikembangkan untuk Universitas Jenderal Achmad Yani Yogyakarta (Unjaya), *interface* untuk *user* Ormawa dan *interface* untuk Pembimbing Ormawa pada sistem ini memiliki tampilan yang sama, hanya saja pembimbing bertugas untuk memantau saja, tidak dapat *add*, *edit* dan *delete*. Pembagian ini bertujuan untuk membedakan hak akses, kebutuhan, serta peran masing-masing *user* dalam pengelolaan dokumen.

4.2.1 Implementasi Halaman *Login*

Gambar 4.1 merupakan implementasi halaman *login* yang digunakan untuk mengakses sistem. Pada halaman ini, *user* terlebih dahulu memilih peran sebagai Ormawa atau Pembimbing, kemudian memasukkan nama *user* dan kata sandi yang sesuai untuk dapat masuk ke dalam sistem.



Gambar 4.1 Halaman *Login*

Kode di bawah ini digunakan untuk mengatur alur proses *login*. Potongan kode dibawah ini menangani proses login untuk pengguna dengan peran "ormawa". Sistem akan mencari data Ormawa yang sesuai dengan username dan masih aktif. Jika ditemukan, sistem memverifikasi kecocokan password. Jika password benar, data pengguna disimpan dalam session dan diarahkan ke halaman home. Jika password salah, muncul pesan "Invalid username or password". Jika akun tidak ditemukan, ditampilkan pesan "Ormawa account not found".

```
if role == 'ormawa':
    try:
        user= Ormawa.objects.get
            (ormawa_code=username, is_active=True)
        if check_password(password, user.password):
            request.session['role'] = 'ormawa'
            request.session['ormawa_id'] = user.id
            request.session['nama_ormawa']=user.nama_ormawa
            request.session['ormawa_code']=user.ormawa_code
            request.session['photo_url']= user.photo.url
            if user.photo else ''
            return redirect('home')
        else:
            return redirect('home')
```

```

        messages.error(request, 'Invalid username
        or password')
    except Ormawa.DoesNotExist:
        messages.error(request, 'Ormawa account
        not found')

```

Potongan kode ini menangani proses login untuk pengguna dengan peran "pembimbing". Sistem akan mencari data dari model PembimbingOrmawa berdasarkan username dan status aktif. Jika ditemukan dan password cocok, data pembimbing disimpan dalam session dan pengguna diarahkan ke halaman home. Jika password salah, muncul pesan "Invalid username or password". Jika akun tidak ditemukan, tampil pesan "Account not found". Jika tidak ada peran yang cocok, form login akan ditampilkan kembali.

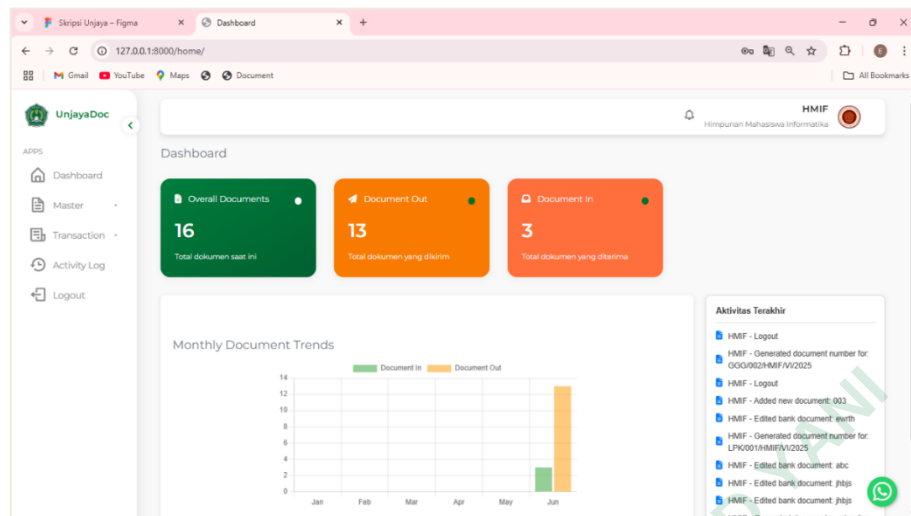
```

elif role == 'pembimbing':
    try:
        pembimbing= PembimbingOrmawa.objects.get
        (username=username, is_active=True)
        if check_password(password,pembimbing.password):
            request.session['role'] = 'pembimbing'
            request.session['pembimbing_id']=
            pembimbing.id
            request.session['pembimbing_nama']=
            pembimbing.nama
            return redirect('home')
        else:
            messages.error(request, 'Invalid username
            or password')
    except PembimbingOrmawa.DoesNotExist:
        messages.error(request, 'Account not found')

```

4.2.2 Implementasi Halaman *Dashboard*

Gambar 4.2 merupakan tampilan halaman *dashboard* yang dapat diakses oleh *user* dengan peran Ormawa maupun Pembimbing. Halaman ini menyajikan informasi sekilas mengenai total data dokumen masuk dan keluar, grafik jumlah dokumen per bulan, serta riwayat aktivitas terakhir *user*.



Gambar 4.2 Halaman *Dashboard*

Potongan kode ini digunakan untuk menyediakan data yang dibutuhkan halaman *dashboard* secara dinamis berdasarkan Ormawa yang sedang *login*. Potongan kode ini digunakan untuk menghitung jumlah dokumen terkait ormawa berdasarkan ID-nya (*ormawa_id*). (*document_in_count*) menghitung jumlah dokumen masuk yang diterima oleh ormawa. (*document_out_count*) menghitung jumlah dokumen keluar yang dikirim oleh ormawa. Kedua jumlah tersebut dijumlahkan untuk mendapatkan total semua dokumen (*total_documents*). Terakhir, (*bank_document_count*) menghitung jumlah dokumen yang disimpan dalam bank dokumen milik ormawa tersebut.

```
document_in_count = IncomingDocument.objects.filter
(receiver_id=ormawa_id).count()
document_out_count =
IncomingDocument.objects.filter(sender_id=ormawa_id).count()
total_documents = document_in_count + document_out_count
bank_document_count = BankDocument.objects.filter
(ormawa_id=ormawa_id).count()
```

Kode ini menghitung jumlah dokumen masuk dan keluar selama 6 bulan terakhir berdasarkan *ormawa_id*. Setiap bulan dihitung jumlah dokumen yang diterima (*receiver_id*) dan dikirim (*sender_id*), lalu disimpan dalam list untuk analisis atau grafik bulanan. Berikut potongan kodenya:

```
now = timezone.now()
this_year = now.year
```

```

this_month = now.month
months = []
document_in_per_month = []
document_out_per_month = []
for i in range(6):
    month = this_month - 5 + i
    year = this_year
    if month <= 0:
        month += 12
        year -= 1
    start = datetime(year, month, 1)
    end = datetime(year + 1, 1, 1)
    if month == 12 else datetime(year, month + 1, 1)
    months.append(start.strftime('%b'))
    document_in_per_month.append(
        IncomingDocument.objects.filter(receiver_id=ormawa_id,
        received_date__range=(start, end)).count())
    document_out_per_month.append(
        IncomingDocument.objects.filter(sender_id=ormawa_id,
        received_date__range=(start, end)).count())

```

Kode ini mengambil 10 data laporan resmi (DataOfficialReport) terbaru milik ormawa berdasarkan ormawa_id, diurutkan dari yang terbaru (created_date secara menurun).

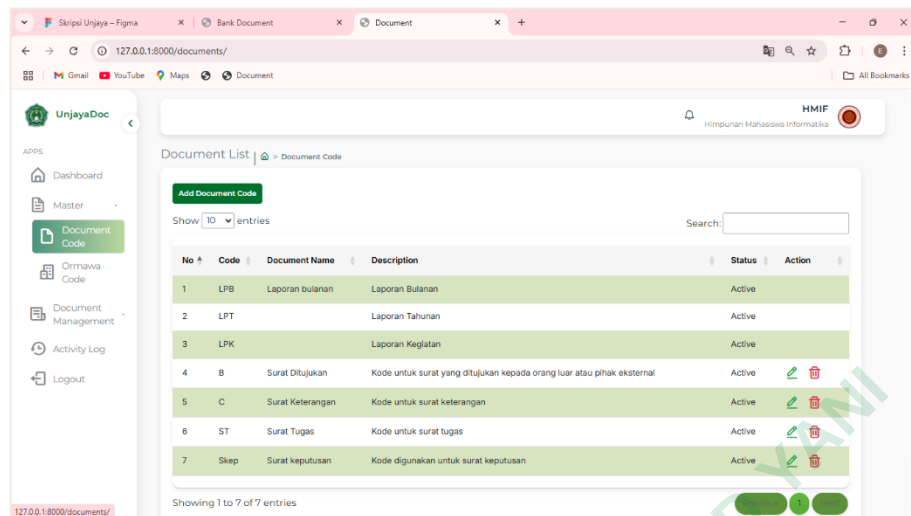
```

official_reports = DataOfficialReport.objects.filter
(ormawa_id=ormawa_id).order_by('-created_date')[:10]

```

4.2.3 Impelentasi Halaman *Document Code*

Gambar 4.3 merupakan tampilan halaman *Document Code* yang menampilkan daftar kode dokumen yang berlaku pada Ormawa tersebut. Halaman ini hanya dapat diakses oleh *user* dengan peran Ormawa dan memiliki fitur untuk melakukan proses CRUD (*Create, Read, Update, dan Delete*).



Gambar 4.3 Halaman *Document Code*

Potongan kode berikut digunakan untuk mengambil dan menampilkan data dokumen tipe berdasarkan kode Ormawa yang sedang aktif dalam sesi *login*. Sistem akan menampilkan *document type* yang dibuat oleh admin atau yang dibuat oleh ormawa yang ada dalam *session*. Setiap dokumen akan dikumpulkan dan disusun dalam bentuk daftar berisi informasi seperti nomor, kode dokumen, nama, deskripsi, status aktif atau tidak, siapa yang membuat, dan kode ormawa yang sedang login.

```
def document_data(request):
    ormawa_code = request.session.get('ormawa_code')

    if not ormawa_code:
        return JsonResponse({'data': []})

    documents = Document_Type.objects.filter(
        Q(created_by='unjaya') | Q(created_by=ormawa_code)
    )
    data = []
    for i, doc in enumerate(documents, start=1):
        data.append({
            'no': i,
            'id': doc.id,
            'code': doc.document_code,
            'name': doc.document_name,
            'description': doc.description,
            'status': 'Active' if doc.is_active else 'Inactive',
            'created_by': doc.created_by,
            'current_user': ormawa_code,
        })
    return JsonResponse({'data': data})
```

Gambar 4.4 merupakan tampilan halaman *Add Document Code* yang digunakan oleh *user* Ormawa untuk menambahkan kode dokumen baru. Pada halaman ini, *user* diwajibkan mengisi seluruh isian (*field*) yang tersedia agar proses penambahan data dapat dilakukan.

Gambar 4.4 *Add Document Code*

Potongan kode berikut digunakan untuk memproses penambahan data pada halaman *Add Document Type*. Sistem mengambil data dari form seperti kode dokumen, nama dokumen, dan deskripsi. Jika kode dokumen tidak diisi, akan muncul pesan kesalahan dan form ditampilkan ulang beserta data yang sudah diisi sebelumnya. Jika semua data valid, sistem mengambil data ormawa yang sedang login, lalu membuat dan menyimpan dokumen baru ke dalam database dengan status aktif dan menyimpan siapa pembuatnya.

```
def add_document(request):
    if request.method == 'POST':
        ormawa_id = request.session.get('ormawa_id')
        document_code = request.POST.get('document_code')
        document_name = request.POST.get('document_name')
        description = request.POST.get('description')
        error = None
        if not document_code:
            error = "Document Code is required."

        if error:
            return render(request, 'add_document.html', {
                'error': error,
                'document_code': document_code,
                'document_name': document_name,
```

```

        'description': description})
ormawa_code = request.session.get('ormawa_code')
current_ormawa = Ormawa.objects.get
(id=ormawa_id, is_active=True)
document = Document_Type(document_code=document_code,
document_name=document_name,
description=description,
is_active=True,
created_by=ormawa_code,
updated_by=None,
updated_date=None,)
document.save()

```

Gambar 4.5 merupakan tampilan halaman *Edit Document*, yang digunakan oleh *user* Ormawa untuk mengubah data kode dokumen yang telah dibuat sebelumnya. Seluruh data yang telah tersimpan akan ditampilkan kembali pada formulir *input* untuk dapat *dieit* sesuai kebutuhan

Gambar 4.5 Halaman *Edit Document Code*

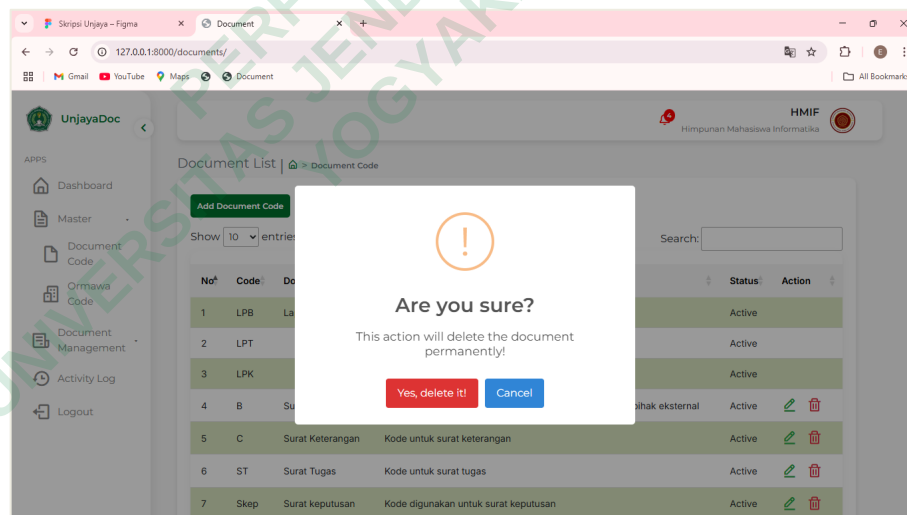
Potongan kode berikut digunakan untuk memproses *edit* data dokumen pada halaman *edit document type*. Fungsi edit document ini digunakan untuk mengubah data dokumen yang sudah ada. Pertama, sistem mengambil dokumen berdasarkan *primary key* (ID dokumen). Jika pengguna menekan tombol simpan, maka data baru seperti kode dokumen, nama, deskripsi, dan status aktif akan diambil dari form. Data tersebut kemudian diperbarui pada dokumen yang dipilih. Informasi siapa yang terakhir mengubah juga dicatat berdasarkan ormawa yang sedang login. Setelah itu, perubahan disimpan ke database.


```
def edit_document(request, pk):
    document = get_object_or_404(Document_Type, pk=pk)

    if request.method == 'POST':
        document_code = request.POST.get('code')
        document_name = request.POST.get('document_name')
        ormawa_id = request.session.get('ormawa_id')
        description = request.POST.get('description')
        is_active = True if request.POST.get('status')
        == 'on' else False
        document.document_code = document_code
        document.document_name = document_name
        document.description = description
        document.is_active = is_active
        document.updated_by = request.session.get
        ('ormawa_code', 'anonymous')
        document.save()

    current_ormawa = Ormawa.objects.get
    (id=ormawa_id, is_active=True)
```

Gambar 4.6 merupakan tampilan saat *user* memilih opsi untuk menghapus *document code*. Sebelum proses penghapusan dilakukan, sistem akan menampilkan notifikasi konfirmasi menggunakan *SweetAlert* untuk memastikan bahwa *user* benar-benar ingin menghapus data tersebut.



Gambar 4.6 Delete Document Code

Potongan kode berikut digunakan untuk menghapus data dokumen pada halaman *Document Type*. Fungsi ini akan memastikan bahwa hanya dokumen yang dibuat oleh Ormawa yang sedang *login* yang dapat dihapus, sehingga menjaga validitas dan keamanan data.

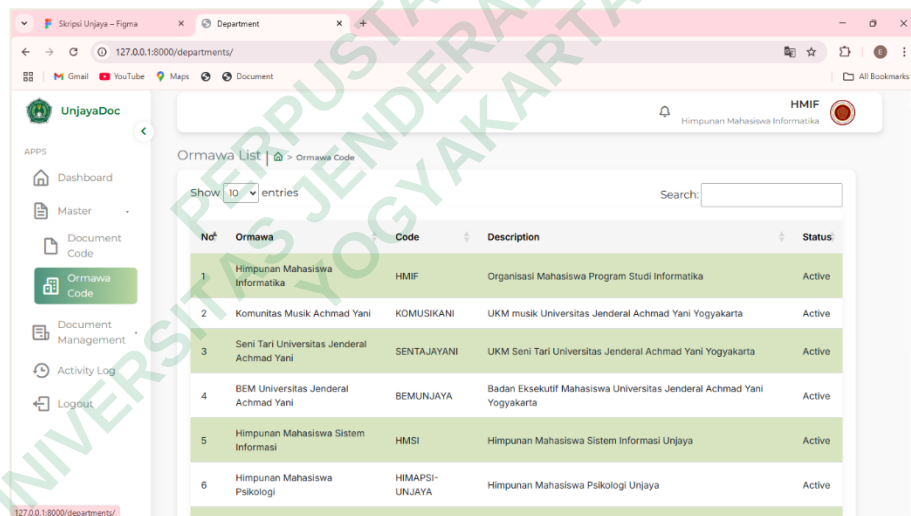
```
def delete_document(request, pk):
    ormawa_code = request.session.get('ormawa_code')
    ormawa_id = request.session.get('ormawa_id')
    document = get_object_or_404(Document_Type, pk=pk)

    if document.created_by == ormawa_code:
        document.delete()

    current_ormawa = Ormawa.objects.get(id=ormawa_id,
is_active=True)
```

4.2.4 Impelentasi Halaman Ormawa Code

Gambar 4.7 merupakan tampilan halaman *Ormawa Code* yang menampilkan data-data Ormawa di lingkungan Universitas Jenderal Achmad Yani Yogyakarta (Unjaya). Halaman ini digunakan ketika *user* ingin mengirimkan pesan ke Ormawa tujuan dengan menggunakan kode Ormawa yang tersedia. Seluruh data Ormawa pada halaman ini di *input* oleh admin, sehingga *user* tidak memiliki akses untuk *add*, *edit*, dan *delete* data tersebut.



No	Ormawa	Code	Description	Status
1	Himpunan Mahasiswa Informatika	HMIF	Organisasi Mahasiswa Program Studi Informatika	Active
2	Komunitas Musik Achmad Yani	KOMUSIKANI	UKM musik Universitas Jenderal Achmad Yani Yogyakarta	Active
3	Seni Tari Universitas Jenderal Achmad Yani	SENTAJAYANI	UKM Seni Tari Universitas Jenderal Achmad Yani Yogyakarta	Active
4	BEM Universitas Jenderal Achmad Yani	BEMUNJAYA	Badan Eksekutif Mahasiswa Universitas Jenderal Achmad Yani Yogyakarta	Active
5	Himpunan Mahasiswa Sistem Informasi	HMSI	Himpunan Mahasiswa Sistem Informasi Unjaya	Active
6	Himpunan Mahasiswa Psikologi	HIMAPSI-UNJAYA	Himpunan Mahasiswa Psikologi Unjaya	Active

Gambar 4.7 Halaman Ormawa Code

Potongan kode berikut digunakan untuk menampilkan data Ormawa yang aktif pada halaman *Ormawa Code*. Fungsi `departments_data` ini digunakan untuk menampilkan daftar ormawa yang aktif. Sistem mengambil semua data ormawa dengan status aktif, lalu menyusunnya menjadi daftar berisi nomor, nama ormawa, kode ormawa, deskripsi (jika ada), dan status aktif/tidak. Data ini kemudian dikirim dalam format JSON, biasanya untuk ditampilkan di tabel pada halaman web.

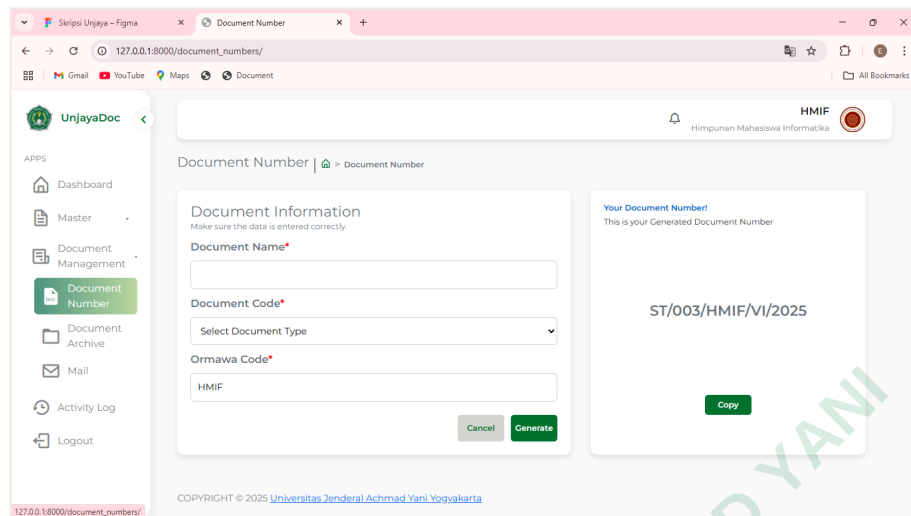
```
def departments_data(request):
    ormawa_qs = Ormawa.objects.filter(is_active=True)
    data = []
    for index, ormawa in enumerate(ormawa_qs, start=1):
        data.append({
            'no': index,
            'nama_ormawa': ormawa.nama_ormawa,
            'ormawa_code': ormawa.ormawa_code,
            'description': ormawa.description or '',
            'is_active': 'Active' if ormawa.is_active
            else 'Inactive', })
    return JsonResponse({'data': data})
```

4.2.5 Impelentasi Halaman *Document Numbering*

Gambar 4.8 merupakan tampilan halaman *Document Number* yang digunakan oleh *user* Ormawa untuk menginputkan keterangan dokumen, nama dokumen, serta memilih kode dokumen yang sesuai. Seluruh isian (*field*) pada halaman ini wajib diisi agar proses penyimpanan data dapat dilakukan.

Gambar 4.8 *Generate Document Number*

Gambar 4.9 menampilkan kode dokumen yang berhasil dibuat oleh sistem. Setelah proses generate selesai, kode dokumen tersebut secara otomatis ditambahkan dan ditampilkan pada halaman *Document Archive*.



Gambar 4.9 Hasil *Generate Number*

Potongan kode berikut digunakan untuk menangani proses pembuatan nomor dokumen secara otomatis pada halaman *Document Number*. Kode ini digunakan untuk membuat nomor urut dokumen berdasarkan jenis dokumen dan ormawa yang sedang login. Jika `document_type` diisi, sistem akan mencari nomor dokumen terakhir yang dibuat oleh ormawa tersebut, lalu mengambil `sequence_number` (nomor urut terakhir). Jika belum ada, maka dimulai dari 0. Setelah itu, nomor urut baru dibuat dengan format tiga digit (misalnya 001, 002, dll). Sistem juga mengubah bulan saat ini ke format angka Romawi (misalnya 07 menjadi VII) menggunakan dictionary `bulan_romawi`, lalu mengambil bulan dan tahun saat ini untuk keperluan format nomor dokumen lengkap.

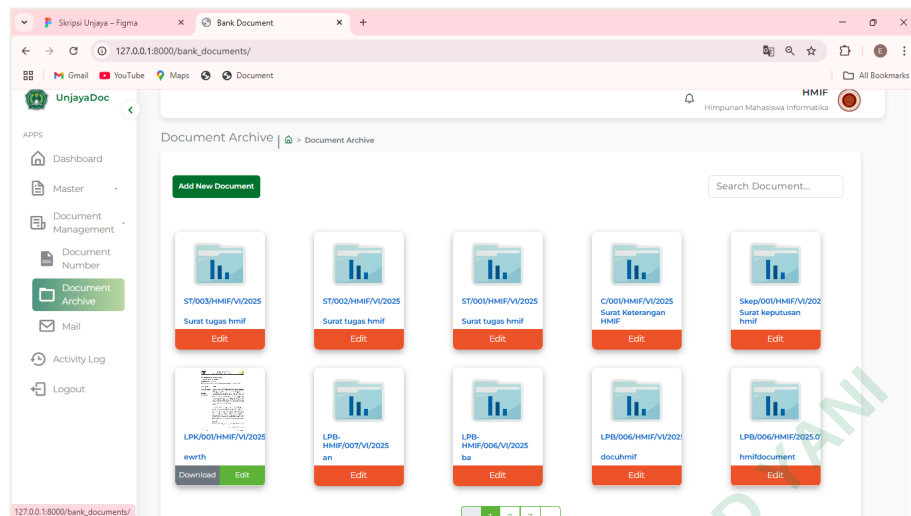
```
if document_type:
    last_doc = DocumentNumber.objects.filter(
        document_type=document_type,
        ormawa=current_ormawa).order_by('-id').first()
    last_seq = int(last_doc.sequence_number)
    if last_doc else 0
    new_seq = f"{last_seq + 1:03}"
    bulan_romawi = {
        '01': 'I', '02': 'II', '03': 'III', '04': 'IV',
        '05': 'V', '06': 'VI', '07': 'VII', '08': 'VIII',
        '09': 'IX', '10': 'X', '11': 'XI', '12': 'XII'}
    now_date = timezone.now()
    bulan = now_date.strftime('%m')
    bulan_roman = bulan_romawi.get(bulan, bulan)
    tahun = now_date.strftime('%Y')
```

Potongan kode ini digunakan untuk menghasilkan nomor dokumen otomatis berdasarkan format yang telah ditentukan oleh ormawa. Jika ormawa memiliki format sendiri (`current_ormawa.format`), sistem akan mengganti bagian-bagian format tersebut seperti `{document_code}`, `{sequence}`, `{document_name}`, `{bulan}`, dan `{tahun}` dengan data yang sesuai, misalnya kode dokumen, nomor urut, nama dokumen, bulan Romawi, dan tahun sekarang. Namun, jika tidak ada format khusus, sistem akan menggunakan format standar yaitu: `kode_dokumen/nomor_urut/kode_ormawa/bulan/tahun`.

```
if current_ormawa.format:
    generated_number = current_ormawa.format
    generated_number = generated_number.replace
    ('{document_code}', document_type.document_code)
    generated_number = generated_number.replace
    ('{sequence}', new_seq)
    generated_number = generated_number.replace
    ('{document_name}', document_type.document_name or
    document_type.document_code)
    generated_number = generated_number.replace
    ('{bulan}', bulan_roman)
    generated_number = generated_number.replace
    ('{tahun}', tahun)
else:
    generated_number = f"{document_type.document_code}/{
    new_seq}/{current_ormawa.ormawa_code}/
    {bulan_roman}/{tahun}"
```

4.2.6 Implementasi Halaman *Document Archive*

Gambar 4.10 merupakan tampilan halaman *Document Archive* yang menampilkan data dokumen dengan nomor yang telah digenerasikan. Apabila tombol *edit* berwarna oranye dan tidak terdapat thumbnail, maka dokumen tersebut hanya berisi nomor dan nama dokumen tanpa disertai berkas. Sebaliknya, jika dokumen telah dilengkapi dengan berkas, maka kartu dokumen akan menampilkan thumbnail dari halaman pertama berkas pdf dan tombol download untuk mengunduhnya. Pada halaman ini juga tersedia tombol *add new document* untuk menambahkan dokumen baru serta fitur *search* untuk memudahkan pencarian data.



Gambar 4.10 Document Archive

Kode dibawah ini digunakan untuk menampilkan data dokumen pada halaman Document Archive berdasarkan identitas Ormawa yang sedang *login*. Fungsi ini digunakan untuk menampilkan daftar dokumen milik ormawa yang sedang *login*. Sistem akan mengambil data dokumen dari tabel Bank Document berdasarkan `ormawa_id`, diurutkan dari yang terbaru.

```
def bank_documents(request):
    ormawa_id = request.session.get('ormawa_id')

    if not ormawa_id:
        return redirect('login')

    query = request.GET.get('q', '').strip()

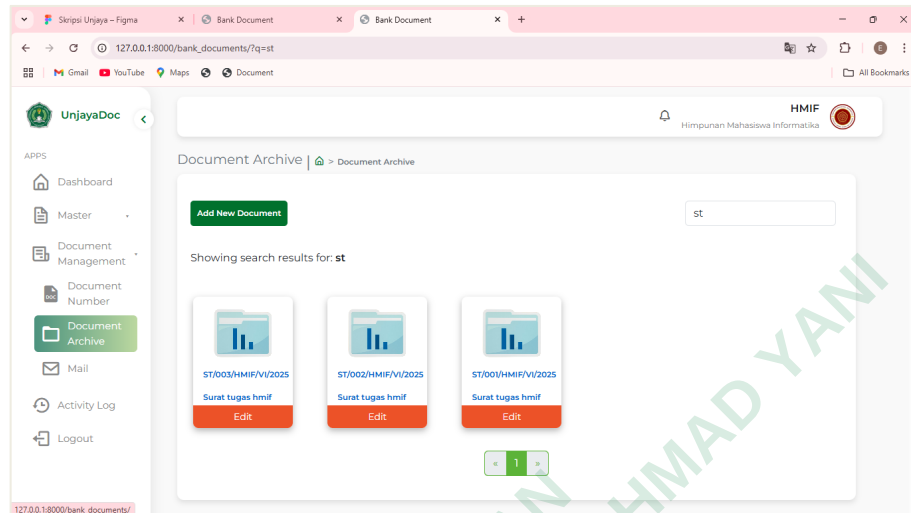
    document_list = BankDocument.objects.filter(
        ormawa_id=ormawa_id).order_by('-id')
```

Dokumen yang ditampilkan dibagi per halaman (10 dokumen per halaman) menggunakan paginator. Berikut potongan kodenya:

```
paginator = Paginator(document_list, 10)
page_number = request.GET.get('page')
page_obj = paginator.get_page(page_number)
```

Gambar 4.11 merupakan tampilan ketika *user* menggunakan kolom *search* untuk mencari dokumen. Fitur ini memudahkan *user* dalam menemukan dokumen tertentu, terutama ketika jumlah dokumen sudah banyak. *User* cukup mengetikkan

nama atau nomor dokumen, maka sistem akan secara otomatis menampilkan hasil pencarian yang sesuai.



Gambar 4.11 *Search Document*

Potongan kode berikut digunakan untuk menangani fitur pencarian pada halaman *Document Archive*. Apabila *user* memasukkan kata kunci pencarian, maka sistem akan memfilter data berdasarkan kecocokan sebagian teks (*partial match*) terhadap nama dokumen atau nomor dokumen.

```
if query:
    document_list = document_list.filter(
        Q(document_name__icontains=query) |
        Q(document_number__icontains=query)
    )
else:
    query = ''
```

Gambar 4.12 merupakan tampilan ketika *user* menekan tombol *edit* berwarna oranye. Halaman ini akan menampilkan formulir isian (*input form*) yang digunakan untuk melengkapi data dokumen setelah nomor dokumen sebelumnya berhasil di *generate*. Melalui halaman ini, *user* dapat menambahkan informasi tambahan serta mengunggah berkas dokumen yang bersangkutan.

UnjayaDoc

HMIF
Himpunan Mahasiswa Informatika

APPS

- Dashboard
- Master
- Document Management
- Document Number
- Document Archive
- Mail
- Activity Log
- Logout

Edit Document Archive | > Document Archive > Edit Document Archive

Document Information
Make sure the data is entered correctly.

Document Number *

ST/003/HMIF/VI/2025

Document Name *

Surat tugas hmif

Document Description *

Upload Thumbnail

Choose File No file chosen

Upload File (PDF) *

Choose File No file chosen

Cancel Save

Gambar 4.12 *Edit Blank Document*

Gambar 4.13 merupakan tampilan halaman ketika *user* menekan tombol *edit* berwarna hijau, yang menandakan bahwa dokumen tersebut telah dilengkapi dengan data sebelumnya. Halaman ini digunakan apabila terdapat kesalahan *input* pada data yang telah dimasukkan. Seluruh data sebelumnya akan ditampilkan kembali dalam formulir agar dapat diperbaiki atau diperbarui oleh *user*

UnjayaDoc

HMIF
Himpunan Mahasiswa Informatika

APPS

- Dashboard
- Master
- Document Management
- Document Number
- Document Archive
- Mail
- Activity Log
- Logout

Edit Document Archive | > Document Archive > Edit Document Archive

Document Information
Make sure the data is entered correctly.

Document Number *

LPK/001/HMIF/VI/2025

Document Name *

ewrth

Document Description *

asdfgth

Upload Thumbnail

Choose File No file chosen

Upload File (PDF) *

Current file: ewrth_file.pdf

Choose File No file chosen

Cancel Save

Gambar 4.13 *Edit Existing Document*

Potongan kode berikut digunakan untuk mengelola proses *edit* data dokumen pada halaman *Document Archive*. Fungsi `edit_bank_document` digunakan untuk mengedit data dokumen milik ormapa yang tersimpan di Bank Dokumen.

Pertama, sistem memastikan pengguna (ormawa) sudah login, lalu mengambil data dokumen berdasarkan primary key (ID). Jika form dikirim deskripsi dokumen akan diperbarui.

```
def edit_bank_document(request, pk):
    document = get_object_or_404(BankDocument, pk=pk)
    ormawa_id = request.session.get('ormawa_id')
    if not ormawa_id:
        return redirect('login')
    curent_ormawa = Ormawa.objects.get(id=ormawa_id,
    is_active=True)

    if request.method == 'POST':
        document.document_description =
        request.POST.get('description')
```

Jika ada file dokumen yang diunggah, nama file dibersihkan dari spasi, lalu diubah menjadi format NamaDokumen_file.pdf dan disimpan di folder bank_documents/files/. Informasi nama file, ukuran, dan lokasi penyimpanan disimpan ke dalam objek dokumen. Berikut potongan kodenya:

```
if 'file_upload' in request.FILES:
    uploaded_file = request.FILES['file_upload']
    document_name_clean =
    document.document_name.replace(' ', '_')
    new_file_name = f"{document_name_clean}_file.pdf"

    fs = FileSystemStorage(
    location=os.path.join(settings.MEDIA_ROOT,
    'bank_documents/files/'))
    filename = fs.save(new_file_name, uploaded_file)

    document.file_name = new_file_name
    document.file_size = uploaded_file.size
    document.file_path = f'bank_documents/
    files/{filename}'
```

Jika pengguna mengunggah thumbnail baru, file tersebut disimpan sebagai thumbnail dokumen, apabila pengguna tidak mengunggah thumbnail, maka sistem akan otomatis membuat thumbnail dari file PDF yang diunggah menggunakan fungsi generate_thumbnail_from_pdf. Berikut potongan kodenya:

```
if 'thumbnail' in request.FILES:
    document.document_thumbnail =
    request.FILES['thumbnail']

def generate_thumbnail_from_pdf
(pdf_path, output_dir, filename):
```

```

try:
    pages = convert_from_path(pdf_path,
                              first_page=1, last_page=1, dpi=150)
    if pages:
        thumbnail_name = filename.replace
        ('.pdf', '_thumb.jpg')
        thumbnail_path = os.path.join
        (output_dir, thumbnail_name)
        pages[0].save(thumbnail_path, 'JPEG')
        return f'bank_documents/thumbnails/{thumbnail_name}'
except Exception as e:
    print(f"Error generating thumbnail: {e}")
return None

```

Gambar 4.14 merupakan tampilan halaman *Add Document* yang digunakan oleh *user* Ormawa untuk menambahkan dokumen secara langsung tanpa melalui proses *generate* nomor dokumen. Menu ini disediakan sebagai alternatif bagi *user* yang ingin mengarsipkan dokumen dengan nomor yang telah ditentukan sebelumnya atau tanpa menggunakan format penomoran otomatis

Gambar 4.14 *Add Document Archive*

Potongan kode berikut digunakan untuk menangani proses penambahan dokumen baru pada halaman Document Archive. Kode ini digunakan untuk menyimpan dokumen baru ke dalam Bank Dokumen. Saat form dikirim, sistem mengambil data seperti nomor dokumen, nama, deskripsi, thumbnail (jika ada), dan file yang diunggah. Jika salah satu field penting tidak diisi (kecuali thumbnail), maka akan muncul pesan error dan form ditampilkan kembali. Kemudian, sistem mengambil salah satu data jenis dokumen (*Document_Type*) untuk disisipkan. File

dokumen disimpan di folder `bank_documents/files/` dengan nama yang telah dibersihkan.

```
if request.method == 'POST':
    document_number = request.POST.get('document_number')
    document_name = request.POST.get('document_name')
    description = request.POST.get('description')
    thumbnail = request.FILES.get('thumbnail')
    file_upload = request.FILES.get('file_upload')

    if not all([document_number, document_name,
               description, file_upload]):
        return render(request, 'add_bank_document.html', {
            'error': 'Semua field harus diisi kecuali
                  thumbnail.'})

    from Documents.models import Document_Type
    document_type = Document_Type.objects.first()
    fs_file = FileSystemStorage(location=os.path.join(
        settings.MEDIA_ROOT, 'bank_documents/files/'))
    cleaned_name = document_name.replace(' ', '_')
    saved_file_name = f"{cleaned_name}_file.pdf"
    saved_file_path = fs_file.save
    (saved_file_name, file_upload)
```

Jika thumbnail tidak diunggah, maka sistem otomatis membuat thumbnail dari file PDF menggunakan fungsi `generate_thumbnail_from_pdf`. Jika thumbnail diunggah, maka digunakan langsung. Berikut potongan kodenya:

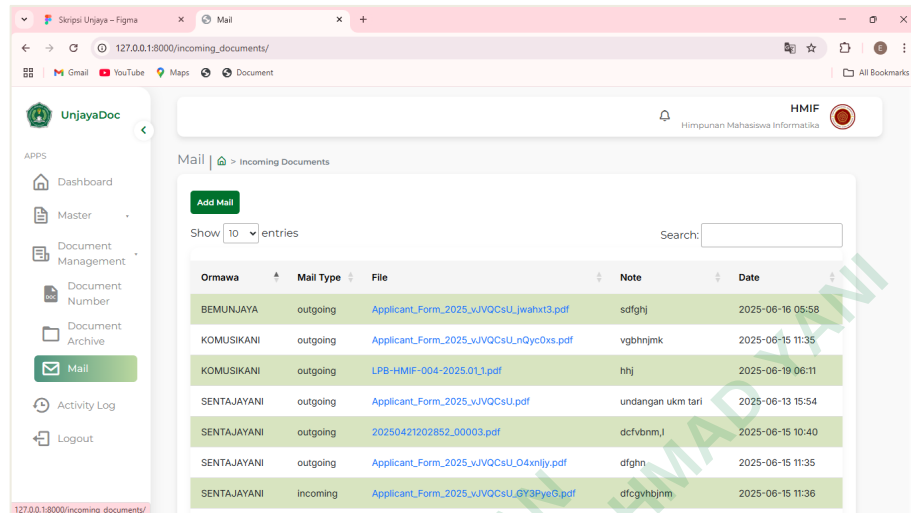
```
if not thumbnail:
    pdf_full_path = os.path.join(settings.MEDIA_ROOT,
                                  'bank_documents/files/', saved_file_path)

    thumbnail_rel_path = generate_thumbnail_from_pdf(
        pdf_full_path, os.path.join(settings.MEDIA_ROOT,
                                      'bank_documents/thumbnails/'), saved_file_name)
else:
    thumbnail_rel_path = thumbnail
```

4.2.7 Implementasi Halaman *Mail*

Gambar 4.15 merupakan tampilan halaman *Mail* yang menyajikan data pesan masuk (*incoming*) dan pesan keluar (*outgoing*). Halaman ini menampilkan informasi penting seperti nama Ormawa pengirim atau penerima, jenis pesan (*Mail Type*), nama file dokumen, catatan (*Note*), dan tanggal pengiriman atau penerimaan pesan. *User* dapat mengklik tautan pada kolom *file* untuk melihat atau mengunduh dokumen yang dikirimkan atau diterima. Selain itu, tersedia pula fitur *search* untuk

memudahkan pencarian data serta tombol *add mail* untuk menambahkan data pesan baru.



Gambar 4.15 Halaman *Mail*

Potongan kode berikut digunakan untuk mengimplementasikan fitur pengiriman dan penerimaan dokumen antar Ormawa dalam sistem. Kode ini digunakan untuk menampilkan data dokumen masuk dan keluar untuk ormawa yang sedang login. Pertama, sistem memeriksa apakah pengguna (ormawa) sudah login lewat session. Sistem mencari dokumen yang dikirim atau diterima oleh ormawa tersebut. Kemudian, data seperti kode ormawa lawan (penerima/pengirim), jenis surat, link file, deskripsi, dan tanggal dibuat diformat dan disusun ke dalam daftar dan ditampilkan di halaman *Mail*.

```
@require_GET
def incoming_documents_data(request):
    ormawa_id = request.session.get('ormawa_id')
    if not ormawa_id:
        return JsonResponse({'data': []})

    current_ormawa = get_object_or_404(Ormawa, id=ormawa_id,
                                       is_active=True)

    documents = IncomingDocument.objects.filter(
        Q(sender=current_ormawa) | Q(receiver=current_ormawa)
    ).select_related('sender', 'receiver')

    data = []
    for doc in documents:
        if doc.sender == current_ormawa:
            mail_type = 'outgoing'
```

```

        other_party = doc.receiver
    else:
        mail_type = 'incoming'
        other_party = doc.sender

    data.append({
        'ormawa_code': other_party.ormawa_code
        if other_party else '-',
        'mail_type': mail_type,
        'file': doc.file.url if doc.file else '',
        'main_description': doc.description,
        'created_date': doc.created_date.strftime
        ("%Y-%m-%d %H:%M"),
    })

    return JsonResponse({'data': data})

```

Gambar 4.16 merupakan tampilan halaman *Send Mail Document* yang digunakan oleh *user* Ormawa untuk mengirim dokumen kepada Ormawa lain. Pada halaman ini, *user* harus mengisi formulir pengiriman dokumen dengan memilih penerima (*Receiver*), mengunggah *file* dalam format pdf, mengisi judul dokumen (*Document Title*), serta menuliskan deskripsi dokumen. Seluruh isian pada formulir ini bersifat wajib diisi agar dokumen dapat dikirimkan dengan benar. Setelah seluruh data terisi, *user* dapat menekan tombol *send* untuk mengirim dokumen atau *cancel* untuk membatalkan proses pengiriman.

Gambar 4.16 Add New Mail

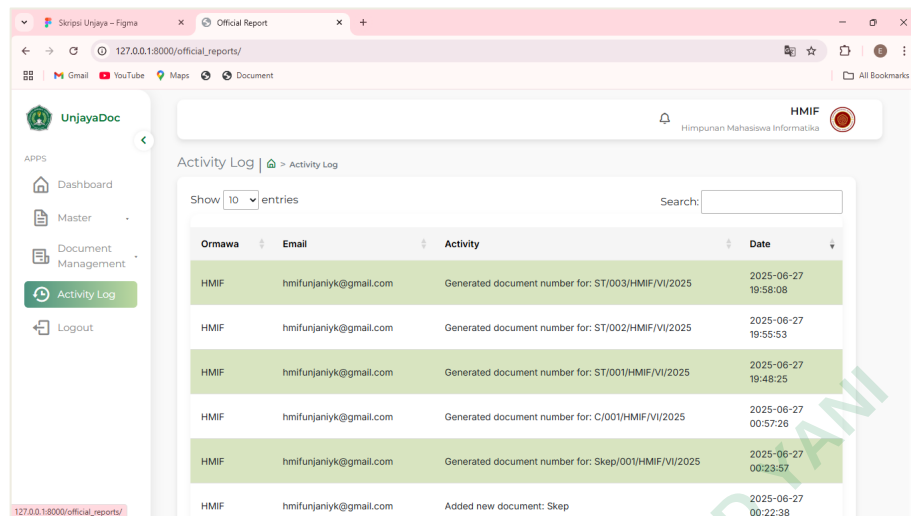
Potongan kode berikut digunakan untuk memproses pengiriman dokumen dari Ormawa pengirim ke Ormawa penerima dalam sistem Fungsi `send_document`

ini digunakan untuk mengirim dokumen antar ormawa. Pertama, sistem mengecek apakah pengguna (ormawa) sudah login melalui session. Jika belum, pengguna akan diarahkan ke halaman login. Setelah login, sistem mengambil data ormawa yang aktif. Jika form dikirim, data dari form `SendDocumentForm` akan diproses, termasuk file yang diunggah. Jika data valid, dokumen tidak langsung disimpan ke database (`commit=False`) agar bisa menambahkan data tambahan, seperti pengirim (`sender`), siapa yang membuat (`created_by`), waktu diterima, dan ukuran file. Setelah itu, dokumen disimpan ke database.

```
def send_document(request):
    ormawa_id = request.session.get('ormawa_id')
    if not ormawa_id:
        return redirect('login')
    current_ormawa = get_object_or_404(Ormawa, id=ormawa_id, is_active=True)
    if request.method == 'POST':
        form = SendDocumentForm(request.POST, request.FILES,
                                ormawa_login=current_ormawa)
        if form.is_valid():
            doc = form.save(commit=False)
            doc.sender = current_ormawa
            doc.created_by = current_ormawa.nama_ormawa
            doc.received_date = timezone.now()
            doc.file_size = doc.file.size
            doc.save()
```

4.2.8 Impelentasi Halaman *Activity log*

Gambar 4.17 merupakan tampilan halaman *Activity Log* yang digunakan untuk memantau seluruh aktivitas *user* di dalam sistem. Halaman ini dapat diakses oleh *user* dengan peran Ormawa maupun Pembimbing. Informasi yang ditampilkan mencakup nama Ormawa, alamat *email*, jenis aktivitas yang dilakukan, serta waktu dan tanggal aktivitas tersebut berlangsung. Fitur ini berguna untuk memastikan transparansi dan jejak audit dari setiap tindakan *user* dalam sistem.



Gambar 4.17 Halaman *Activity log*

Potongan kode berikut digunakan untuk menampilkan data laporan aktivitas *user* yang aktif dari seluruh Ormawa. Data yang ditampilkan meliputi kode Ormawa, *email*, judul aktivitas, serta tanggal dan waktu dibuatnya laporan yang ditampilkan di dalam table.

```
def official_reports_data(request):
    ormawa_id = request.session.get('ormawa_id')

    if not ormawa_id:
        return JsonResponse({'data': []})

    reports = DataOfficialReport.objects.filter(
        is_active=True
    ).select_related('ormawa').order_by('-created_date')

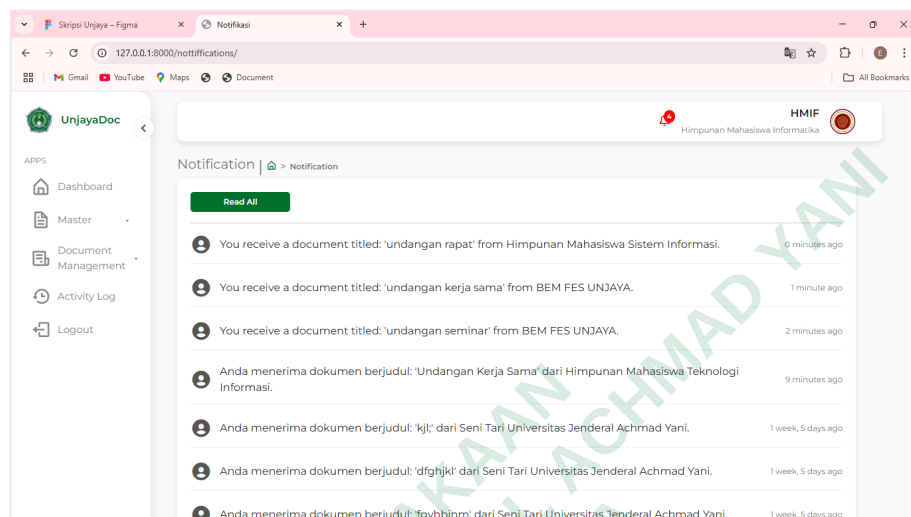
    data = []
    for report in reports:
        data.append({
            'ormawa': report.ormawa.ormawa_code,
            'email': report.ormawa.email,
            'Activity': report.report_title,
            'date': localtime(report.created_date).strftime(
                '%Y-%m-%d %H:%M:%S'),
        })

    return JsonResponse({'data': data})
```

4.2.9 Impelentasi Halaman *Notification*

Gambar 4.18 merupakan tampilan halaman *Notification* yang menampilkan pemberitahuan ketika Ormawa menerima pesan dari Ormawa lain. Ketika terdapat

pesan masuk, sistem akan menampilkan notifikasi berupa banner merah pada ikon notifikasi, dengan angka yang menunjukkan jumlah pesan yang belum dibaca. *User* dapat menekan tombol *read All* untuk menandai seluruh pesan sebagai telah dibaca. Setelah itu, *user* dapat melihat isi pesan secara lengkap melalui halaman *mail*.



Gambar 4.18 Halaman *Notification*

Potongan kode berikut digunakan untuk mengelola notifikasi pada halaman *Notification*. Fungsi *notifications* digunakan untuk menampilkan daftar notifikasi yang diterima oleh Ormawa berdasarkan nama penerima. Fungsi *mark_all_read* digunakan untuk menandai semua notifikasi yang belum dibaca menjadi terbaca, sedangkan *unread_notification_count_ajax* digunakan untuk menghitung jumlah notifikasi yang belum dibaca dan mengirimkannya ke sisi klien secara dinamis melalui permintaan AJAX.

```
def notifications(request):
    ormawa_id = request.session.get('ormawa_id')
    if not ormawa_id:
        return render(request, 'notification.html',
            {'notifications': []})

    current_ormawa = get_object_or_404(Ormawa, id=ormawa_id)
    notifications = Notification.objects.filter(
        recipient=current_ormawa.nama_ormawa
    ).order_by('-created_date')
    return render(request, 'notification.html',
        {'notifications': notifications})

def mark_all_read(request):
    ormawa_id = request.session.get('ormawa_id')
```



```

if not ormawa_id:
    return JsonResponse({'success': False})
current_ormawa = get_object_or_404(Ormawa, id=ormawa_id)

Notification.objects.filter(recipient=current_ormawa.nama_ormawa,
                             is_read=False).update(is_read=True)

return JsonResponse({'success': True})

def unread_notification_count_ajax(request):
    ormawa_id = request.session.get('ormawa_id')
    if not ormawa_id:
        return JsonResponse({'count': 0})

```

4.3 IMPLEMENTASI BASIS DATA

Implementasi basis data dilakukan dengan menggunakan *framework* Django yang terintegrasi dengan sistem manajemen basis data MySQL. Seluruh struktur basis data yang telah dirancang sebelumnya direalisasikan dalam bentuk model Django, yang kemudian secara otomatis dikonversi menjadi tabel dalam basis data melalui proses migrasi. Proses ini mencakup pembuatan model untuk setiap entitas, penentuan tipe data, serta pengaturan relasi antar tabel seperti *foreign key*. Implementasi ini memastikan bahwa data yang digunakan dalam aplikasi, seperti data *user*, dokumen, jenis dokumen, dan aktivitas, dapat disimpan dan dikelola secara terstruktur dan konsisten dalam basis data MySQL. Basis data pada sistem ini diberi nama *eunjayadocument_db* dengan 8 tabel utama sesuai dengan kebutuhan fungsional sistem. Gambar 4.19 menampilkan struktur *database eunjayadocument_db*.

Table	Action	Rows	Type	Collation	Size	Overh
accounts_ormawa	★ Browse Structure Search Insert Empty Drop	12	InnoDB	utf8mb4_general_ci	48.0 KIB	
accounts_pembimbingormawa	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	64.0 KIB	
auth_group	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 KIB	
auth_group_permissions	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KIB	
auth_permission	★ Browse Structure Search Insert Empty Drop	68	InnoDB	utf8mb4_general_ci	32.0 KIB	
auth_user	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	32.0 KIB	
auth_user_groups	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KIB	
auth_user_user_permissions	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KIB	
bank_document_bankdocument	★ Browse Structure Search Insert Empty Drop	59	InnoDB	utf8mb4_general_ci	48.0 KIB	
django_admin_log	★ Browse Structure Search Insert Empty Drop	38	InnoDB	utf8mb4_general_ci	48.0 KIB	
django_content_type	★ Browse Structure Search Insert Empty Drop	17	InnoDB	utf8mb4_general_ci	48.0 KIB	
django_migrations	★ Browse Structure Search Insert Empty Drop	54	InnoDB	utf8mb4_general_ci	16.0 KIB	
django_session	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	32.0 KIB	
documents_document_type	★ Browse Structure Search Insert Empty Drop	14	InnoDB	utf8mb4_general_ci	16.0 KIB	
document_number_documentnumber	★ Browse Structure Search Insert Empty Drop	75	InnoDB	utf8mb4_general_ci	48.0 KIB	
incoming_documents_incomingdocument	★ Browse Structure Search Insert Empty Drop	21	InnoDB	utf8mb4_general_ci	48.0 KIB	
notification_notification	★ Browse Structure Search Insert Empty Drop	19	InnoDB	utf8mb4_general_ci	16.0 KIB	
official_report_dataofficialreport	★ Browse Structure Search Insert Empty Drop	257	InnoDB	utf8mb4_general_ci	96.0 KIB	

Gambar 4.19 Implementasi *database eunjayadoc*

4.3.1 Implementasi Tabel Ormawa

Gambar 4.20 merupakan implementasi tabel Ormawa dengan id sebagai *primary key*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1 id	bigint(20)			No	None		AUTO_INCREMENT	Change Drop More
☐	2 nama_ormawa	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐	3 ormawa_code	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐	4 password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐	5 description	longtext	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	6 is_active	tinyint(1)			No	None			Change Drop More
☐	7 created_by	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	8 updated_by	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	9 created_date	datetime(6)			No	None			Change Drop More
☐	10 updated_date	datetime(6)			Yes	NULL			Change Drop More
☐	11 photo	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	12 email	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	13 format	longtext	utf8mb4_general_ci		Yes	NULL			Change Drop More

Gambar 4.20 Tabel *Account* Ormawa

4.3.2 Implementasi *Document Type*

Gambar 4.21 merupakan implementasi tabel *Document Type* dengan id sebagai *primary key*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1 id	bigint(20)			No	None		AUTO_INCREMENT	Change Drop More
☐	2 document_code	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐	3 description	longtext	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	4 is_active	tinyint(1)			No	None			Change Drop More
☐	5 created_by	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	6 updated_by	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	7 created_date	datetime(6)			No	None			Change Drop More
☐	8 updated_date	datetime(6)			Yes	NULL			Change Drop More
☐	9 document_name	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Gambar 4.21 Tabel *Document Type*

4.3.3 Implementasi Tabel Pembimbing Ormawa

Gambar 4.22 merupakan implementasi tabel Pembimbing Ormawa dengan id sebagai *primary key* dan ormawa_id sebagai *foreign key*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1 id	bigint(20)			No	None		AUTO_INCREMENT	Change Drop More
☐	2 nama	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐	3 username	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐	4 email	varchar(254)	utf8mb4_general_ci		No	None			Change Drop More
☐	5 password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐	6 is_active	tinyint(1)			No	None			Change Drop More
☐	7 created_date	datetime(6)			No	None			Change Drop More
☐	8 updated_date	datetime(6)			No	None			Change Drop More
☐	9 ormawa_id	bigint(20)			No	None			Change Drop More

Gambar 4.22 Tabel Pembimbing Ormawa

4.3.4 Implementasi Tabel *Document Number*

Gambar 4.23 merupakan implementasi tabel *Document Number* dengan id sebagai *primary key* dan ormawa_id sebagai *foreign key*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1 id	bigint(20)			No	None		AUTO_INCREMENT	Change Drop More
☐	2 document_name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐	3 generate_document_number	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐	4 sequence_number	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐	5 is_active	tinyint(1)			No	None			Change Drop More
☐	6 is_new_document	tinyint(1)			No	None			Change Drop More
☐	7 document_version	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐	8 is_upgrade	tinyint(1)			No	None			Change Drop More
☐	9 updated_date	datetime(6)			No	None			Change Drop More
☐	10 created_date	datetime(6)			Yes	NULL			Change Drop More
☐	11 document_type_id	bigint(20)			Yes	NULL			Change Drop More
☐	12 ormawa_id	bigint(20)			Yes	NULL			Change Drop More

Gambar 4.23 Tabel *Document Number*

4.3.5 Implementasi Tabel *Bank Document*

Gambar 4.24 merupakan implementasi tabel *Bank Document* digunakan untuk halaman *document archive* dengan id sebagai *primary key*, serta ormawa_id dan document_type_id sebagai *foreign key*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1 id	bigint(20)			No	None		AUTO_INCREMENT	Change Drop More
☐	2 document_name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐	3 document_number	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐	4 file_name	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	5 file_size	bigint(20)			Yes	NULL			Change Drop More
☐	6 file_path	varchar(500)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	7 document_flow	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐	8 is_active	tinyint(1)			No	None			Change Drop More
☐	9 created_by	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐	10 created_date	datetime(6)			No	None			Change Drop More
☐	11 updated_by	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	12 updated_date	datetime(6)			Yes	NULL			Change Drop More
☐	13 document_type_id	bigint(20)			No	None			Change Drop More
☐	14 ormawa_id	bigint(20)			No	None			Change Drop More
☐	15 document_thumbnail	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	16 document_description	longtext	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	17 bankdocument_type	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Gambar 4.24 Tabel *Bank Document*

4.3.6 Implementasi Tabel *Incoming Document*

Gambar 4.24 merupakan implementasi tabel *Incoming Document* digunakan untuk halaman *mail* dengan id sebagai *primary key*, serta receiver_id dan sender_id sebagai *foreign key*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1 id	bigint(20)			No	None		AUTO_INCREMENT	Change Drop More
☐	2 document_title	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐	3 received_date	datetime(6)			No	None			Change Drop More
☐	4 description	longtext	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	5 created_by	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	6 created_date	datetime(6)			No	None			Change Drop More
☐	7 file_size	bigint(20)			Yes	NULL			Change Drop More
☐	8 sender_id	bigint(20)			Yes	NULL			Change Drop More
☐	9 receiver_id	bigint(20)			Yes	NULL			Change Drop More
☐	10 file	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Gambar 4.25 Tabel *Incoming Document*

4.3.7 Implementasi Tabel *Official Report*

Gambar 4.24 merupakan implementasi tabel *Official Report* digunakan untuk halaman *activity log* dengan id sebagai *primary key*, dan ormawa_id sebagai *foreign key*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐ 1	id	bigint(20)			No	None		AUTO_INCREMENT	Change Drop More
☐ 2	report_title	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐ 3	created_by	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐ 4	created_date	datetime(6)			No	None			Change Drop More
☐ 5	updated_by	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐ 6	updated_date	datetime(6)			No	None			Change Drop More
☐ 7	is_active	tinyint(1)			No	None			Change Drop More
☐ 8	ormawa_id	bigint(20)			No	None			Change Drop More

Gambar 4.26 Tabel *Official Report*

4.3.8 Implementasi Tabel *Notification*

Gambar 4.24 merupakan implementasi tabel *Notification* dengan id sebagai *primary key*.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐ 1	id	bigint(20)			No	None		AUTO_INCREMENT	Change Drop More
☐ 2	title	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐ 3	message	longtext	utf8mb4_general_ci		No	None			Change Drop More
☐ 4	is_read	tinyint(1)			No	None			Change Drop More
☐ 5	recipient	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐ 6	created_date	datetime(6)			No	None			Change Drop More

Gambar 4.27 Tabel *Notification*

4.4 PENGUJIAN SISTEM

Pengujian sistem dilakukan dalam dua tahap untuk memastikan bahwa seluruh fungsi aplikasi berjalan sesuai dengan kebutuhan dan tidak menimbulkan kesalahan selama penggunaan. Tahap-tahap pengujian meliputi, pengujian internal dengan metode *black box testing* yaitu pengujian fungsionalitas setiap fitur pada sistem, dan pengujian pengalaman *user* dengan metode *User Acceptance Test* (UAT) untuk mengetahui apakah aplikasi telah memenuhi kebutuhan dan harapan *user* akhir.

4.4.1 Pengujian *Black Box*

Pengujian *black box* menguji antarmuka dan fungsionalitas sistem dengan memberikan *input* tertentu dan mengamati hasilnya. Tujuan dari pengujian ini adalah untuk memastikan bahwa sistem telah bekerja sesuai dengan alur kerja yang diharapkan tanpa memperhatikan struktur internal dan kode programnya.

4.4.1.1 Pengujian Form *Login*

Tabel 4.1 menunjukkan hasil *black box testing* pada halaman *login*.

Tabel 4.1 *Black Box Testing Form Login*

Fungsi yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
<i>Login</i> Ormawa (Berhasil)	User memilih role Ormawa, mengisi <i>username</i> dan password yang benar dan aktif	User berhasil <i>login</i> , diarahkan ke halaman <i>home</i> , <i>session</i> Ormawa aktif	Berhasil
<i>Login</i> Ormawa (<i>Username</i> Salah)	User memilih Ormawa, mengisi <i>username</i> yang tidak terdaftar	Pesan error " <i>Invalid username or password</i> " ditampilkan	Berhasil
<i>Login</i> Ormawa (<i>Password</i> Salah)	User memilih Ormawa, mengisi <i>username</i> benar namun password salah	Pesan error " <i>Invalid username or password</i> " ditampilkan	Berhasil
<i>Login</i> Ormawa (<i>Role</i> salah)	User memilih role Pembimbing, mengisi <i>username</i> dan password yang benar dan aktif	Pesan error " <i>Ormawa account not found</i> " ditampilkan	Berhasil
<i>Login</i> Pembimbing (Berhasil)	User memilih role Pembimbing, mengisi <i>username</i> dan password yang benar	<i>Login</i> berhasil, diarahkan ke halaman <i>home</i> , <i>session</i> Pembimbing aktif	Berhasil
<i>Login</i> Pembimbing (<i>Password</i> Salah)	User memilih role Pembimbing, password salah	Pesan error " <i>Invalid username or password</i> "	Berhasil
<i>Login</i> Pembimbing (<i>Username</i> Salah)	User memilih role Pembimbing, <i>username</i> salah, <i>password</i> benar	Pesan error " <i>Invalid username or password</i> "	Berhasil
<i>Login</i> Pembimbing (<i>Role</i> Salah)	User memilih role Ormawa,	Pesan error " <i>Ormawa account</i> "	Berhasil

	<i>username</i> benar, <i>password</i> benar	<i>not found</i> " ditampilkan	
Submit Form Kosong	Form <i>login</i> dikirim tanpa mengisi field apapun	Validasi form gagal, tetap di halaman <i>login</i>	Berhasil

4.4.1.2 Pengujian Halaman *Dashboard*

Tabel 4.2 menunjukkan hasil *black box testing* pada halaman *dashboard*.

Tabel 4.2 *Black Box Testing* Halaman *Dashboard*

Fungsi yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
Akses <i>dashboard</i> tanpa <i>login</i>	Tidak ada session ormawa atau pembimbing	Sistem mengarahkan <i>user</i> ke halaman <i>login</i>	Berhasil
Akses <i>dashboard</i> sebagai pembimbing	<i>Session</i> pembimbing aktif	Halaman <i>dashboard</i> tampil dengan data pembimbing dan Ormawa yang dibimbing	Berhasil
Akses <i>dashboard</i> sebagai ormawa	<i>Session</i> ormawa_id aktif dan ID valid	<i>Dashboard</i> tampil dengan jumlah dokumen masuk, keluar, total, bank, grafik, dan laporan aktivitas	Berhasil
Hitung dokumen masuk/keluar	Ormawa memiliki beberapa dokumen masuk dan keluar	Nilai <i>document_in_count</i> dan <i>document_out_count</i> sesuai jumlah data di database	Berhasil
Hitung dokumen 6 bulan terakhir	Ada data dokumen dalam 6 bulan terakhir	Grafik tren menampilkan data jumlah dokumen per bulan (dalam 6 bulan terakhir)	Berhasil

Laporan aktivitas (<i>activity log</i>)	Ormawa memiliki lebih dari 10 data aktivitas	Hanya 10 data terbaru yang ditampilkan di <i>dashboard</i>	Berhasil
Tidak ada dokumen sama sekali	Tidak ada data masuk/keluar/arsip dalam database	<i>Dashboard</i> tetap tampil, semua nilai 0 pada jumlah dokumen dan grafik	Berhasil

4.4.1.3 Pengujian Halaman *Document Code*

Tabel 4.3 menunjukkan hasil *black box testing* pada halaman *document code*.

Tabel 4.3 *Black Box Testing Document Code*

Fungsi yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
Menampilkan halaman daftar jenis dokumen	<i>User</i> membuka menu daftar kode dokumen	Sistem menampilkan halaman yang berisi daftar kode dokumen	Berhasil
Menampilkan data jenis dokumen	<i>User</i> memiliki sesi <i>login</i> yang valid	Sistem menampilkan semua kode dokumen yang dibuat oleh <i>user</i> dan administrator	Berhasil
Menambahkan jenis dokumen	<i>User</i> mengisi semua kolom (kode, nama, deskripsi) dengan benar	Sistem menyimpan data dan menampilkan kembali ke daftar jenis dokumen	Berhasil
Menambahkan jenis dokumen	<i>User</i> tidak mengisi kolom data dokumen	Sistem menampilkan pesan peringatan bahwa kode dokumen wajib diisi	Berhasil
<i>Edit</i> jenis dokumen	<i>User</i> mengubah data dokumen	Sistem memperbarui data	Berhasil

	dengan mengisi semua form <i>input</i>	dokumen dan kembali ke daftar	
<i>Edit</i> jenis dokumen	<i>User</i> mengubah data dokumen dengan tidak mengisi semua form <i>input</i>	Sistem menampilkan pesan peringatan bahwa kode dokumen wajib diisi	Berhasil
Mencari jenis dokumen	<i>User</i> mencari kode dokumen yang tidak tersedia	Sistem menampilkan pesan bahwa data tidak ditemukan	Berhasil
Menghapus jenis dokumen	<i>User</i> menghapus dokumen yang dibuat oleh dirinya sendiri	Sistem menghapus dokumen dari daftar	Berhasil

4.4.1.4 Pengujian Halaman Ormawa Code

Tabel 4.4 menunjukkan hasil *black box testing* pada halaman *ormawa code*.

Tabel 4.4 *Black Box Testing* Ormawa Code

Fungsi yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
Menampilkan halaman Ormawa Code	<i>User</i> mengakses menu Ormawa Code	Halaman Ormawa Code ditampilkan tanpa error	Berhasil
Menampilkan data Ormawa aktif	Sistem mengambil semua data Ormawa yang statusnya aktif	Sistem mengembalikan data berisi daftar nama, kode, deskripsi, dan status aktif	Berhasil
Menampilkan deskripsi kosong	Salah satu data ormawa tidak memiliki deskripsi	Sistem tetap menampilkan data dengan kolom deskripsi kosong	Berhasil

4.4.1.5 Pengujian Halaman Document Number

Tabel 4.5 menunjukkan hasil *black box testing* pada halaman *document number*.

Tabel 4.5 *Black Box Testing Document Number*

Fungsi yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
Akses halaman <i>Generate Document Number</i>	<i>User</i> membuka halaman <i>generate</i> dokumen	Halaman menampilkan form <i>input</i> seperti nama dokumen, jenis dokumen, dan kode Ormawa	Berhasil
Validasi saat form kosong	<i>User</i> menekan tombol <i>generate</i> tanpa mengisi salah satu atau semua isian	Muncul peringatan bahwa semua isian harus diisi dan proses tidak dilanjutkan	Berhasil
Konfirmasi sebelum generate	<i>User</i> mengisi semua isian, lalu menekan tombol <i>generate</i>	Muncul kotak konfirmasi sebelum nomor dokumen benar-benar dibuat	Berhasil
Pembuatan nomor dokumen berhasil	Nomor dokumen dibuat setelah <i>user</i> menekan <i>yes</i> pada konfirmasi	Nomor dokumen ditampilkan dengan format tertentu, dan tersimpan dalam sistem	Berhasil
Penyalinan nomor dokumen	<i>User</i> menekan tombol <i>copy</i> setelah nomor dokumen muncul	Muncul notifikasi bahwa nomor berhasil disalin	Berhasil
Format nomor sesuai	<i>User</i> berasal dari Ormawa	Nomor dokumen dibuat mengikuti template yang telah ditentukan oleh Ormawa	Berhasil

template Ormawa	dengan format nomor khusus		
Format nomor menggunakan format standar	User berasal dari Ormawa tanpa format khusus	Nomor dokumen dibuat dengan format standar (kode/urutan/Ormawa/bulan/tahun)	Berhasil

4.4.1.6 Pengujian Halaman *Document Archive*

Tabel 4.6 menunjukkan hasil *black box testing* pada halaman *document archive*.

Tabel 4.6 *Black Box Testing Document Archive*

Fungsi yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
Akses halaman daftar dokumen	User masuk ke halaman <i>Document Archive</i>	Daftar dokumen milik Ormawa yang sedang <i>login</i> ditampilkan	Berhasil
Pencarian dokumen	User mengisi kolom pencarian dan menekan enter	Dokumen dengan nama atau nomor sesuai kata kunci ditampilkan	Berhasil
Tambah dokumen dengan semua isian valid	User mengisi semua isian dan menekan tombol <i>save</i>	Dokumen berhasil disimpan, thumbnail dibuat jika tidak diunggah	Berhasil
Tambah dokumen tanpa mengunggah thumbnail	User mengisi semua isian dan tidak mengunggah thumbnail	Sistem otomatis membuat thumbnail dari file pdf	Berhasil
Tambah dokumen dengan isian wajib kosong	User mengirim form tanpa mengisi nama dokumen, nomor, deskripsi, atau file pdf	Muncul pesan peringatan bahwa semua field wajib harus diisi	Berhasil

<i>Edit</i> deskripsi dokumen	<i>User</i> mengubah deskripsi dan menyimpan	Deskripsi dokumen diperbarui, data disimpan, dan log ditambahkan	Berhasil
<i>Edit</i> dokumen dan unggah thumbnail baru	<i>User</i> mengunggah thumbnail baru saat <i>mengedit</i> dokumen	Thumbnail dokumen diperbarui sesuai file yang baru	Berhasil
<i>Edit</i> dokumen dan unggah file pdf baru	<i>User</i> mengunggah file pdf baru saat <i>mengedit</i>	File diganti, ukuran diperbarui, dan jika tidak ada thumbnail baru maka thumbnail otomatis diperbarui	Berhasil
Simpan dokumen tanpa deskripsi saat <i>edit</i>	<i>User</i> menghapus deskripsi dan mencoba menyimpan	Muncul peringatan bahwa deskripsi wajib diisi	Berhasil
Unduh dokumen yang tersedia	<i>User</i> menekan tombol download pada dokumen yang memiliki file	File pdf berhasil diunduh	Berhasil

4.4.1.7 Pengujian Halaman Mail

Tabel 4.7 menunjukkan hasil *black box testing* pada halaman *mail*.

Tabel 4.7 *Black Box Testing Mail*

Fungsi yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
Akses Halaman <i>Mail</i>	<i>User</i> membuka halaman <i>mail</i>	Halaman ditampilkan dengan benar dan menampilkan daftar <i>incoming</i> dan <i>outgoing document</i>	Berhasil
Menampilkan Data <i>Mail</i>	<i>User</i> melihat tabel surat masuk dan	Semua data surat ditampilkan	Berhasil

	keluar setelah <i>login</i>	sesuai <i>user</i> yang <i>login</i>	
Mengirim <i>Mail</i> dengan <i>Data</i> Lengkap	<i>User</i> mengisi semua kolom form dan menekan tombol kirim	Dokumen berhasil dikirim dan masuk ke daftar dokumen	Berhasil
Mengirim <i>Mail</i> Tanpa Mengisi Form	<i>User</i> menekan tombol kirim tanpa mengisi semua kolom wajib	Muncul pesan peringatan bahwa kolom wajib harus diisi	Berhasil
Validasi Form Sebelum Dikirim	<i>User</i> mengisi form secara tidak lengkap dan menekan tombol kirim	Dokumen berhasil dikirim dan masuk ke daftar dokumen	Berhasil
Notifikasi ke Penerima	Setelah dokumen dikirim ke Ormawa lain	Penerima menerima notifikasi bahwa ada dokumen masuk	Berhasil
Melihat File yang Dikirim	<i>User</i> menekan link file pada daftar dokumen	File dokumen terbuka di tab baru	Berhasil
Akses Tanpa <i>Login</i>	<i>User</i> mencoba mengakses halaman dokumen tanpa <i>login</i>	<i>User</i> diarahkan ke halaman <i>login</i>	Berhasil
Format Tanggal	<i>User</i> melihat kolom tanggal pada daftar dokumen	Format tanggal ditampilkan dalam bentuk tahun-bulan-hari dan jam-menit	Berhasil

4.4.1.8 Pengujian Halaman *Activity log*

Tabel 4.8 menunjukkan hasil *black box testing* pada halaman *Activity log*.

Tabel 4.8 *Black Box Testing Activity log*

Fungsi yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
Akses Halaman <i>Activity log</i>	<i>User</i> membuka halaman <i>Activity log</i> melalui menu	Halaman menampilkan tabel log aktivitas Ormawa	Berhasil
Menampilkan Ormawa dan <i>Email</i>	<i>User</i> melihat kolom Ormawa dan <i>Email</i>	Kolom Ormawa dan <i>Email</i> tampil sesuai data yang tersimpan	Berhasil
Menampilkan Aktivitas dan Tanggal	<i>User</i> melihat aktivitas dan tanggal log	Aktivitas dan waktu log ditampilkan dalam format yang sesuai (YYYY-MM-DD HH:MM:SS)	Berhasil
Urutan Tanggal Terbaru	Data log diurutkan berdasarkan waktu paling baru	Data ditampilkan dari yang terbaru ke yang terlama	Berhasil

4.4.1.9 Pengujian Halaman *Notification*

Tabel 4.1 menunjukkan hasil *black box testing* pada halaman *notification*.

Tabel 4.9 *Black Box Testing Notification*

Fungsi yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
Menampilkan Notifikasi	<i>User</i> mengakses halaman notifikasi	Semua notifikasi milik <i>user</i> ditampilkan sesuai urutan waktu	Berhasil
Menampilkan Waktu Notifikasi	<i>User</i> melihat informasi waktu pada setiap notifikasi	Waktu ditampilkan dalam format relatif (contoh: "2 hours ago")	Berhasil

Tombol "Read All" Berfungsi	User menekan tombol "Read All"	Semua notifikasi yang belum dibaca ditandai sebagai telah dibaca dan halaman reload	Berhasil
Menampilkan Pesan Jika Kosong	Tidak ada notifikasi untuk user	Pesan " <i>There are no new notifications.</i> " ditampilkan	Berhasil
Menghitung Notifikasi Belum Dibaca	Sistem menghitung jumlah notifikasi is_read=False untuk user tertentu	Jumlah yang dikembalikan sesuai dengan notifikasi yang belum dibaca	Berhasil

4.4.2 Pengujian UAT

Pengujian User Acceptance Testing (UAT) pada sistem ini dilakukan dengan menggunakan metode kuesioner yang melibatkan 25 responden sebagai pengguna akhir dengan 20 pertanyaan. Setiap pertanyaan dinilai menggunakan skala Likert 1–3, di mana nilai 1 menunjukkan tingkat ketidaksesuaian paling rendah dan nilai 3 menunjukkan tingkat kepuasan atau kesesuaian paling tinggi. Hasil pengujian ini digunakan untuk mengetahui sejauh mana sistem telah memenuhi kebutuhan pengguna.

Data yang diperoleh dari para responden dalam pengujian UAT kemudian dianalisis untuk mengukur tingkat penerimaan pengguna terhadap sistem yang dikembangkan. Analisis dilakukan dengan menghitung nilai rata-rata dari setiap item pertanyaan, berdasarkan skor yang diberikan responden pada skala penilaian 1–3. Tabel 4.10 menampilkan table skala *Likert*.

Tabel 4.10 Tabel Skala *Likert*

Skala Jawaban	Keterangan	Skor	Persentase
TS	Tidak setuju	1	0% - 33%
S	Setuju	2	34% - 66%
SS	Sangat setuju	3	67% - 100%

Rumus yang digunakan untuk menghitung persentase kepuasan atau penerimaan pengguna adalah sebagai berikut:

$$P = \frac{S}{\text{Skor Ideal}} \times 100\%$$

Keterangan:

P = Presentase

S = Jumlah frekuensi jawaban x dengan skor setiap jawaban

Skor Ideal = Skor tertinggi x jumlah responden

$$\begin{aligned}\text{Skor Ideal} &= 3 \times \text{jumlah responden} \\ &= 3 \times 25 \\ &= 75\end{aligned}$$

Berikut ini adalah persentase jawaban dari setiap pertanyaan yang diperoleh melalui kuesioner:

1. Seberapa menarik tampilan aplikasi menurut Anda? (Skala 1–3)

1 = Tidak menarik, 3 = Sangat menarik.

Tabel 4.11 Skor Pertanyaan 1

Jawaban	Skor	Frekuensi	S
TS	1	0	$1 \times 0 = 0$
S	2	7	$2 \times 7 = 14$
SS	3	18	$3 \times 18 = 54$
Jumlah			68

$$\begin{aligned}P &= \frac{68}{75} \times 100\% \\ &= 90.66\%\end{aligned}$$

Berdasarkan hasil dari pertanyaan nomor 1, dengan persentase sebesar 90.66% dapat disimpulkan bahwa tampilan aplikasi dinilai sangat menarik oleh mayoritas pengguna.

2. Seberapa mudah aplikasi ini digunakan? (Skala 1–3)

1 = Sulit digunakan, 3 = Sangat mudah

Tabel 4.12 Skor Pertanyaan 2

Jawaban	Skor	Frekuensi	S
TS	1	0	$1 \times 0 = 0$
S	2	11	$2 \times 11 = 22$
SS	3	14	$3 \times 14 = 42$
Jumlah			64

$$P = \frac{64}{75} \times 100\%$$

$$= 85.33\%$$

Berdasarkan hasil dari pertanyaan nomor 2, dengan persentase sebesar 85.33%, dapat disimpulkan bahwa tampilan aplikasi dinilai sangat mudah digunakan oleh mayoritas pengguna

3. Apakah aplikasi ini membantu Anda dalam mengelola dokumen dan surat Ormawa?

Tabel 4.13 Skor Pertanyaan 3

Jawaban	Skor	Frekuensi	S
TS	1	0	$1 \times 0 = 0$
S	2	7	$2 \times 7 = 14$
SS	3	18	$3 \times 18 = 54$
Jumlah			68

$$P = \frac{68}{75} \times 100\%$$

$$= 90.66\%$$

Berdasarkan hasil dari pertanyaan nomor 3, dengan persentase sebesar 90.66%, dapat disimpulkan bahwa aplikasi dinilai sangat mudah membantu oleh mayoritas pengguna.

4. Sistem penomoran otomatis sangat membantu mencegah duplikasi dan kesalahan administrasi dokumen?

Tabel 4.14 Skor Pertanyaan 4

Jawaban	Skor	Frekuensi	S
TS	1	0	1 x 0 = 0
S	2	6	2 x 6 = 12
SS	3	19	3 x 19 = 57
Jumlah			69

$$P = \frac{69}{75} \times 100\%$$

$$= 92\%$$

Berdasarkan hasil perhitungan pada pertanyaan nomor 4, diperoleh skor persentase sebesar 92%, yang menunjukkan bahwa mayoritas responden menilai sistem penomoran otomatis sangat membantu dalam mencegah duplikasi dan kesalahan administrasi dokumen.

5. Saya merasa sistem ini dapat membantu menyimpan dan mencari dokumen dengan lebih mudah dibanding cara manual?

Tabel 4.15 Skor Pertanyaan 5

Jawaban	Skor	Frekuensi	S
TS	1	0	1 x 0 = 0
S	2	6	2 x 6 = 12
SS	3	19	3 x 19 = 57
Jumlah			69

$$P = \frac{69}{75} \times 100\%$$

$$= 92\%$$

Berdasarkan hasil perhitungan pada pertanyaan nomor 5, diperoleh skor persentase sebesar 92%, yang menunjukkan bahwa mayoritas responden merasa sistem ini membantu dalam menyimpan dan mencari dokumen dengan lebih mudah dibandingkan cara manual.

6. Sistem ini memberikan fleksibilitas dalam mengakses dokumen dari mana saja dan kapan saja?

Tabel 4.16 Skor Pertanyaan 6

Jawaban	Skor	Frekuensi	S
TS	1	0	$1 \times 0 = 0$
S	2	4	$2 \times 4 = 8$
SS	3	21	$3 \times 21 = 63$
Jumlah			71

$$P = \frac{71}{75} \times 100\%$$

$$= 94.66\%$$

Berdasarkan hasil perhitungan pada pertanyaan nomor 6, diperoleh skor persentase sebesar 94,66%, yang menunjukkan bahwa mayoritas responden menilai sistem ini memberikan fleksibilitas tinggi dalam mengakses dokumen kapan saja dan dari mana saja.

7. Penggunaan kontrol akses dan riwayat perubahan penting untuk menjaga keamanan dokumen organisasi?

Tabel 4.17 Skor Pertanyaan 7

Jawaban	Skor	Frekuensi	S
TS	1	0	$1 \times 0 = 0$
S	2	5	$2 \times 5 = 10$
SS	3	20	$3 \times 20 = 60$
Jumlah			70

$$P = \frac{69}{75} \times 100\%$$

$$= 93.33\%$$

Berdasarkan hasil perhitungan pada pertanyaan nomor 7, diperoleh skor persentase sebesar 93,33%, yang menunjukkan bahwa mayoritas responden menganggap penggunaan kontrol akses dan riwayat perubahan sangat penting untuk menjaga keamanan dokumen organisasi.

8. Saya lebih menyukai sistem terpusat dibanding penyimpanan di Google Drive atau manual karena kontrol perubahannya lebih jelas?

Tabel 4.18 Skor Pertanyaan 8

Jawaban	Skor	Frekuensi	S
TS	1	0	$1 \times 0 = 0$
S	2	12	$2 \times 12 = 24$
SS	3	13	$3 \times 13 = 39$
Jumlah			63

$$P = \frac{63}{75} \times 100\%$$

$$= 84\%$$

Berdasarkan hasil perhitungan pada pertanyaan nomor 8, diperoleh skor persentase sebesar 84%, yang menunjukkan bahwa mayoritas responden lebih menyukai sistem terpusat karena kontrol penyimpanan dan perubahannya lebih jelas.

9. Apakah anda setuju akses ke dokumen dibatasi sesuai hak pengguna yang telah ditentukan membuat dokumen lebih aman dari user yang tidak bertanggung jawab?

Tabel 4.19 Skor Pertanyaan 9

Jawaban	Skor	Frekuensi	S
TS	1	0	$1 \times 0 = 0$
S	2	5	$2 \times 5 = 10$
SS	3	20	$3 \times 20 = 60$
Jumlah			70

$$P = \frac{70}{75} \times 100\%$$

$$= 93.33\%$$

Berdasarkan hasil perhitungan pada pertanyaan nomor 9, diperoleh skor persentase sebesar 93,33%, yang menunjukkan bahwa mayoritas responden

menilai pembatasan akses dokumen sesuai hak pengguna efektif dalam menjaga keamanan dokumen.

10. Fitur pengiriman dokumen antar Ormawa mempermudah ormawa dalam proses administrasi antar Ormawa?

Tabel 4.20 Skor Pertanyaan 10

Jawaban	Skor	Frekuensi	S
TS	1	0	$1 \times 0 = 0$
S	2	8	$2 \times 10 = 20$
SS	3	17	$3 \times 15 = 45$
Jumlah			65

$$P = \frac{65}{75} \times 100\%$$

$$= 86.66\%$$

Berdasarkan hasil perhitungan pada pertanyaan nomor 10, diperoleh skor persentase sebesar 86.66%, yang menunjukkan bahwa fitur pengiriman dokumen antar Ormawa membantu ormawa mengelola administrasi.

11. Dashboard laporan membantu penulis memantau aktivitas surat menyurat secara efisien?

Tabel 4.21 Skor Pertanyaan 11

Jawaban	Skor	Frekuensi	S
TS	1	0	$1 \times 0 = 0$
S	2	4	$2 \times 4 = 8$
SS	3	21	$3 \times 21 = 63$
Jumlah			71

$$P = \frac{71}{75} \times 100\%$$

$$= 94.66\%$$

Berdasarkan hasil perhitungan pada pertanyaan nomor 11, diperoleh skor persentase sebesar 94,66%, yang menunjukkan bahwa dashboard laporan

sangat membantu penulis dalam memantau aktivitas surat menyurat secara efisien menurut mayoritas responden.

12. Apakah form inputan berjalan dengan baik?

Tabel 4.22 Skor Pertanyaan 12

Jawaban	Skor	Frekuensi	S
TS	1	0	$1 \times 0 = 0$
S	2	12	$2 \times 12 = 24$
SS	3	13	$3 \times 13 = 39$
Jumlah			63

$$P = \frac{63}{75} \times 100\%$$

$$= 84\%$$

Berdasarkan hasil perhitungan pada pertanyaan nomor 12, diperoleh skor persentase sebesar 84%, yang menunjukkan bahwa mayoritas responden menilai form inputan pada sistem berjalan dengan baik.

13. Proses pengunggahan dokumen dan pengisian data berjalan dengan lancar?

Tabel 4.23 Skor Pertanyaan 13

Jawaban	Skor	Frekuensi	S
TS	1	0	$1 \times 0 = 0$
S	2	6	$2 \times 6 = 12$
SS	3	19	$3 \times 19 = 57$
Jumlah			69

$$P = \frac{69}{75} \times 100\%$$

$$= 92\%$$

Berdasarkan hasil perhitungan pada pertanyaan nomor 13, diperoleh skor persentase sebesar 92%, yang menunjukkan bahwa proses pengunggahan dokumen dan pengisian data dinilai berjalan dengan lancar oleh sebagian besar responden.

14. Apakah nomor yang dihasilkan sesuai format ormawa?

Tabel 4.24 Skor Pertanyaan 14

Jawaban	Skor	Frekuensi	S
TS	1	0	1 x 0 = 0
S	2	8	2 x 8 = 16
SS	3	17	3 x 17 = 51
Jumlah			67

$$P = \frac{67}{75} \times 100\%$$

$$= 89.33\%$$

Berdasarkan hasil perhitungan pada pertanyaan nomor 14, diperoleh skor persentase sebesar 89,33%, yang menunjukkan bahwa mayoritas responden menilai nomor yang dihasilkan sudah sesuai dengan format Ormawa.

15. Apakah Sistem mencatat aktivitas pengguna terhadap dokumen (seperti pengeditan atau penghapusan) berguna bagi ormawa?

Tabel 4.25 Skor Pertanyaan 15

Jawaban	Skor	Frekuensi	S
TS	1	0	1 x 0 = 0
S	2	8	2 x 8 = 16
SS	3	17	3 x 17 = 51
Jumlah			67

$$P = \frac{67}{75} \times 100\%$$

$$= 89.33\%$$

Berdasarkan hasil perhitungan pada pertanyaan nomor 15, diperoleh skor persentase sebesar 89,33%, yang menunjukkan bahwa mayoritas responden menganggap pencatatan aktivitas pengguna terhadap dokumen sangat berguna bagi Ormawa.

16. Fitur pengiriman dokumen antar Ormawa berfungsi dengan baik?

Tabel 4.26 Skor Pertanyaan 16

Jawaban	Skor	Frekuensi	S
TS	1	0	$1 \times 0 = 0$
S	2	8	$2 \times 8 = 16$
SS	3	17	$3 \times 17 = 51$
Jumlah			67

$$P = \frac{67}{75} \times 100\%$$

$$= 89.33\%$$

Berdasarkan hasil perhitungan pada pertanyaan nomor 16, diperoleh skor persentase sebesar 89,33%, yang menunjukkan bahwa fitur pengiriman dokumen antar Ormawa berfungsi dengan baik menurut sebagian besar responden.

17. Apakah fitur notifikasi berfungsi ketika ada pesan masuk?

Tabel 4.27 Skor Pertanyaan 17

Jawaban	Skor	Frekuensi	S
TS	1	0	$1 \times 0 = 0$
S	2	12	$2 \times 12 = 24$
SS	3	13	$3 \times 13 = 39$
Jumlah			63

$$P = \frac{63}{75} \times 100\%$$

$$= 84\%$$

Berdasarkan hasil perhitungan pada pertanyaan nomor 17, diperoleh skor persentase sebesar 84%, yang menunjukkan bahwa mayoritas responden menilai fitur notifikasi berfungsi dengan baik saat ada pesan masuk.

18. Apakah setiap menu dan tombol di aplikasi berfungsi dengan baik?

Tabel 4.28 Skor Pertanyaan 18

Jawaban	Skor	Frekuensi	S
TS	1	0	$1 \times 0 = 0$
S	2	9	$2 \times 9 = 18$
SS	3	16	$3 \times 16 = 48$
Jumlah			66

$$P = \frac{66}{75} \times 100\%$$

$$= 88\%$$

Berdasarkan hasil perhitungan pada pertanyaan nomor 18 diperoleh skor persentase sebesar 88%. Persentase ini menunjukkan bahwa mayoritas responden menilai semua menu dan tombol dalam aplikasi berfungsi dengan baik. Hal ini mencerminkan bahwa aspek teknis dari aplikasi sudah berjalan dengan lancar dan mendukung kenyamanan pengguna dalam berinteraksi dengan sistem.

19. Saya merasa puas menggunakan sistem ini dan bersedia menggunakannya secara berkelanjutan?

Tabel 4.29 Skor Pertanyaan 19

Jawaban	Skor	Frekuensi	S
TS	1	0	$1 \times 0 = 0$
S	2	11	$2 \times 11 = 22$
SS	3	14	$3 \times 14 = 42$
Jumlah			64

$$P = \frac{64}{75} \times 100\%$$

$$= 85.33\%$$

Berdasarkan hasil perhitungan pada pertanyaan nomor 19 diperoleh skor persentase sebesar 85,33%. Nilai ini termasuk dalam kategori sangat baik,

yang menunjukkan bahwa sebagian besar responden merasa puas dan berniat untuk terus menggunakan sistem tersebut ke depannya.

20. Seberapa puas anda terhadap fitur yang disediakan?

Tabel 4.30 Skor Pertanyaan 20

Jawaban	Skor	Frekuensi	S
TS	1	1	$1 \times 0 = 0$
S	2	7	$2 \times 7 = 14$
SS	3	18	$3 \times 18 = 54$
Jumlah			68

$$P = \frac{68}{75} \times 100\%$$

$$= 90.66\%$$

Berdasarkan hasil perhitungan pada pertanyaan nomor 20 mengenai kepuasan terhadap fitur yang disediakan, diperoleh skor persentase sebesar 90,66%. Nilai ini menunjukkan bahwa mayoritas responden merasa sangat puas terhadap fitur yang tersedia. Dengan demikian, dapat disimpulkan bahwa fitur-fitur yang disediakan telah memenuhi harapan pengguna dengan sangat baik.

4.5 PEMBAHASAN

Bab ini membahas hasil rancang bangun serta pengujian sistem informasi manajemen surat pada organisasi mahasiswa (Ormawa) Universitas Jenderal Achmad Yani Yogyakarta yang dikembangkan menggunakan *framework* Django. Pembahasan difokuskan pada bagaimana sistem ini menjawab permasalahan yang telah dirumuskan sebelumnya, serta efektivitasnya dalam meningkatkan efisiensi, akurasi, keamanan, dan kemudahan dalam pengelolaan dokumen Ormawa.

4.5.1 Implementasi Sistem

Sistem informasi ini terdiri dari beberapa fitur utama seperti *login multi-role* (Ormawa dan Pembimbing), *dashboard monitoring* dokumen, manajemen kode

dokumen, *generate* nomor dokumen otomatis, pengarsipan dokumen, pengiriman dokumen antar Ormawa (*mail*), log aktivitas, dan notifikasi pengguna.

Setiap fitur dirancang melalui pendekatan *user-centered design*, di mana desain antarmuka dan alur penggunaan sistem telah disesuaikan berdasarkan masukan dan kebutuhan pengguna. Rancangan basis data, struktur tabel, dan relasi antar entitas juga telah dibuat untuk memastikan keterhubungan antar fitur bekerja optimal dan efisien dalam proses penyimpanan serta pengambilan data.

Seluruh fitur yang ditampilkan dalam sistem ini merupakan hasil dari proses pengembangan bertahap (*sprint*) sesuai metode Agile, di mana setiap tahap pengembangan dilakukan melalui iterasi singkat, pengujian, dan evaluasi langsung bersama pengguna.

Perubahan desain, penambahan fitur, serta penyempurnaan sistem dilakukan berdasarkan feedback pengguna secara berkala. Proses ini dikukuhkan melalui pelaksanaan User Acceptance Test (UAT) dan pengumpulan dokumen umpan balik dari pihak Ormawa, yang dijadikan dasar dalam validasi dan penyempurnaan sistem.

Hasil kuisioner tersebut menunjukkan bahwa fitur-fitur yang dikembangkan telah sesuai dengan kebutuhan pengguna dan mendapatkan respons positif, sebagaimana tercantum pada Lampiran 1-3.

4.5.2 Analisis Terhadap Tujuan Penelitian

Subbab ini membahas analisis terhadap pencapaian tujuan penelitian yang telah dirumuskan pada bab sebelumnya. Analisis ini dilakukan berdasarkan hasil implementasi sistem, pengujian, serta data yang diperoleh selama proses penelitian. Tujuannya adalah untuk mengevaluasi sejauh mana sistem yang dikembangkan mampu memenuhi sasaran yang telah ditetapkan dan memberikan solusi terhadap permasalahan yang diangkat dalam penelitian ini. Berikut hasilnya:

1. Kemudahan dalam manajemen dokumen: Fitur form *input*, upload *file*, dan penyimpanan terpusat telah memudahkan proses pencatatan dan pengiriman dokumen antar Ormawa. Hal ini menjawab pertanyaan pertama mengenai kemudahan pengelolaan dokumen.

2. Pencegahan kesalahan penomoran: Sistem generate document number secara otomatis telah terbukti mencegah duplikasi serta menjamin konsistensi format penomoran dokumen berdasarkan jenis, ormawa, dan tanggal.
3. Pencarian dan pelacakan dokumen: *DataTable* dan fitur pencarian yang tersedia di halaman dokumen memudahkan pengguna untuk menemukan dokumen tertentu secara cepat berdasarkan kata kunci tertentu.
4. Keamanan dokumen: Sistem ini menerapkan pembatasan akses berdasarkan sesi *login* dan *role* pengguna, serta mencatat semua aktivitas penting dalam fitur *activity log* dan notifikasi.

4.5.3 Kesimpulan Pengujian

Berdasarkan hasil pengujian yang telah dilakukan, baik melalui metode *black box testing* maupun *user acceptance testing* (UAT), dapat disimpulkan bahwa sistem informasi manajemen dokumen Ormawa Universitas Jenderal Achmad Yani Yogyakarta telah bekerja secara optimal dan sesuai dengan tujuan pengembangannya. Pada pengujian *black box*, seluruh fitur utama diuji. Hasilnya menunjukkan bahwa semua fungsi seperti *login*, *dashboard*, pengelolaan kode dokumen, pengarsipan dokumen, pengiriman surat antar Ormawa, notifikasi, serta riwayat aktivitas berjalan sesuai harapan. Setiap skenario pengujian menghasilkan output yang tepat, dan sistem mampu menampilkan validasi yang benar saat terjadi kesalahan input. Tidak ditemukan kendala yang signifikan yang mengganggu alur kerja sistem selama proses pengujian.

Sementara itu, pada pengujian UAT yang melibatkan 25 responden dengan 20 pertanyaan, pengguna memberikan penilaian 90.66% puas terhadap fitur yang disediakan oleh sistem, sistem ini membantu ormawa dalam mengelola dokumen dan tampilan yang menarik. Hasil keseluruhan kuesioner menunjukkan bahwa setiap pertanyaan, mayoritas responden memberikan penilaian diatas 83% merasa fitur-fiturnya dan semua fungsionalitas sistem berjalan dengan baik.

Berdasarkan kedua pengujian ini, sistem terbukti layak digunakan dan telah memenuhi kebutuhan pengguna dengan baik. Sistem ini dapat diimplementasikan

secara penuh untuk menggantikan metode manual yang sebelumnya digunakan, guna mendukung pengelolaan dokumen yang lebih tertata, efisien, dan aman.

4.5.4 Kelebihan Sistem

1. Otomatisasi penomoran dokumen. Sistem mampu membuat nomor dokumen secara otomatis dan konsisten sesuai format yang ditentukan, sehingga mengurangi kesalahan penulisan dan duplikasi nomor.
2. Pengelolaan dokumen terpusat. Semua dokumen disimpan dalam *database* terpusat, yang memudahkan pelacakan, pencarian, dan pengelolaan dokumen tanpa perlu membuka banyak file atau folder manual.
3. Aksesibilitas fleksibel dan *Role-Based*. Sistem dapat diakses kapan saja dan di mana saja oleh pengguna yang memiliki akun dan hak akses. Fitur login dibedakan antara Ormawa dan pembimbing untuk menjamin keamanan dan keterbatasan akses.
4. Notifikasi dokumen masuk. Ketika dokumen dikirim ke Ormawa lain, sistem secara otomatis mengirim notifikasi ke penerima sehingga mempermudah komunikasi antar organisasi.
5. Riwayat aktivitas. Aktivitas penting pengguna dicatat dalam log sistem sehingga administrator dapat melacak setiap perubahan atau aktivitas yang terjadi.
6. *Dashboard* interaktif. Tersedia *dashboard* yang menampilkan statistik dokumen masuk, keluar, serta grafik aktivitas selama enam bulan terakhir, yang mempermudah monitoring kinerja.
7. *User interface simpel* dan responsif. Antarmuka sistem dirancang agar mudah dipahami, dengan layout yang rapi dan penggunaan DataTable untuk menampilkan data secara dinamis.

4.5.5 Kekurangan Sistem

1. Belum mendukung pengiriman dokumen antar lembaga di luar Ormawa
2. Belum tersedia fitur tanda tangan digital. Sistem belum mendukung validasi melalui tanda tangan elektronik atau digital, sehingga untuk saat ini legalitas dokumen masih dilakukan secara manual.

3. Tergantung paada koneksi internet. Karena berbasis web, sistem ini membutuhkan koneksi internet yang stabil. Kinerja sistem akan menurun jika pada jaringan yang lambat.

PERPUSTAKAAN
UNIVERSITAS JENDERAL ACHMAD YANI
YOGYAKARTA