

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Hasil Pengumpulan Data

Pengumpulan data adalah proses pencarian informasi atau sampel data yang dapat digunakan untuk mengetahui proses pengolahan data pada tempat atau objek untuk melakukan penelitian dan pembuatan suatu sistem, karena teknik pengumpulan data merupakan proses yang sangat penting maka data yang dikumpulkan harus berupa data yang benar dan akurat. Dalam penelitian ini data diperoleh dari tahapan observasi dan wawancara dengan pihak yang bersangkutan serta tahap studi literatur yang didapat dari berbagai jurnal dan beberapa buku yang berhubungan dengan penelitian ini. Adapun hasil dari proses pengumpulan data adalah sebagai berikut:

##### 4.1.1 Hasil Observasi dan Wawancara

###### 1. Observasi

Dalam penelitian ini observasi dilakukan dengan cara terjun dan mengamati secara langsung pada tempat atau objek penelitian untuk mendapatkan data dan informasi mengenai permasalahan yang terjadi pada pengarsipan data inpassing dosen Universitas Nurul Jadid. Adapun hasil dari observasi yang dilakukan ialah proses pengarsipan data inpassing dosen masih manual sehingga membutuhkan waktu yang cukup lama untuk mendapatkan kembali data yang dibutuhkan dikarenakan banyaknya tumpukan berkas hard copy yang ada.

###### 2. Wawancara

Proses wawancara yang dilakukan dalam penelitian ini adalah untuk mendapatkan beberapa jawaban penting mengenai pengarsipan data inpassing dosen Universitas Nurul Jadid dengan sampel data dari tiga fakultas yaitu fakultas teknik, fakultas kesehatan dan fakultas soshum. Pada proses ini, wawancara dilakukan kepada kepala tim PAK Universitas Nurul Jadid terkait

proses pengarsipan data inpassing dosen yang dilakukan selama ini.

**Tabel 4.1** Hasil Wawancara

No	Pertanyaan	Jawaban
1	Bagaimana proses pengarsipan data inpassing dosen UNUJA yang terjadi selama ini?	Proses pengarsipan data inpassing yang berjalan selama ini masih menggunakan cara manual yaitu disimpan dan diarsipkan dengan cara meletakkan berkas tersebut pada tempat yang sudah tersedia
2	Apa permasalahan yang sering terjadi dalam proses pengarsipan inpassing?	Petugas sering kuwalahan saat membutuhkan suatu berkas yang ada pada arsip dikarenakan banyaknya tumpukan berkas yang ada
3	Terdapat berapa petugas yang dapat mengakses atau mengurus proses pengarsipan inpassing?	Untuk petugas itu ada beberapa akan tetapi hak akses cukup satu saja
4	Bagaimana cara mendapatkan data inpassing yang akan disimpan dan menjadi arsip?	Dengan cara menerima berkas pengajuan dari dosen yang mengajukan
5	Apa yang dilakukan setelah data tersebut didapatkan dan menjadi arsip?	Setelah data tersebut didapat, maka data tersebut diserahkan kepada kantor pusat untuk diproses agar dosen pengaju mendapat SK Inpassing
6	Sistem seperti apa yang diharapkan untuk mempermudah dalam pengarsipan data inpassing dosen?	Petugas mengharapkan adanya aplikasi yang dapat membantu mempermudah proses pengarsipan data inpassing berbasis web dengan menginputkan data dosen pengaju dan hasil scan dari berkas pengajuan inpassing

**Tabel 4.2** Hasil Wawancara (Lanjutan)

7	Bagaimana cara mengetahui berkas dosen yang telah diterima atau telah dinyatakan lolos inpassing?	Maka tim pusat akan memberikan link SK kepada admin lalu admin memberikan link tersebut kepada dosen melewati sebuah group telegram
8	Bagaimana cara memberitahu dosen tentang berkas inpassing yang telah dikumpulkan?	Dengan cara mmbuka link yang telah diberikan oleh pusat untuk yang sudah lolos dan memberikan informasi di group telegram untuk berkas yang masih diproses atau ditolak dengan kategori tertentu

### 3. Studi literatur

Dalam penelitian ini proses pengumpulan data tidak hanya dikumpulkan sengan cara observasi dan wawancara saja melainkan juga dengan tahap studi literatur yaitu mengumpulkan data dari berbagai sumber untuk menjadi referensi dari beberapa buku dan perbandingan permasalahan dari berbagai jurnal dengan penelitian ini tentang penelitian yang diambil terkait proses pengalihan jabatan atau inpassing.

## 4.2 Hasil Dan Analisis Desain

### 4.2.1 Analisis

Analisis dilakukan untuk menganalisa dan mengetahui bagaimana proses pengarsipan sistem lama untuk dibandingkan dengan sistem baru yang diharapkan dalam proses pengarsipan data inpassing dosen Universitas Nurul Jadid.

#### 1. Analisis sistem Lama

Berdasarkan analisa yang telah dilakukan, didapatkan sebuah sistem yang masih rawan akan hilang dan rusaknya data yang ada. Dimana dalam sistem lama proses pengarsipan data inpassing

dosen Universitas Nurul Jadid masih dilakukan secara manual yaitu menyimpan berkas hard copy pada tempat yang sudah disediakan yang mengakibatkan seringkali terjadi kesulitan dalam pencarian suatu berkas saat data tersebut dibutuhkan kembali.

## 2. Analisis Sistem Baru

Dengan adanya beberapa permasalahan yang terdapat pada sistem lama, didapatkan sebuah solusi yang diharapkan dapat mengatasi permasalahan yang ada, yaitu dengan adanya sistem baru berupa aplikasi pengarsipan data inpassing dosen Universitas Nurul Jadid berbasis web yang diharapkan dapat membantu mempermudah petugas dalam mengarsipkan data inpassing dengan baik juga mempermudah petugas dalam mencari data yang telah diarsipkan sebelumnya tanpa khawatir akan rusak dan hilangnya data yang ada.

### 4.2.2 Desain sistem

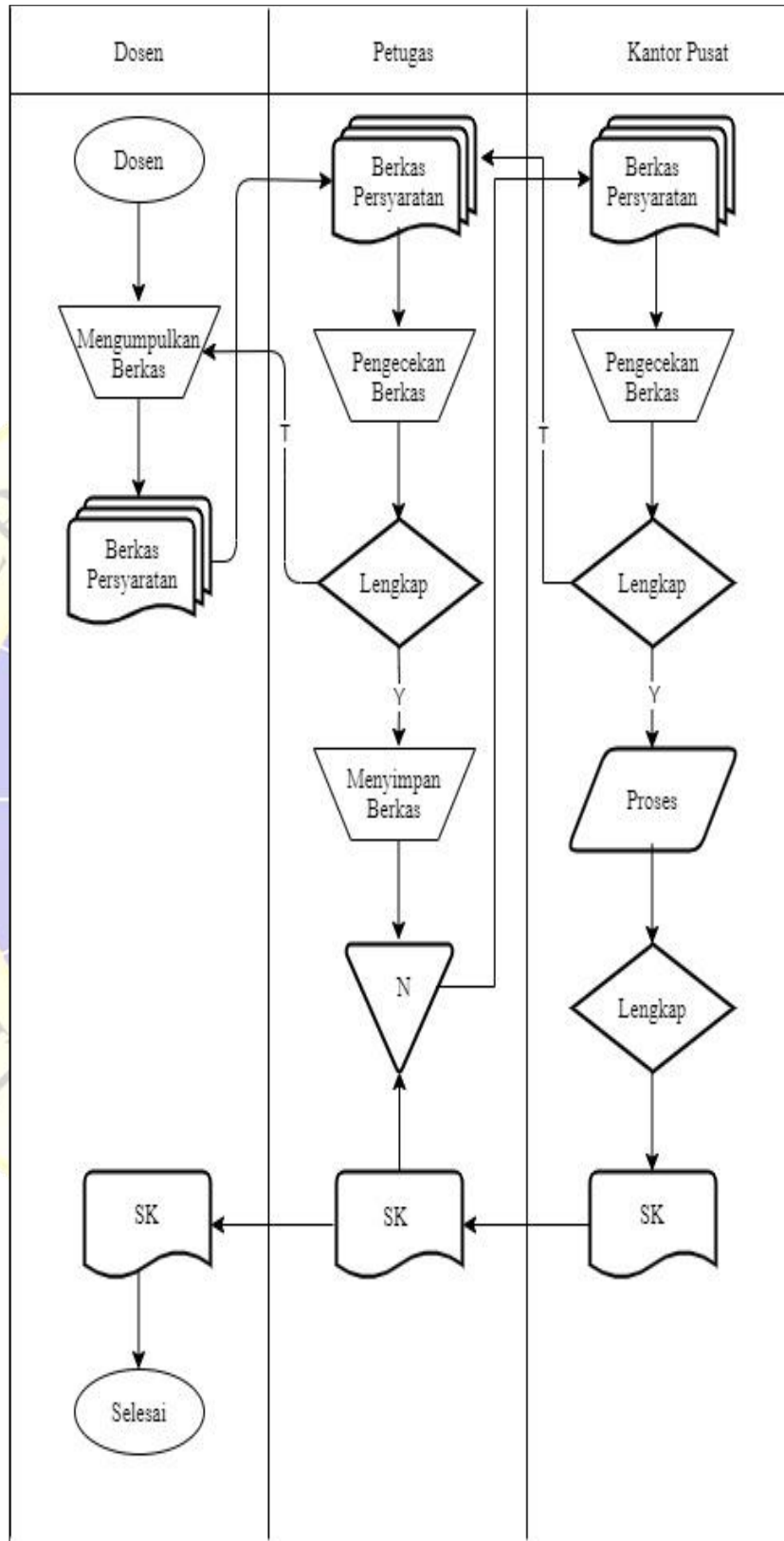
Desain sistem dilakukan untuk memberikan sebuah gambaran mengenai proses berjalannya sistem secara keseluruhan. Dari hasil analisa yang dilakukan, didapatkan beberapa rancangan untuk mendesain dan mengetahui jalannya sebuah sistem yaitu perancangan berupa *flowchart*, *context diagram*, bagan berjenjang, *DFD (Data Flow Diagram)* dan *ERD (Entity Relationship Diagram)*.

Berikut rincian beberapa rancangan sistem yang digunakan:

#### a) *Flowchart*

*Flowchart* menggambarkan sebuah aliran pada sistem yang ada dan yang akan dibangun yaitu *flowchart* sistem lama dan *flowchart* sistem baru atau *flowchart* aplikasi.

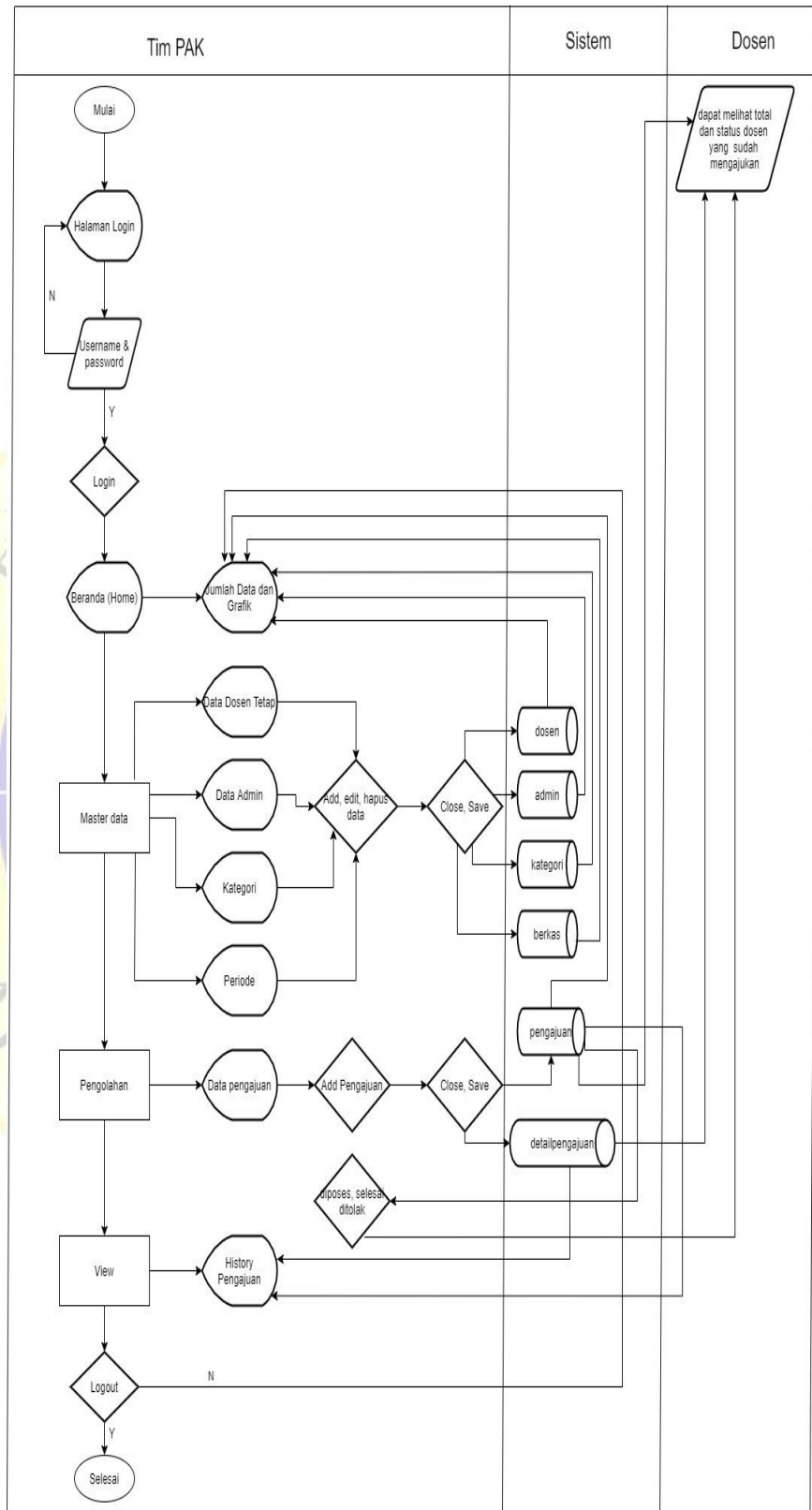
1. Flowchart Sistem lama



Gambar 4.1 Flowchart Sistem lama



## 2. Flowchart Sistem Baru (aplikasi)



Gambar 4.2 Flowchart Sistem baru (aplikasi)

b) DFD (*Data Flow Diagram*)

DFD (*Data Flow Diagram*) atau model proses (*process model*) adalah sekumpulan proses analisis yang digunakan untuk menggambarkan aliran input dalam sebuah sistem serta *output* yang dihasilkan atau gambaran dari sebuah system yang dibuat agar pengguna lebih memahami tentang struktur data dari sistem secara logika. Berikut gambaran dari Data Flow Diagram aplikasi pengarsipan data inpassing dosen Universitas Nurul Jadid.

1. *ContextDiagram* (DFD level 0)

*Context Diagram* adalah gambaran yang mendefinisikan aliran data dari sebuah sistem yang dibuat dan apa saja yang dapat dilakukan oleh admin dengan adanya sistem tersebut dengan lebih jelas lagi serta mengidentifikasi hubungan antara setiap entitas dari sistem itu sendiri yang ditampilkan dalam satu proses.



Gambar 4.3 Context Diagram (DFD Level 0)

## 2. Bagan Berjenjang

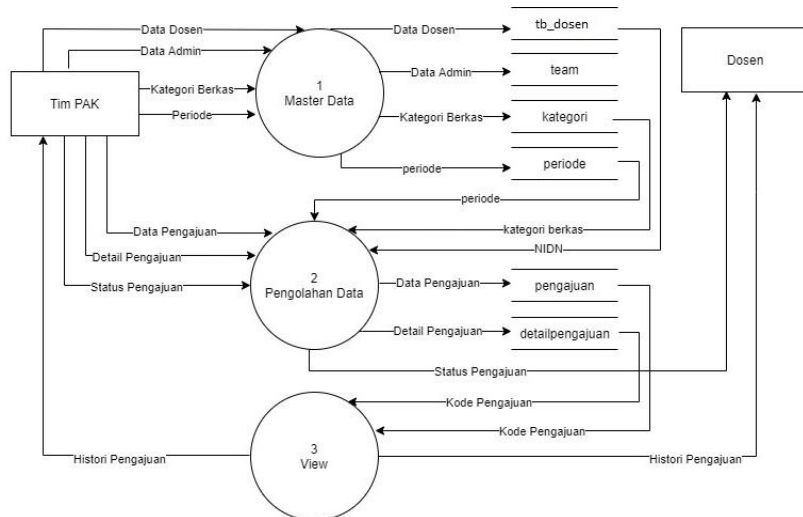


Gambar 4.4 Bagan Berjenjang

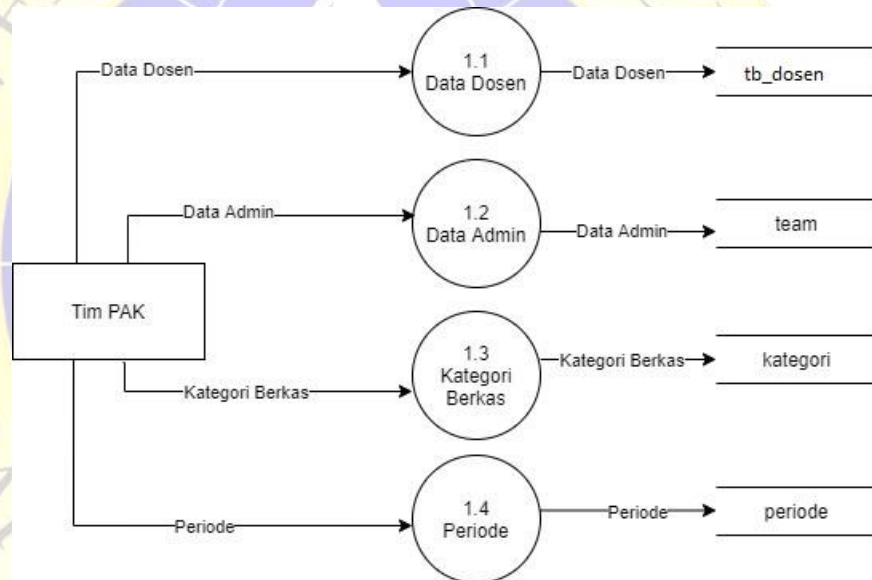
## 3. Data Flow Diagram Level 1 dan 2

Setelah menggambarkan hubungan eksternal dengan sistem dan membuat bagan berjenjang atau juga bisa disebut dengan dekomposisi diagram, maka langkah selanjutnya ialah membuat gambaran DFD level 1 dan 2 sebagai solusi pemecahan dari digram konteks dan penyimpanan dari seluruh data yang ada.

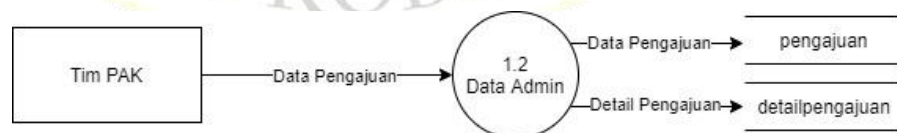




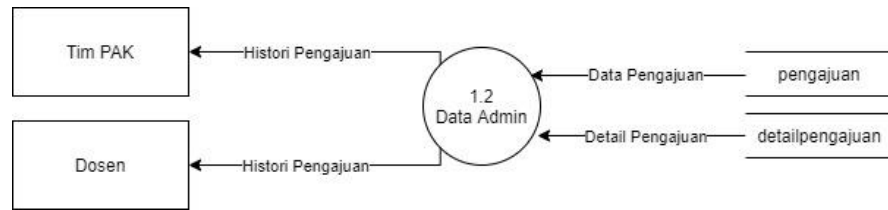
Gambar 4.5 DFD level 1



Gambar 4.6 DFD level 2 proses 1 Master Data



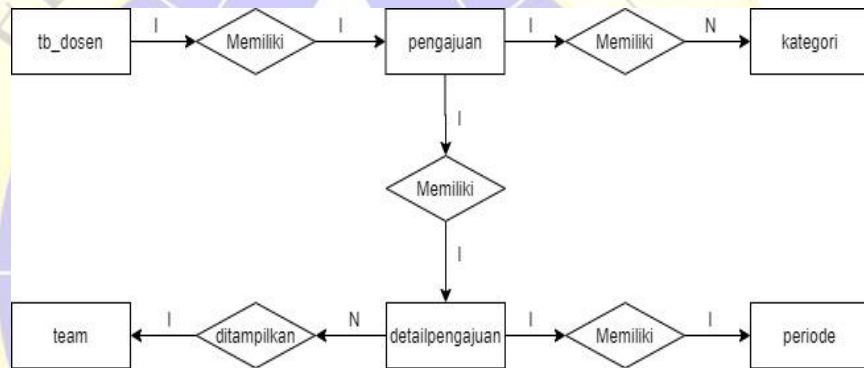
Gambar 4.7 DFD Level 2 Proses 2 pengolahan Data



Gambar 4.8 DFD Level 2 Proses 3 View

#### 4. ERD (Entity Relationship Diagram)

ERD (Entity Relationship Diagram) merupakan suatu diagram yang menjelaskan hubungan antara beberapa entitas dalam suatu basis data. Berikut gambaran dari ERD aplikasi pengarsipan data insiping dosen Universitas Nurul Jadid.



Gambar 4.9 ERD Entity Relationship Diagram

#### Kamus Data:

- team : id, nidn, nama\_dosen, jns\_kelamin, tgl\_lahir, alamat, kontak, fakultas, foto, username, password.
- tb\_dosen : id, nidn, nama\_dosen, jns\_kelamin, tgl\_lahir, alamat, kontak, fakultas, foto, username, password.
- pengajuan : id, nidn, tgl\_pengajuan, status, kode\_pengajuan, periode, komentar.
- periode : id, kode\_periode, nama\_periode, tgl\_mulai, tgl\_selesai, status.
- detailpengajuan : id, kode\_pengajuan, kategori, berkas.
- kategori : id, nama\_berkas.

## 5. Struktur Basis Data

Basis data merupakan wadah atau tempat penyimpanan dari semua data yang dapat diakses dan dikelola oleh pengguna. Berikut struktur basis data yang terhubung dengan sistem yang akan dibangun.

Tabel 4.3 Tabel team

No	Nama	Type
1	Id	bigint(20)
2	Nidn	varchar(50)
3	nam_dosen	varchar(100)
4	jns_kelamin	varchar(1)
5	tgl_lahir	Date
6	Alamat	varchar(100)
7	Kontak	varchar(13)
8	Fakultas	int(11)
9	Foto	varchar(100)
10	Username	varchar(50)
11	Password	varchar(100)
12	Status	varchar(1)

Tabel 4.4 Tabel tb\_dosen

No	Nama	Type
1	Id	bigint(20)
2	Nidn	varchar(50)
3	nam_dosen	varchar(100)
4	jns_kelamin	varchar(1)
5	tgl_lahir	Date
6	Alamat	varchar(100)
7	Kontak	varchar(13)
8	Fakultas	int(11)
9	Foto	varchar(100)
10	Username	varchar(50)

11	Password	varchar(100)
12	Status	varchar(1)

Tabel 4.5 Tabel pengajuan

No	Nama	Type
1	Id	bigint(20)
2	Nidn	varchar(50)
3	tgl_pengajuan	Datetime
4	Status	enum('0', '1', '2', '3')
5	kode_pengajuan	varchar(25)
6	Periode	varchar(25)
7	Komentar	varchar(255)

Tabel 4.6 Tabel periode

No	Nama	Type
1	Id	bigint(20)
2	kode_periode	varchar(25)
3	nama_periode	varchar(100)
4	tgl_mulai	Date
5	tgl_selesai	Date
6	Status	int(11)

Tabel 4.7 Tabel detailpengajuan

No	Nama	Type
1	Id	bigint(20)
2	kode_pengajuan	varchar(100)
3	Berkas	varchar(3)
4	File	varchar(25)

Tabel 4.8 Tabel kategori

No	Nama	Type
1	Id	bigint(20)
2	nama_berkas	varchar(255)

6. Desain Interface

a. Login

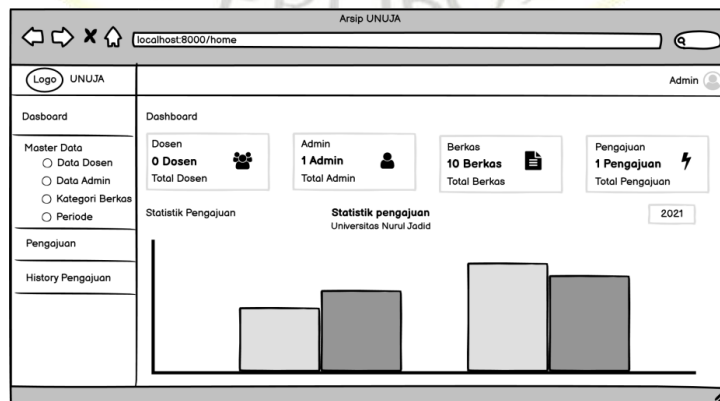
Halaman awal pada aplikasi ini ialah halaman login, yaitu login untuk dosen dan admin. Dosen dapat mengisi username dan password dengan NIDN yang dimiliki, sedangkan untuk admin ialah di isi dengan username dan password yang telah disetting sebelumnya.



Gambar 4.10 Desain Login

b. Halaman Utama (Home)

Setelah login berhasil, maka akan ditampilkan halaman utaman atau Home yang berisi total dari seluruh data didalamnya serta terdapat grafik atau statistika pengajuan.

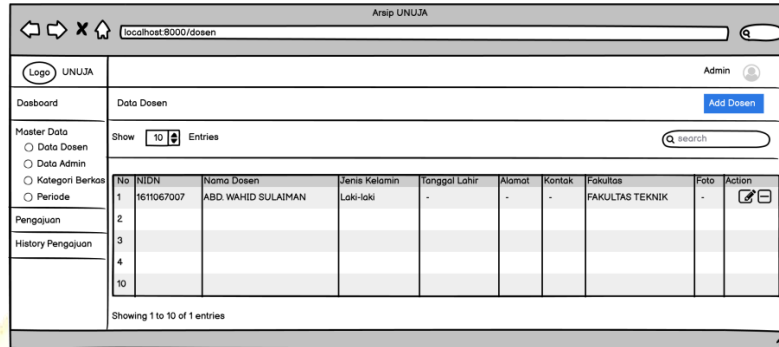


Gambar 4.11 Desain Home



c. Master Data (Data Dosen)

Pada halaman selanjutnya yaitu halaman master data yang ada pada halaman khusus untuk admin, yaitu untuk menginputkan setiap data dosen yang telah melakukan pengajuan inpassing.



Gambar 4.12 Desain Data Dosen

d. Tambah Data Dosen

Setelah menampilkan halaman Data Dosen, didalamnya terdapat tombol tambah data, yaitu admin dapat menambah data dosen yang dibutuhkan.

NIDN	Nama Dosen	Fakultas
Masukan NIDN	Nama Dosen	Pilih Fakultas
Jenis Kelamin	Tanggal Lahir	Kontak
<input type="radio"/> Laki-laki <input type="radio"/> Perempuan		Masukan Kontak
Alamat	Username	Password
Masukan Alamat	Masukan Username	Masukan Password
Foto		
Choose File	No File Chosen	
		Close Save Data

Gambar 4.13 Tambah data Dosen

e. Edit Data Dosen

Pada halaman data dosen juga terdapat tombol Edit Data, yaitu mengubah atau mengupdate data sebelumnya yang belum valid menjadi data yang valid lalu data yang telah diedit dapat disimpan dan akan terdaftar pada halaman data dosen.

**Form Dosen**

NIDN:

Nama Dosen:

Fakultas:

Jenis Kelamin:  Laki-laki  Perempuan

Tanggal Lahir:

Kontak:

Alamat:

Username:

Password:

Foto:  No File Chosen

Gambar 4.14 Edit data Dosen

f. Master Data (Data Team)

Pada master data yang kedua yaitu data team atau bisa disebut juga dengan data admin, yaitu data seluruh admin yang bertugas untuk mengurus pengajuan dan pengarsipan data inpassing dalam setiap periode yang aktif dan dapat mengubah seluruh data didalam aplikasi.

Arsip UNUJA

localhost8000/team

Logo UNUJA Admin

Dashboard Data Admin

Master Data Show 10 Entries search

- Data Dosen
- Data Admin
- Kategori Berkas
- Periode

No	NIDN	Nama Dosen	Jenis Kelamin	Tanggal Lahir	Alamat	Kontak	Fakultas	Username	Password	Foto	Action
1											<input type="button" value="edit"/> <input type="button" value="delete"/>
2											<input type="button" value="edit"/> <input type="button" value="delete"/>
3											<input type="button" value="edit"/> <input type="button" value="delete"/>
4											<input type="button" value="edit"/> <input type="button" value="delete"/>
5											<input type="button" value="edit"/> <input type="button" value="delete"/>

Showing 1 to 10 of 10 entries Previous 1 Next

Gambar 4.15 Data Team

g. Tambah Data Team

Setelah ditampilkan halaman data team, didalamnya terdapat tombol tambah data yaitu memasukkan data admin baru jika terdapat tambahan petugas baru.

The screenshot shows a web form titled "Form Admin" with the following fields and controls:

- NIDN:** A text input field labeled "Masukan NIDN".
- Nama Dosen:** A text input field labeled "Nama Dosen".
- Fakultas:** A dropdown menu labeled "Pilih Fakultas".
- Jenis Kelamin:** Radio buttons for "Laki-laki" and "Perempuan".
- Tanggal Lahir:** A date input field.
- Kontak:** A text input field labeled "Masukan Kontak".
- Alamat:** A text input field labeled "Masukan Alamat".
- Username:** A text input field labeled "Masukan Username".
- Password:** A text input field labeled "Masukan Password".
- Foto:** A file upload area with a "Choose File" button and "No File Chosen" text.

At the bottom right, there are two buttons: a red "Close" button and a blue "Save Data" button.

Gambar 4.16 Tambah data Team

#### h. Edit Data Team

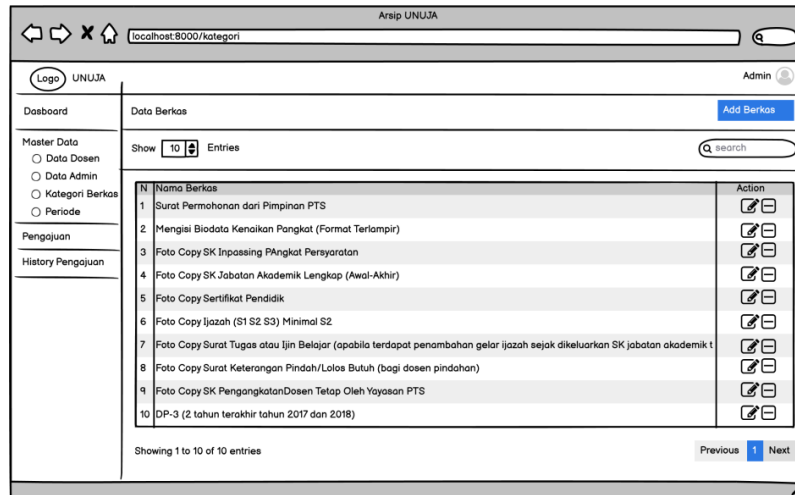
Didalam halaman team juga terdapat tombol edit data, yaitu mengubah atau mengupdate data yang semisal sebelumnya belum lengkap.

This screenshot is identical to the one in Gambar 4.16, showing the "Form Admin" for adding team data. It includes the same set of input fields and controls: NIDN, Nama Dosen, Fakultas, Jenis Kelamin, Tanggal Lahir, Kontak, Alamat, Username, Password, and Foto, with "Close" and "Save Data" buttons at the bottom right.

Gambar 4.17 Edit data Team

#### i. Master Data (Kategori Berkas)

Pada master data yang ketiga yaitu kategori dari setiap berkas persyaratan yang harus dipenuhi oleh setiap dosen ketika akan melakukan pengajuan inpassing dosen.



Gambar 4.17 Kategori Berkas

#### j. Tambah Kategori

Didalam halaman kategori berkas juga terdapat tombol tambah kategori, yaitu memasukan kategori baru jika terdapat pembaruan berkas yang harus dipenuhi oleh dosen pengaju.

Gambar 4.13 Tambah Kategori

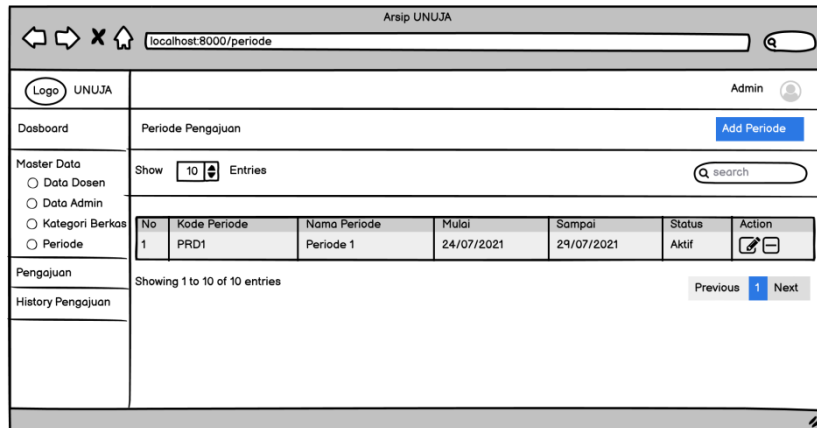
#### k. Edit Kategori

Setelah tombol tambah data terdapat juga tombol edit data, yaitu mengupdate data sebelumnya belum lengkap atau terdapat tambahan fild.

Gambar 4.14 Edit Kategori

#### l. Master Data (Periode)

Master data yang terakhir yaitu periode pengajuan, dalam setiap pengajuan terdapat satu periode aktif dan jika jangka waktu periode aktif telah selesai maka akan diganti dengan periode selanjutnya yang otomatis periode sebelumnya akan non aktif.



Gambar 4.15 Periode

m. Tambah Periode

Didalam halaman periode terdapat tombol tambah periode, yaitu untuk menambah periode baru jika jangka waktu periode sebelumnya telah selesai.

Gambar 4.16 Tambah Periode

n. Edit Periode

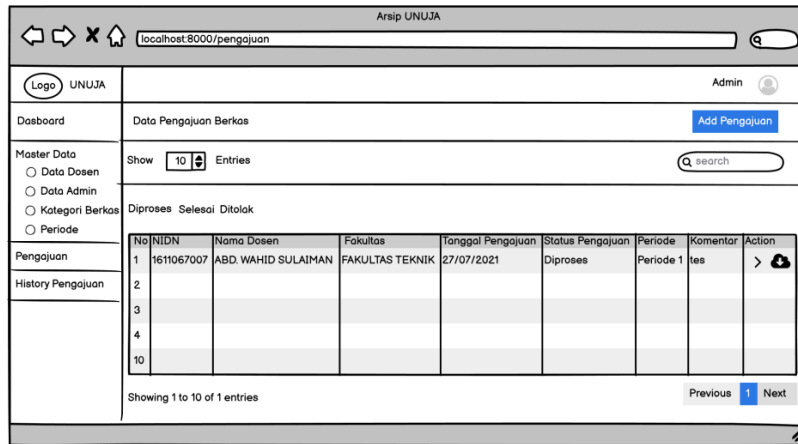
Selanjutnya yaitu tombol edit periode, yaitu untuk mengganti masa aktif dalam setiap periode dengan jangka waktu tertentu.

Gambar 4.17 Edit Periode

o. Pengajuan (Data Pengajuan)

Halaman selanjutnya ialah halaman pengajuan yang dapat diinputkan juga oleh admin dimaksudkan jika suatu hari nanti terdapat salah satu dosen tidak dapat mengajukan inpassing dikarenakan kesalahan jaringan atau semacamnya dan didalamnya juga terdapat tombol detail dan download file dalam bentuk rar/zip.

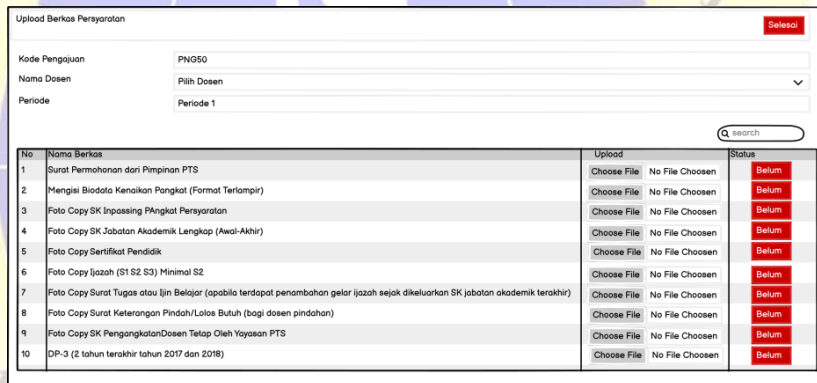




Gambar 4.18 Data Pengajuan

p. Tambah Pengajuan

Didalam halaman pengajuan terdapat tombol tambah pengajuan yaitu sama seperti fungsi sebelumnya, menambah data pengajuan yang dibutuhkan.



Gambar 4.19 Tambah Pengajuan

q. Detail Pengajuan

Didalam halaman pengajuan juga terdapat tombol detail pengajuan, yaitu yaitu tombol untuk melihat kembali file persyaratan apa saja yang telah diupload saat melakukan pengajuan inpassing dosen.

Form Berkas

Kode Pengajuan PNG48

search

No	Nama Berkas	Upload	Status
1	Surat Permohonan dari Pimpinan PTS	Choose File No File Chosen	View Berkas
2	Mengisi Biodata Kenaikan Pangkat (Format Terlampir)	Choose File No File Chosen	View Berkas
3	Foto Copy SK Inpassing PAngkat Persyaratan	Choose File No File Chosen	View Berkas
4	Foto Copy SK Jabatan Akademik Lengkap (Awal-Akhir)	Choose File No File Chosen	View Berkas
5	Foto Copy Sertifikat Pendidik	Choose File No File Chosen	View Berkas
6	Foto Copy Ijazah (S1 S2 S3) Minimal S2	Choose File No File Chosen	View Berkas
7	Foto Copy Surat Tugas atau Ijin Belajar (apabila terdapat penambahan gelar ijazah sejak dikeluarkan SK jabatan akademik te	Choose File No File Chosen	View Berkas
8	Foto Copy Surat Keterangan Pindah/Lolos Butuh (bagi dosen pindahan)	Choose File No File Chosen	View Berkas
9	Foto Copy SK PengangkatanDosen Tetap Oleh Yayasan PTS	Choose File No File Chosen	View Berkas
10	DP-3 (2 tahun terakhir tahun 2017 dan 2018)	Choose File No File Chosen	View Berkas

Gambar 4.20 Detail Pengajuan

r. Status Pengajuan

Tombol selanjutnya yaitu tombol status pengajuan, yaitu untuk menentukan apakah pengajuan yang dilakukan oleh dosen masih diproses, sudah selesai atau ditolak dikarenakan kelengkapan data yang belum valid yang nantinya akan muncul juga pada halaman pengajuan untuk dosen.

Form Berkas

Status Pengajuan

Diproses

Komentar

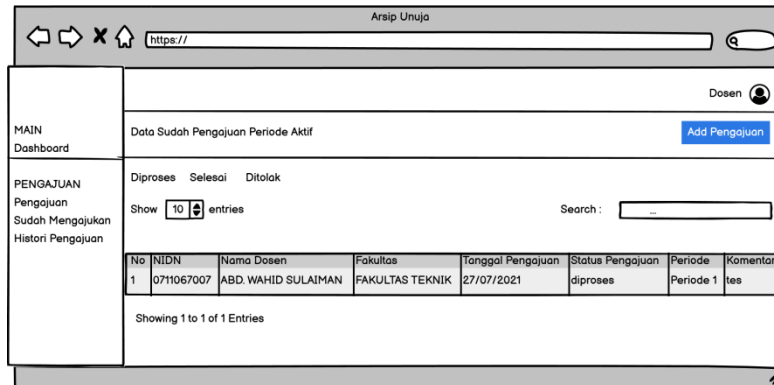
Data Masih Diproses

Close Save Data

Gambar 4.21 Status Pengajuan

s. Sudah Mengajukan (dosen)

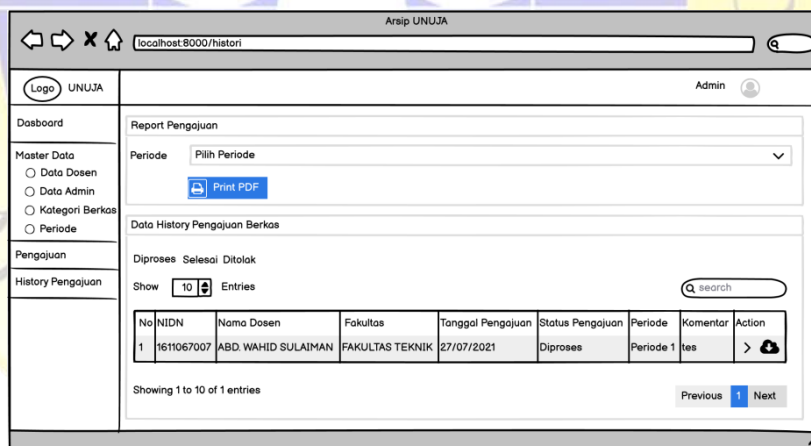
Pada halaman ini terdapat data dari seluruh dosen yang telah mengajukan serta status dari setiap pengajuan yang ada.



Gambar 4.22 Sudah Mengajukan

t. View (Histori Pengajuan)

Halaman terakhir yaitu halaman histori atau riwayat dari semua pengajuan yang telah diinputkan baik itu dari dosen yang bersangkutan atau dari admin sendiri, halaman histori untuk admin terdapat tombol report pdf jika nanti dibutuhkan untuk laporan, sedangkan untuk halaman dosen hanya terdapat data riwayat dari satu dosen itu sendiri dan juga terdapat download data dalam bentuk rar/zip.



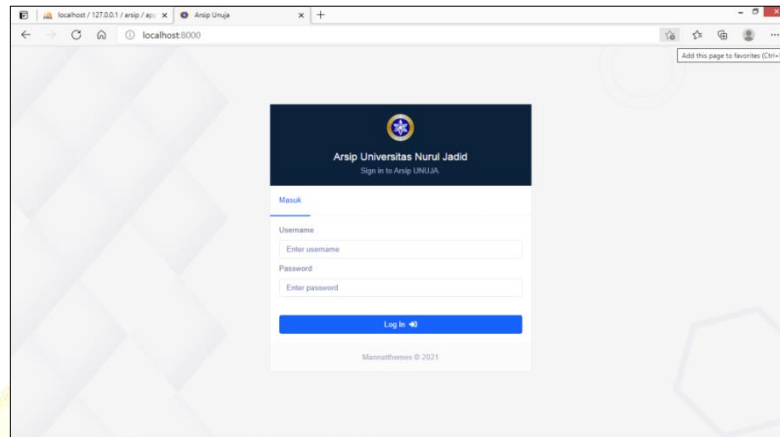
Gambar 4.22 Histori Pengajuan

### 4.3 Implementation

Setelah tahap perancangan alur dan desain sistem selesai, maka langkah selanjutnya ialah mengimplementasikan desain yang telah dibuat kepada sistem. Dalam penelitian ini implementasi desain sistem menggunakan aplikasi Visual Studio Code dan untuk desain databasenya menggunakan MySql. Aplikasi yang dibangun ialah aplikasi pengarsipan data inpassing dosen Universitas Nurul Jadid.

Berikut hasil implementasi dari aplikasi pengarsipan data inpassing dosen Universitas Nurul Jadid :

1. Halaman Login (admin dan dosen)



Gambar 4.23 Tampilan Login

**Segmen Program 4.1** Tampilan Login

```
def login(request):
    return render(request, "login.html")
@csrf_exempt
def auth(request):
    arr = 0
    if request.method == "POST":
        username = request.POST.get("username")
        password = request.POST.get("password")
        auth = (
            Dosen.objects.filter(username=username).filter(
                password=password).exists()
        )
    if auth:
        data = (
            Dosen.objects.filter(username=username).filter(
                password=password).all()
        )
    for val in data.iterator():
        request.session["id"] = val.id
```

```

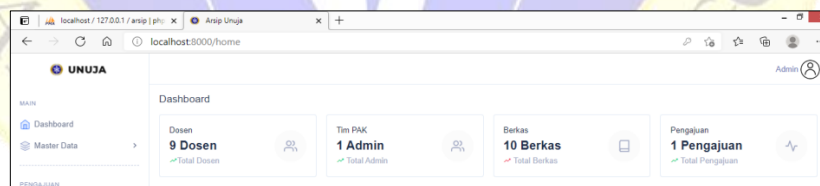
request.session["nidn"] = val.nidn
request.session["nama"] = val.nama_dosen
request.session["tgl"] = val.tgl_lahir.strftime("%Y-%m-%d")

request.session["gender"] = val.jns_kelamin
request.session["alamat"] = val.alamat
request.session["kontak"] = val.kontak
request.session["fakultas"] = val.fakultas
request.session["foto"] = MEDIA_URL + str(val.foto)
request.session["username"] = val.username
request.session["password"] = val.password
request.session["status"] = val.status
request.session["login"] = True

arr = 1
else:
    arr = 2
else:
    arr = 3
return HttpResponse(arr)

```

## 2. Halaman Utama (admin dan dosen)



Gambar 4.24 Tampilan Home

### Segmen Program 4.2 Tampilan Home

```

def home(request):
    if request.session["nidn"] == None:
        return render(request, "index.html")
    else:
        dosen = Dosen.objects.filter(status="D").all().count()
        tim = Dosen.objects.filter(status="A").all().count()

```



```

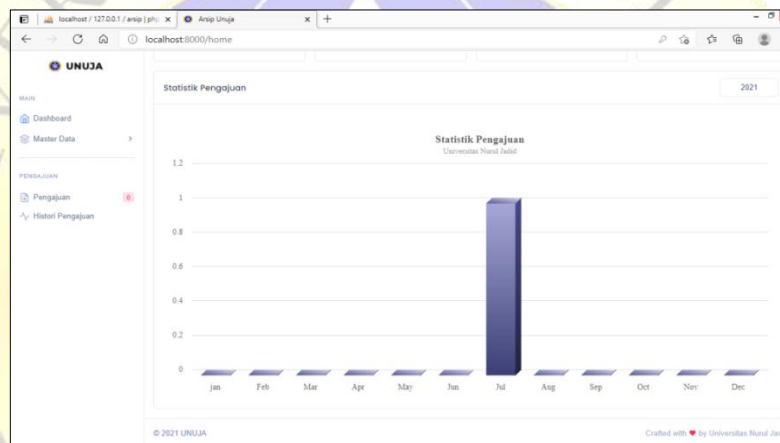
berkas = Berkas.objects.all().count()
pengajuan = Pengajuan.objects.all().count()

count = {
    "dosen": dosen,
    "pengajuan": pengajuan,
    "berkas": berkas,
    "tim": tim,
    "url": "home",
}

return render(request, "home.html", count)

```

### 3. Grafik (Statistik Pengajuan)



Gambar 4.25 Tampilan Grafik

#### Segmen Program 4.3 Tampilan Grafik

```
@csrf_exempt
```

```
def vchart(request):
```

```
    thn = request.POST.get("thn")
```

```
    bulan = [
```

```
        "jan",
```

```
        "Feb",
```

```
        "Mar",
```

```
        "Apr",
```

```
        "May",
```

```
        "Jun",
```

```
        "Jul",
```

```

"Aug",
"Sep",
"Oct",
"Nov",
"Dec",
]
bln = [
"01",
"02",
"03",
"04",
"05",
"06",
"07",
"08",
"09",
"10",
"11",
"12",
]
idx = 0
value = {}
data = []
for bl in bulan:

# asisten ahli
with connection.cursor() as con:
con.execute(
"SELECT COUNT(id) as total FROM `app_pengajuan`
WHERE substr(tgl_pengajuan,1,7)=%s",
[thn + "-" + bln[idx]],
)

```

```

row = con.fetchone()
val = str(row[0])

```

```

value = {'label': bl, 'value': val}

```

```

data.append(value)

```

```

idx += 1

```

```

output = json.dumps(

```

```

{"data": data})

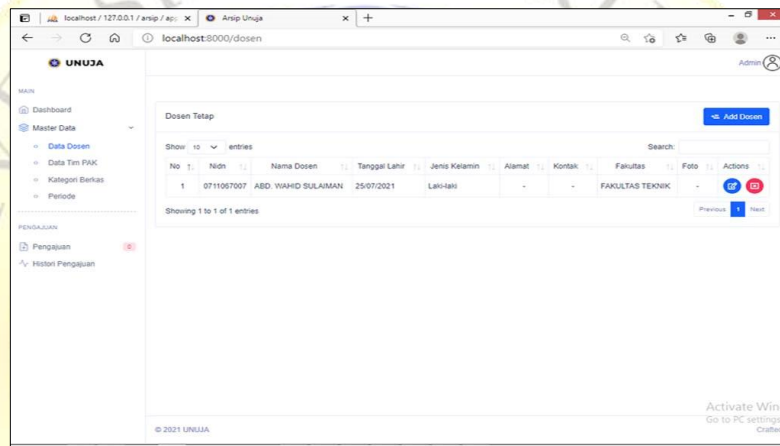
```

```

return HttpResponse(output, content_type="application/json")

```

#### 4. Master Data (Data Dosen)



Gambar 4.26 Tampilan Data Dosen

#### Segmen Program 4.4 Tampilan Data Dosen

```

def dosen(request):

```

```

    if request.session["nidn"] == None:

```

```

        return render(request, "index.html")

```

```

    else:

```

```

        return render(request, "dosen.html")

```

#### 5. Tambah Data Dosen

Gambar 4.27 Tampilan Tambah Data Dosen

### Segmen Program 4.5 Tampilan Data Dosen

@csrf\_exempt

def vfakultas(request):

    sql = Fakultas.objects.all()

    data = "<option value="">Pilih Fakultas</option>"

    for val in sql.iterator():

        data += (

            '<option value=""

            + str(val.id)

            + "">'

            + val.nama\_fakultas.upper()

            + "</option>"

        )

    return HttpResponse(data)

### 6. Edit Data Dosen

Gambar 4.28 Tampilan Edit Data Dosen

### Segmen Program 4.6 Tampilan Edit Data Dosen

@csrf\_exempt

def svdosen(request):

    if request.method == "POST":

        pk = request.POST.get("id")

        nidn = request.POST.get("nidn")

        nama = request.POST.get("nama\_dosen")

        gender = request.POST.get("jns\_kelamin")

        tgl = request.POST.get("tgl\_lahir")

        alamat = request.POST.get("alamat")

        kontak = request.POST.get("kontak")

        fakultas = request.POST.get("fakultas")

        username = request.POST.get("username")

        password = request.POST.get("password")

        status = request.POST.get("status")

        if pk != "":

            editdata = Dosen.objects.get(pk=pk)

            save = DosenForm(

                request.POST or None, request.FILES or None,

                instance=editdata

            )

            save.save()

            arr = 1

        else:

            save = DosenForm(request.POST or None, request.FILES or

None)

        if save.is\_valid:

            if tgl == "":

                tgl = None

            else:

                tgl = tgl

            if alamat == "":

```
        alamat = None
    else:
        alamat = alamat
    if kontak == "":
        kontak = None
    else:
        kontak = kontak
    if fakultas == "":
        fakultas = None
    else:
        fakultas = fakultas
    if username == "":
        username = nidn
    else:
        username = username
    if password == "":
        password = nidn
    else:
        password = password
    if request.POST.get("foto1") == "":
        save = Dosen(
            nidn=nidn,
            nama_dosen=nama,
            jns_kelamin=gender,
            tgl_lahir=tgl,
            alamat=alamat,
            fakultas=fakultas,
            username=username,
            password=password,
            foto="0",
            status=status,
        )
```



else:

```
save = Dosen(  
    nidn=nidn,  
    nama_dosen=nama,  
    jns_kelamin=gender,  
    tgl_lahir=tgl,  
    alamat=alamat,  
    fakultas=fakultas,  
    username=username,  
    password=password,  
    foto=request.FILES["foto"],  
    status=status,  
)
```

```
save.save()
```

```
arr = 1
```

```
else:
```

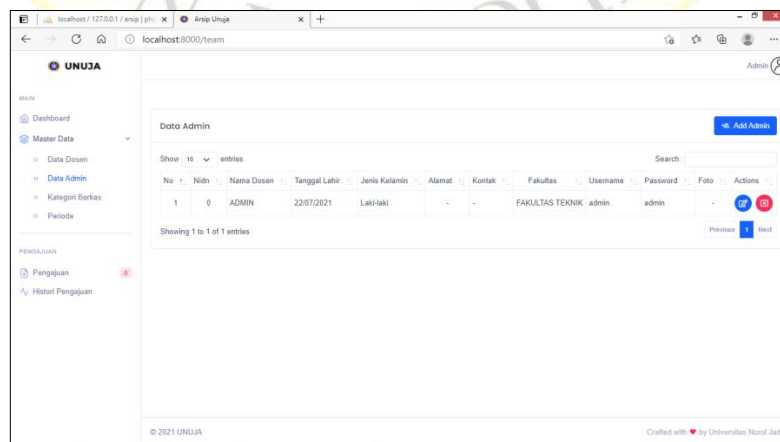
```
arr = 0
```

```
else:
```

```
arr = 0
```

```
return HttpResponse(arr)
```

## 7. Master Data (Data Admin)

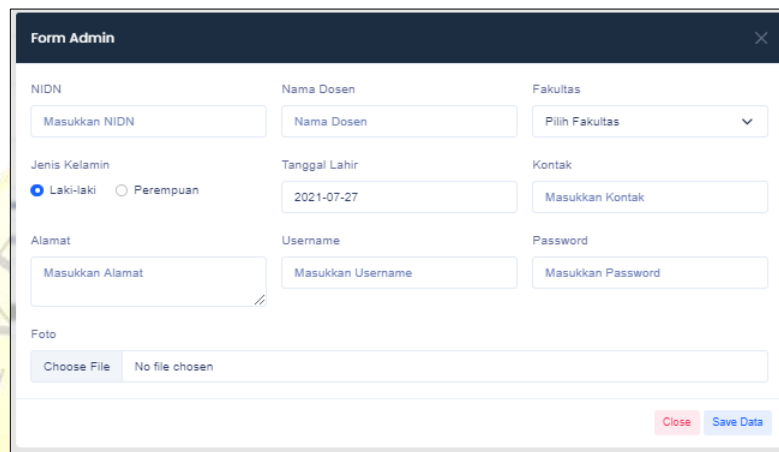


Gambar 4.29 Tampilan Data Admin

### Segmen Program 4.7 Data Admin

```
def team(request):  
    if request.session["nidn"] == None:  
        return render(request, "index.html")  
    else:  
        return render(request, "team.html")
```

### 8. Tambah Data Admin

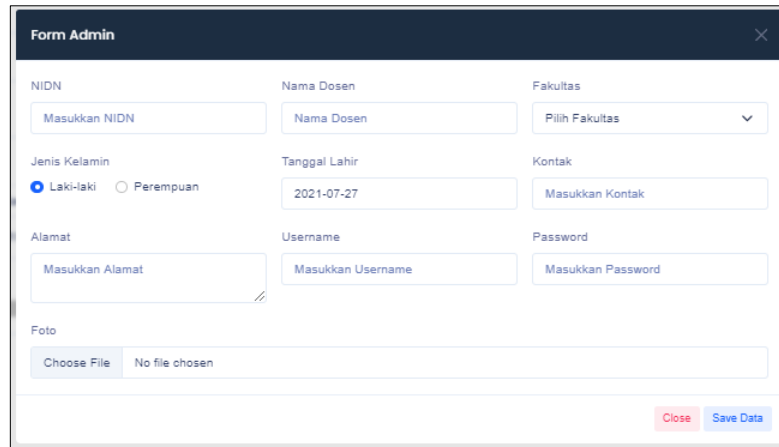


Gambar 4.30 Tampilan Tambah Data Admin

### Segmen Program 4.8 Tambah Data Admin

```
@csrf_exempt  
def vfakultas(request):  
    sql = Fakultas.objects.all()  
    data = "<option value="">Pilih Fakultas</option>"  
    for val in sql.iterator():  
        data += (  
            '<option value="'  
            + str(val.id)  
            + "'>'  
            + val.nama_fakultas.upper()  
            + "</option>"  
        )  
    return HttpResponse(data)
```

## 9. Edit Data Admin



Gambar 4.31 Tampilan Edit data Admin

### Segmen Program 4.9 Tampilan Edit Data Admin

@csrf\_exempt

```
def svdosen(request):  
    if request.method == "POST":  
        pk = request.POST.get("id")  
        nidn = request.POST.get("nidn")  
        nama = request.POST.get("nama_dosen")  
        gender = request.POST.get("jns_kelamin")  
        tgl = request.POST.get("tgl_lahir")  
        alamat = request.POST.get("alamat")  
        kontak = request.POST.get("kontak")  
        fakultas = request.POST.get("fakultas")  
        username = request.POST.get("username")  
        password = request.POST.get("password")  
        status = request.POST.get("status")  
        if pk != "":  
            editdata = Dosen.objects.get(pk=pk)  
            save = DosenForm(  
                request.POST or None, request.FILES or None,  
                instance=editdata  
            )  
            save.save()
```

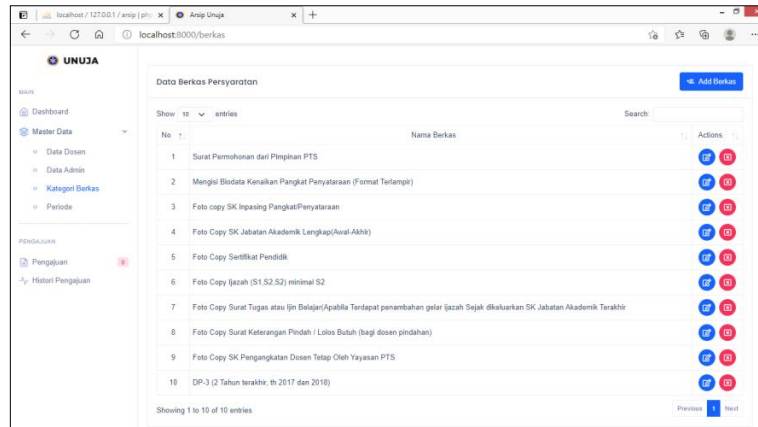
```

arr = 1
else:
    save = DosenForm(request.POST or None, request.FILES or
None)
    if save.is_valid:
        if tgl == "":
            tgl = None
        else:
            tgl = tgl
        if alamat == "":
            alamat = None
        else:
            alamat = alamat
        if kontak == "":
            kontak = None
        else:
            kontak = kontak
        if fakultas == "":
            fakultas = None
        else:
            fakultas = fakultas
        if username == "":
            username = nidn
        else:
            username = username
        if password == "":
            password = nidn
        else:
            password = password
        if request.POST.get("foto1") == "":
            save = Dosen(
                nidn=nidn,

```

```
        nama_dosen=nama,
        jns_kelamin=gender,
        tgl_lahir=tgl,
        alamat=alamat,
        fakultas=fakultas,
        username=username,
        password=password,
        foto="0",
        status=status,
    )
    else:
        save = Dosen(
            nidn=nidn,
            nama_dosen=nama,
            jns_kelamin=gender,
            tgl_lahir=tgl,
            alamat=alamat,
            fakultas=fakultas,
            username=username,
            password=password,
            foto=request.FILES["foto"],
            status=status,
        )
        save.save()
        arr = 1
    else:
        arr = 0
else:
    arr = 0
return HttpResponse(arr)
```

## 10. Master Data (Kategori Berkas)



Gambar 4.32 Tampilan Kategori Berkas

### Segmen Program 4.10 Tampilan Kategori Berkas

def berkas(request):

```
    if request.session["nidl"] == None:
```

```
        return render(request, "index.html")
```

```
    else:
```

```
        return render(request, "berkas.html")
```

```
@csrf_exempt
```

```
def hberkas(request):
```

```
    if request.method == "POST":
```

```
        pk = request.POST.get("id")
```

```
        data = Berkas.objects.get(pk=pk)
```

```
        sql = data.delete()
```

```
    if sql:
```

```
        arr = 1
```

```
    else:
```

```
        arr = 0
```

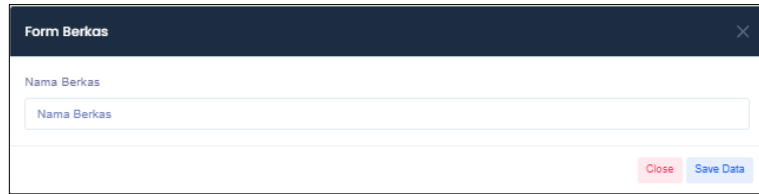
```
    else:
```

```
        arr = 0
```

```
    return HttpResponseRedirect(arr)
```

## 11. Tambah Berkas





Gambar 4.33 Tampilan Tambah Berkas

**Segmen Program 4.11** Tampilan Tambah Berkas

@csrf\_exempt

def svberkas(request):

    if request.method == "POST":

        pk = request.POST.get("id")

        if pk != "":

            editdata = Berkas.objects.get(pk=pk)

            save = BerkasForm(request.POST, instance=editdata)

        else:

            save = BerkasForm(request.POST or None)

        if save.is\_valid:

            sql = save.save()

            if sql:

                arr = 1

            else:

                arr = 0

        else:

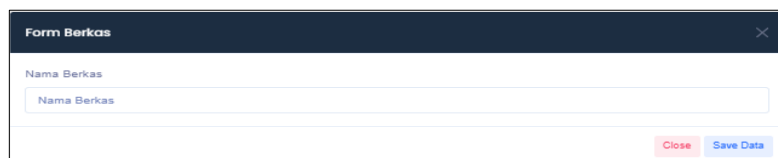
            arr = 0

    else:

        arr = 0

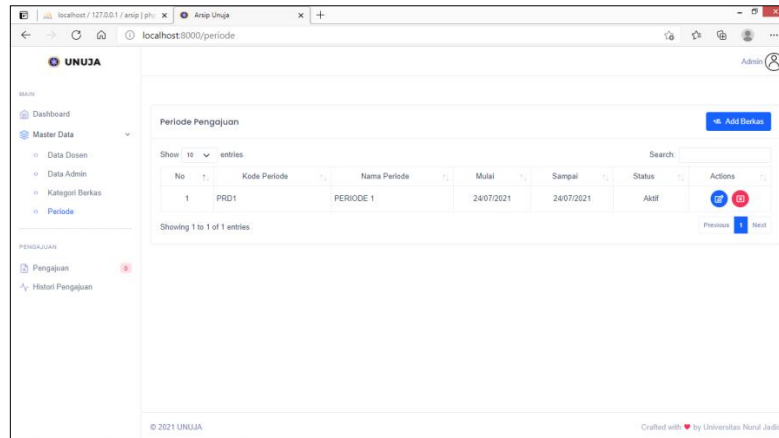
    return HttpResponseRedirect(arr)

12. Edit Berkas



Gambar 4.34 Tampilan Edit Berkas

### 13. Master Data (Periode)



Gambar 4.35 Tampilan Periode

#### Segmen Program 4.12 Tampilan Periode

def periode(request):

    if request.session["nidn"] == None:

        return render(request, "index.html")

    else:

        return render(request, "periode.html")

@csrf\_exempt

def hperiode(request):

    if request.method == "POST":

        pk = request.POST.get("id")

        data = Periode.objects.get(pk=pk)

        sql = data.delete()

        if sql:

            arr = 1

        else:

            arr = 0

    else:

        arr = 0

    return HttpResponse(arr)

### 14. Tambah Periode

Gambar 4.36 Tampilan Tambah Periode

**Segmen Program 4.13** Tampilan Tambah Periode

@csrf\_exempt

def cekkode(request):

    with connection.cursor() as con:

        con.execute("SELECT \* FROM `app\_parameter` WHERE status='periode'")

        row = con.fetchone()

        kode = str(row[2]) + str(row[1])

    with connection.cursor() as con:

        con.execute(

            "UPDATE `app\_parameter` SET nomor=nomor+1 WHERE status='periode'")

        return HttpResponse(kode)

15. Edit Periode

Gambar 4.37 Tampilan Edit Periode

**Segmen Program 4.14** Tampilan Edit Periode

@csrf\_exempt

def svperiode(request):

    if request.method == "POST":

        pk = request.POST.get("id")

        if pk != "":

            editdata = Periode.objects.get(pk=pk)

            save = PeriodeForm(request.POST, instance=editdata)

```

else:
    save = PeriodeForm(request.POST or None)
if save.is_valid:
    with connection.cursor() as con:
        con.execute("UPDATE app_periode SET status='0'
WHERE 1")
    sql = save.save()
    if sql:
        arr = 1
    else:
        arr = 0
else:
    arr = 0
else:
    arr = 0
return HttpResponse(arr)
@csrf_exempt
def vperiode(request):
    with connection.cursor() as con:
        con.execute("SELECT * FROM app_periode ORDER BY
status DESC")
    data = []
    out = []
    no = 1
    for val in con.fetchall():
        nom = "<center>" + str(no) + "</center>"
        kode = val[1]
        nama = val[2].upper()
        mulai = "<center>" + val[3].strftime("%d/%m/%Y") +
"</center>"
        sampai = "<center>" + val[4].strftime("%d/%m/%Y") +
"</center>"

```

```

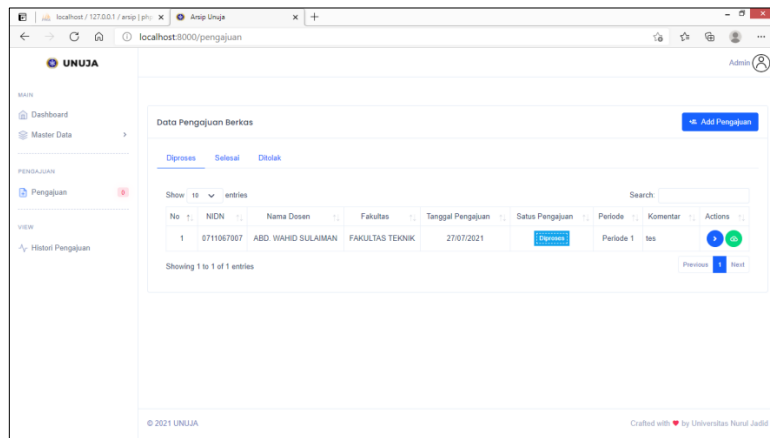
if val[5] == 1:
    status = "<center>Aktif</center>"
else:
    status = "<center>Non Aktif</center>"
button = (
    '<center><a href="javascript:ePeriode('
    + str(val[0])
    + ')" title = "Edit details" class="btn btn-primary btn-icon-
circle btn-icon-circle-sm"><i class="far fa-edit"></i>'
)
button += "</a> &nbsp;"
button += (
    '<a href="javascript:hPeriode('
    + str(val[0])
    + ')" title = "Delete" class="btn btn-danger btn-icon-circle
btn-icon-circle-sm"><i class="far fa-window-close"></i>'
)
button += "</a></center>"
out = [
    nom,
    kode,
    nama,
    mulai,
    sampai,
    status,
    button,
]
data.append(out)
no += 1

output = json.dumps({"data": data})

```

```
return HttpResponse(output, content_type="application/json")
```

## 16. Pengajuan (admin)



Gambar 4.38 Tampilan Data Pengajuan

## Segmen Program 4.15 Tampilan Data Pengajuan

```
def pengajuan(request):
```

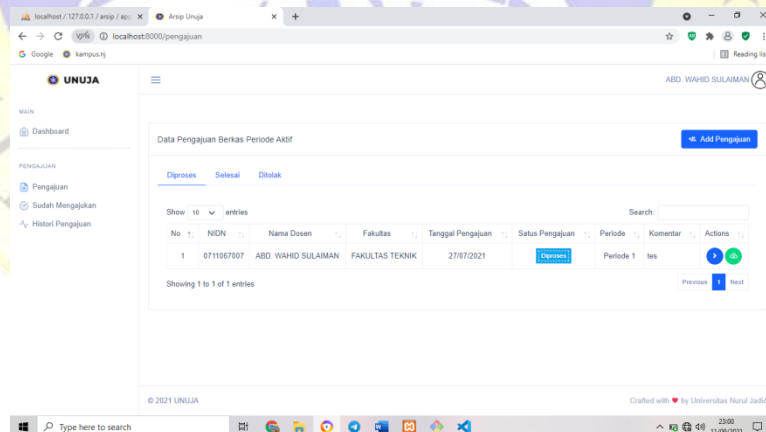
```
    if request.session["nidn"] == None:
```

```
        return render(request, "index.html")
```

```
    else:
```

```
        return render(request, "pengajuan.html")
```

## 17. Pengajuan (dosen)



Gambar 4.39 Tampilan Data Pengajuan (Dosen)

## Segmen Program 4.16 Tampilan Data Pengajuan (Dosen)

```
def pengajuan(request):
```

```
    if request.session["nidn"] == None:
```

```
        return render(request, "index.html")
```

```
    else:
```



```
return render(request, "pengajuan.html")
```

```
def download(request):
```

```
    """
```

A django view to zip files in directory and send it as downloadable response to the browser.

Args:

@request: Django request object

@file\_name: Name of the directory to be zipped

Returns:

A downloadable Http response

```
    """
```

```
    nidn = request.GET["nidn"]
```

```
    periode = request.GET["periode"]
```

```
    file_name = nidn + "_" + periode
```

```
    files_path = MEDIA_URL + "berkas/" + nidn + "/" + periode +  
    "/"
```

```
    path_to_zip = make_archive(files_path, "zip", files_path)
```

```
    response = HttpResponse(  
        FileWrapper(open(path_to_zip, "rb")),  
        content_type="application/zip"  
    )
```

```
        response["Content-Disposition"] = 'attachment;  
filename="{filename}.zip".format(  
    filename=file_name.replace(" ", "_")  
    )
```

```
    )
```

```
    response["Content-Disposition"] = 'attachment;  
filename="{filename}.zip".format(  
        filename=file_name.replace(" ", "_")  
    )
```

```
    response["Content-Disposition"] = 'attachment;  
filename="{filename}.zip".format(  
        filename=file_name.replace(" ", "_")  
    )
```

```
    response["Content-Disposition"] = 'attachment;  
filename="{filename}.zip".format(  
        filename=file_name.replace(" ", "_")  
    )
```

```
    )
```

```
    return response
```



```

def getperiode(request):
    sql = Periode.objects.filter(status="1").all()
    data = ""
    for val in sql.iterator():
        data += (
            '<option value="'
            + str(val.kode_periode)
            + "'>'
            + val.nama_periode.upper()
            + " ("
            + val.tgl_mulai.strftime("%Y")
            + ")"
            + "</option>"
        )
    return HttpResponse(data)

@csrf_exempt
def vuplodberkas(request):
    kode = request.POST.get("kode")
    nidn = request.POST.get("nidn")
    periode = request.POST.get("periode")
    with connection.cursor() as con:
        con.execute("SELECT * FROM app_berkas")
        data = []
        out = []
        no = 1
        for val in con.fetchall():
            nom = "<center>" + str(no) + "</center>"
            nama = val[1]
            upload = (
                '<center><input type="file" class="form-control"
                accept="application/pdf" id="pdf" data-value="'

```

```

+ str(val[0])
+ "' name='pdf' style='width:200px'></center>'
)
with connection.cursor() as query:
    query.execute(
        "SELECT o.kode_periode,d.file FROM
app_detailpengajuan d LEFT JOIN app_pengajuan p ON
d.kode_pengajuan=p.kode_pengajuan LEFT JOIN app_periode o
ON o.kode_periode=p.periode WHERE p.nidn='"
+ nidn
+ "' AND p.kode_pengajuan='"
+ kode
+ "' AND d.berkas='"
+ str(val[0])
+ "' AND o.kode_periode='"
+ periode
+ "'"
    )
    if query.rowcount > 0:
        row = query.fetchone()
        status = (
            "<center><a href='"
            + MEDIA_URL
            + "berkas/"
            + nidn
            + "/"
            + row[0]
            + "/"
            + str(row[1])
            + "' target='_blank' class='badge bg-primary'>View
Berkas</a></center>"
        )

```

```

else:
    status = (
        "<center><span class='badge bg-
danger'>Belum</span></center>"
    )
    button = (
        '<center><a href="javascript:hBerkas('
        + str(val[0])
        + ')" title = "Delete" class="btn btn-danger btn-icon-circle
btn-icon-circle-sm"><i class="far fa-window-close"></i>'
    )
    button += "</a></center>"
    out = [
        nom,
        nama,
        upload,
        status,
    ]
    data.append(out)
    no += 1
    output = json.dumps({"data": data})
    return HttpResponse(output, content_type="application/json")
@csrf_exempt
def cekpengajuan(request):
    with connection.cursor() as con:
        con.execute("SELECT * FROM `app_parameter` WHERE
status='pengajuan'")
        row = con.fetchone()
        kode = str(row[2]) + str(row[1])
    with connection.cursor() as con:
        con.execute(

```



```

        "UPDATE `app_parameter` SET nomor=nomor+1 WHERE
status='pengajuan')
    return HttpResponse(kode)
@csrf_exempt
def svpengajuan(request):
    if request.method == "POST":
        kode = request.POST.get("kode")
        nidn = request.POST.get("dosen")
        periode = request.POST.get("periode")
        berkas = request.POST.get("berkas")
        cek = Pengajuan.objects.filter(kode_pengajuan=kode).all().count()
        if cek == 0:
            save = PengajuanForm(request.POST or None)
            if save.is_valid():
                success = Pengajuan(
                    nidn=nidn,
                    tgl_pengajuan=datetime.now(),
                    status="1",
                    kode_pengajuan=kode,
                    periode=periode,
                )
                if success.save():
                    arr = 1
                else:
                    arr = 0
            else:
                arr = 2
        # create folder
        directory = MEDIA_URL + "berkas/" + nidn + "/" + periode +
        "/"
        path = "berkas/" + nidn + "/" + periode + "/"

```

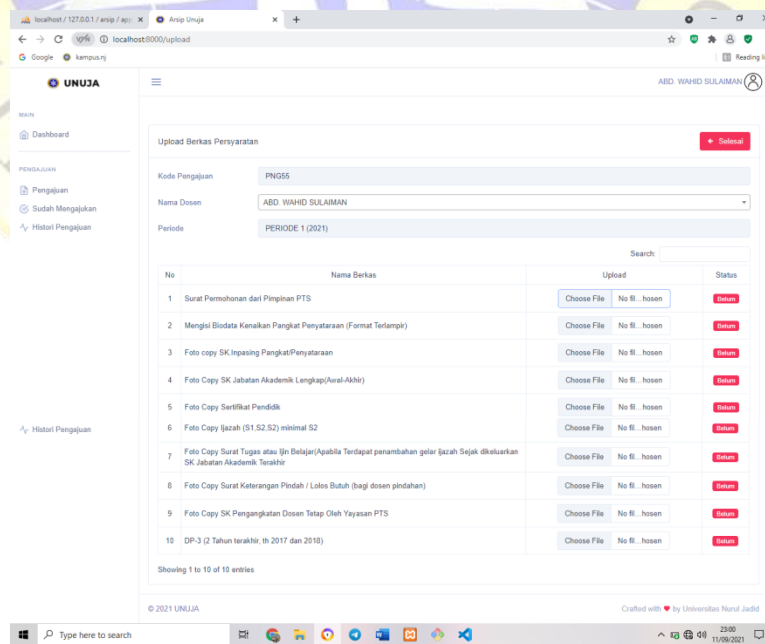


```

if not os.path.exists(directory):
    os.makedirs(directory)
# upload file folder
nama = Berkas.objects.filter(id=berkas).all()
for br in nama.iterator():
    filename = (
        str(br.nama_berkas)
        .replace(" ", "_")
        .replace("/", "_")
        .replace(".", "_")
    )
    file = request.FILES["file"].name
    ext = file.split(".")[1]
    full_filename = os.path.join(MEDIA_URL, path, filename + "."
+ ext)
    # hapus File
    if os.path.isfile(full_filename):
        os.remove(full_filename)

```

### 19. Tambah Pengajuan (dosen)



Gambar 4.41 Tambah Pengajuan untuk dosen

#### Segmen Program 4.18 Tampilan Tambah Pengajuan (Dosen)

```
@csrf_exempt
def vpengajuan(request):
    status = request.POST.get("status")
    nidn = request.session["nidn"]
    po = Periode.objects.filter(status="1").all()
    for p in po.iterator():
        kodepo = p.kode_periode
    if request.session["status"] == "A":
        query = (
            Pengajuan.objects.filter(status=status, periode=kodepo)
            .all()
            .order_by("-tgl_pengajuan")
        )
    else:
        query = (
            Pengajuan.objects.filter(status=status, nidn=nidn,
            periode=kodepo)
            .all()
            .order_by("-tgl_pengajuan")
        )
    data = []
    out = []
    no = 1
    pengajuan = ""
    for val in query.iterator():
        nom = "<center>" + str(no) + "</center>"
        dosen = Dosen.objects.filter(nidn=val.nidn).all()
        for ds in dosen.iterator():
            nama = "<center>" + ds.nama_dosen + "</center>"
            fakul = Fakultas.objects.filter(id=ds.fakultas).all()
            for fl in fakul.iterator():
```

```

        fakultas = "<center>" + fl.nama_fakultas + "</center>"
        tgl = "<center>" +
val.tgl_pengajuan.strftime("%d/%m/%Y") + "</center>"
        if val.status == "0":
            pengajuan = (
                '<center><button type="button" onclick="upstatus('
                + str(val.id)
                + ')" class="btn btn-primary btn-square btn-outline-
dashed btn-sm">Pengajuan</button><center>'
            )
        elif val.status == "1":
            pengajuan = (
                '<center><button type="button" onclick="upstatus('
                + str(val.id)
                + ')" class="btn btn-info btn-square btn-outline-dashed
btn-sm">Diproses</button><center>'
            )
        elif val.status == "2":
            pengajuan = (
                '<center><button type="button" onclick="upstatus('
                + str(val.id)
                + ')" class="btn btn-success btn-square btn-outline-
dashed btn-sm">Selesai</button><center>'
            )
        elif val.status == "3":
            pengajuan = (
                '<center><button type="button" onclick="upstatus('
                + str(val.id)
                + ')" class="btn btn-danger btn-square btn-outline-
dashed btn-sm">Ditolak</button><center>'
            )

```

```

period = Periode.objects.filter(kode_periode=val.periode)
for pr in period.iterator():
    periode = "<center>" + pr.nama_periode + "</center>"
    kodeperiode = pr.kode_periode
    if val.komentar == None:
        komentar = '<center>-<komentar>'
    else:
        komentar = val.komentar

    button = (
        "<center>"
        + '<a href="javascript:vBerkas('
        + str(val.id)
        + ')" title = "Detail" class="btn btn-primary btn-icon-circle
        btn-icon-circle-sm"><i class="fas fa-angle-right"></i></a>'
        + '&nbsp;<button id="download" data-nidn="'
        + val.nidn
        + '" data-periode="'
        + kodeperiode
        + '" title = "Download File" class="btn btn-success btn-
        icon-circle btn-icon-circle-sm"><i class="mdi mdi-cloud-
        download-outline"></i></button>'
        )
    button += "</center>"
    out = [
        nom,
        "<center>" + val.nidn + "</center>",
        nama,
        fakultas,
        tgl,
        pengajuan,
        periode,

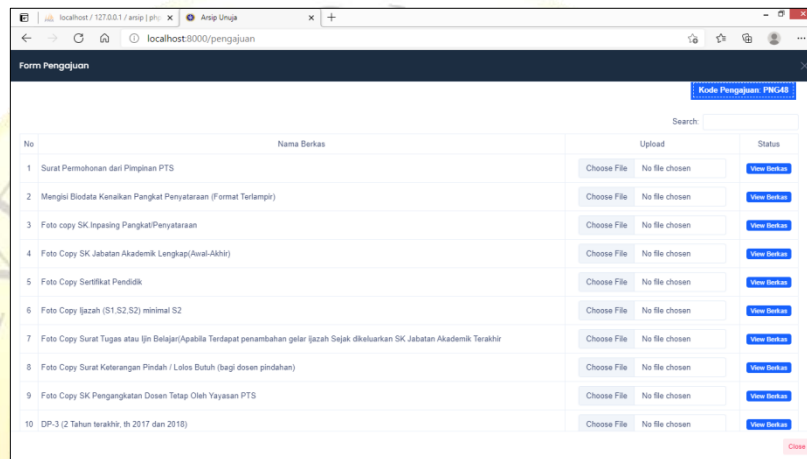
```

```

    komentar,
    button,
]
data.append(out)
no += 1
output = json.dumps({"data": data})
return HttpResponse(output, content_type="application/json")

```

## 20. Detail Pengajuan (admin dan dosen)



Gambar 4.42 Tampilan Detail Pengajuan

### Segmen Program 4.19 Tampilan Detail Pengajuan

```

# save file to folder
fout = open(full_filename, "wb+")
file_content = ContentFile(request.FILES["file"].read())
for chunk in file_content.chunks():
    fout.write(chunk)
fout.close()
cekberkas = (
    DetailPengajuan.objects.filter(kode_pengajuan=kode,
berkas=berkas)
    .all()
    .count()
)
if cekberkas == 0:
    dtl = DetailPengajuan(

```

```

        kode_pengajuan=kode, berkas=berkas, file=filename + "."
+ ext
    )
    dtl.save()
    arr = 1
    return HttpResponse(arr)
@csrf_exempt
def cekberkas(request):
    id = request.POST.get("id")
    query = Pengajuan.objects.filter(id=id).all()
    for val in query.iterator():
        kode = val.kode_pengajuan
        nidn = val.nidn
    return HttpResponse(kode + "|" + nidn)
@csrf_exempt
def showberkas(request):
    kode = request.POST.get("kode")
    nidn = request.POST.get("nidn")
    with connection.cursor() as con:
        con.execute("SELECT * FROM app_berkas")
    data = []
    out = []
    no = 1
    for val in con.fetchall():
        nom = "<center>" + str(no) + "</center>"
        nama = val[1]
        upload = (
            '<center><input    type="file"    class="form-control"
accept="application/pdf" id="uppdf" data-value="'
            + str(val[0])
            + "' name="pdf" style="width:250px"></center>'
        )
    )

```



```

with connection.cursor() as query:
    query.execute(
        "SELECT          o.kode_periode,d.file          FROM
app_detailpengajuan d LEFT JOIN app_pengajuan p ON
d.kode_pengajuan=p.kode_pengajuan LEFT JOIN app_periode o
ON o.kode_periode=p.periode WHERE p.nidn="
        + nidn
        + "' AND p.kode_pengajuan="
        + str(kode)
        + "' AND d.berkas="
        + str(val[0])
        + ""
    )
    if query.rowcount > 0:
        row = query.fetchone()
        status = (
            "<center><a href="
            + MEDIA_URL
            + "berkas/"
            + nidn
            + "/"
            + row[0]
            + "/"
            + str(row[1])
            + "' target='_blank' class='badge bg-primary'>View
Berkas</a></center>"
        )
    else:
        status = "<center class='badge bg-
danger'>Belum</center>"
        button = (
            '<center><a href="javascript:hBerkas('

```

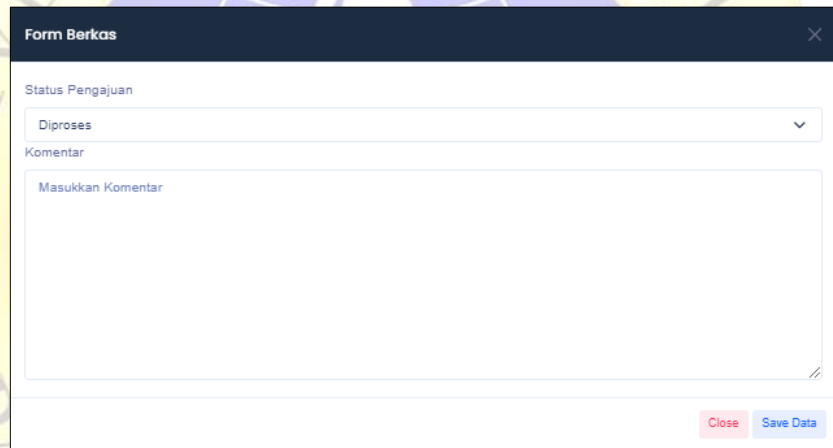
```

+ str(val[0])
+ ")" title = "Delete" class="btn btn-danger btn-icon-circle
btn-icon-circle-sm"><i class="far fa-window-close"></i>'
)
button += "</a></center>"

out = [nom, nama, upload, status]
data.append(out)
no += 1
output = json.dumps({"data": data})
return HttpResponse(output, content_type="application/json")

```

## 21. Status Pengajuan



Gambar 4.42 Tampilan Status Pengajuan

### Segmen Program 4.20 Tampilan Status Pengajuan

```
@csrf_exempt
```

```

def vstatus(request):
    id = request.POST.get("id")
    kat = Pengajuan.objects.filter(id=id).all()
    data = []
    value = {}
    for val in kat.iterator():
        value = {"status": val.status}
        data.append(value)
    output = json.dumps({"data": data})

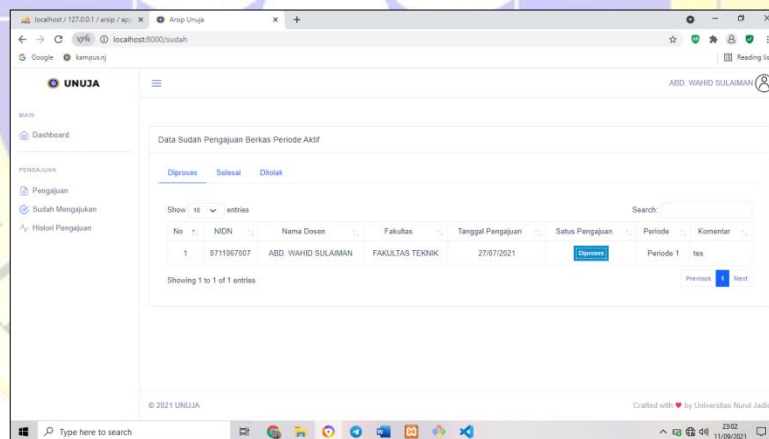
```

```

return HttpResponse(output, content_type="application/json")
@csrf_exempt
def upstatus(request):
    if request.method == "POST":
        pk = request.POST.get("id")
        status = request.POST.get("status")
        komentar = request.POST.get("komentar")
        data = Pengajuan.objects.get(pk=pk)
        data.status = status
        data.komentar = komentar
        data.save()
        arr = 1
    else:
        arr = 0
    return HttpResponse(arr)

```

## 22. Sudah mengajukan (dosen)



Gambar 4.43 Tampilan Sudah Mengajukan

## Segemen Program 4.21 Tampilan Sudah Mengajukan

```

def sudah(request):
    if request.session["nidn"] == None:
        return render(request, "index.html")
    else:
        return render(request, "sudah.html")
@csrf_exempt

```

```

def vsudah(request):
    status = request.POST.get("status")
    po = Periode.objects.filter(status="1").all()
    for p in po.iterator():
        kodepo = p.kode_periode
    query = (
        Pengajuan.objects.filter(status=status, periode=kodepo)
        .all()
        .order_by("-tgl_pengajuan")
    )
    data = []
    out = []
    no = 1
    pengajuan = ""
    for val in query.iterator():
        nom = "<center>" + str(no) + "</center>"
        dosen = Dosen.objects.filter(nidn=val.nidn).all()
        for ds in dosen.iterator():
            nama = "<center>" + ds.nama_dosen + "</center>"
            fakul = Fakultas.objects.filter(id=ds.fakultas).all()
            for fl in fakul.iterator():
                fakultas = "<center>" + fl.nama_fakultas + "</center>"
            tgl = "<center>" +
            val.tgl_pengajuan.strftime("%d/%m/%Y") + "</center>"
            if val.status == "0":
                pengajuan = (
                    '<center><button type="button" class="btn btn-primary
                    btn-square btn-outline-dashed btn-
                    sm">Pengajuan</button><center>'
                )
            elif val.status == "1":
                pengajuan = (

```

```

        '<center><button type="button" class="btn btn-info btn-
square btn-outline-dashed btn-sm">Diproses</button><center>'
    )
    elif val.status == "2":
        pengajuan = (
            '<center><button type="button" class="btn btn-success
btn-square btn-outline-dashed btn-sm">Selesai</button><center>'
        )
    elif val.status == "3":
        pengajuan = (
            '<center><button type="button" class="btn btn-danger
btn-square btn-outline-dashed btn-
sm">Ditolak</button><center>'
        )
    period = Periode.objects.filter(kode_periode=val.periode)
    for pr in period.iterator():
        periode = "<center>" + pr.nama_periode + "</center>"
        kodeperiode = pr.kode_periode
        if val.komentar == None:
            komentar = '<center><komentar>'
        else:
            komentar = val.komentar
    out = [
        nom,
        "<center>" + val.nidn + "</center>",
        nama,
        fakultas,
        tgl,
        pengajuan,
        periode,
        komentar
    ]
]

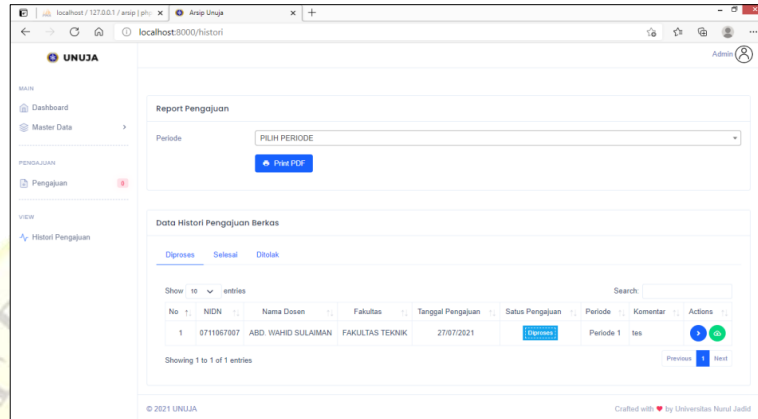
```

```

data.append(out)
no += 1
output = json.dumps({"data": data})
return HttpResponse(output, content_type="application/json")

```

### 23. Histori Pengajuan (admin)



Gambar 4.43 Tampilan Histori pengajuan (admin)

#### Segmen Program 4.22 Tampilan Histori Pengajuan (admin)

```

def histori(request):
    if request.session["nidn"] == None:
        return render(request, "index.html")
    else:
        return render(request, "histori.html")
@csrf_exempt
def vhistori(request):
    status = request.POST.get("status")
    nidn = request.session["nidn"]
    if request.session["status"] == "A":
        query = Pengajuan.objects.filter(
            status=status).all().order_by("-tgl_pengajuan")
    else:
        query = Pengajuan.objects.filter(
            status=status, nidn=nidn).all().order_by("-tgl_pengajuan")
    data = []
    out = []

```



```

no = 1
pengajuan = ""
for val in query.iterator():
    nom = "<center>" + str(no) + "</center>"
    dosen = Dosen.objects.filter(nidn=val.nidn).all()
    for ds in dosen.iterator():
        nama = "<center>" + ds.nama_dosen + "</center>"
        fakul = Fakultas.objects.filter(id=ds.fakultas).all()
        for fl in fakul.iterator():
            fakultas = "<center>" + fl.nama_fakultas + "</center>"
            tgl = "<center>" + val.tgl_pengajuan.strftime("%d/%m/%Y") +
"</center>"
            if val.status == "0":
                pengajuan = '<center><button type="button" class="btn btn-
primary          btn-square          btn-outline-dashed          btn-
sm">Pengajuan</button><center>'
            elif val.status == "1":
                pengajuan = '<center><button type="button" class="btn btn-
info          btn-square          btn-outline-dashed          btn-
sm">Diproses</button><center>'
            elif val.status == "2":
                pengajuan = '<center><button type="button" class="btn btn-
success          btn-square          btn-outline-dashed          btn-
sm">Selesai</button><center>'
            elif val.status == "3":
                pengajuan = '<center><button type="button" class="btn btn-
danger          btn-square          btn-outline-dashed          btn-
sm">Ditolak</button><center>'
            period = Periode.objects.filter(kode_periode=val.periode)
            kodeperiode = ""
            for pr in period.iterator():
                periode = "<center>" + pr.nama_periode + "</center>"

```

```

        kodeperiode = pr.kode_periode
    if val.komentar == None:
        komentar = '<center>-<komentar>'
    else:
        komentar = val.komentar
    button = (
        "<center>"
        + '<a href="javascript:vBerkas('
        + str(val.id)
        + ')" title = "Detail" class="btn btn-primary btn-icon-circle
        btn-icon-circle-sm"><i class="fas fa-angle-right"></i></a>'
        + '&nbsp;<button id="download" data-nidn="'
        + val.nidn
        + '" data-periode="'
        + kodeperiode
        + '" title = "Download File" class="btn btn-success btn-icon-
        circle btn-icon-circle-sm"><i class="mdi mdi-cloud-download-
        outline"></i></button>'
        )
    button += "</a></center>"
    out = [
        nom,
        "<center>" + val.nidn + "</center>",
        nama,
        fakultas,
        tgl,
        pengajuan,
        periode,
        komentar,
        button,
    ]
    data.append(out)

```

```

        no += 1
    output = json.dumps({"data": data})
    return HttpResponse(output, content_type="application/json")

@csrf_exempt
def showhistori(request):
    kode = request.POST.get("kode")
    nidn = request.POST.get("nidn")
    with connection.cursor() as con:
        con.execute("SELECT * FROM app_berkas")
        data = []
        out = []
        no = 1
        for val in con.fetchall():
            nom = "<center>" + str(no) + "</center>"
            nama = val[1]
            with connection.cursor() as query:
                query.execute(
                    "SELECT o.kode_periode,d.file FROM app_detailpengajuan d
                    LEFT JOIN app_pengajuan p ON
                    d.kode_pengajuan=p.kode_pengajuan LEFT JOIN app_periode o
                    ON o.kode_periode=p.periode WHERE p.nidn="
                    + nidn
                    + " AND p.kode_pengajuan="
                    + str(kode)
                    + " AND d.berkas="
                    + str(val[0])
                    + ""
                )
            if query.rowcount > 0:
                row = query.fetchone()
                status = (

```

```

        "<center><a href=\""
        + MEDIA_URL
        + "berkas/"
        + nidn
        + "/"
        + row[0]
        + "/"
        + str(row[1])
        + "' target='_blank' class='badge bg-primary'>View
Berkas</a></center>"
    )
    else:
        status = "<center class='badge bg-
danger'>Belum</center>"
        button = (
            '<center><a href="javascript:hBerkas('
            + str(val[0])
            + ')" title = "Delete" class="btn btn-danger btn-icon-circle
btn-icon-circle-sm"><i class="far fa-window-close"></i>'
        )
        button += "</a></center>"
        out = [nom, nama, status]
        data.append(out)
        no += 1
        output = json.dumps({"data": data})
        return HttpResponse(output, content_type="application/json")
def download(request):
    """

```

A django view to zip files in directory and send it as downloadable response to the browser.

Args:

@request: Django request object

@file\_name: Name of the directory to be zipped

Returns:

A downloadable Http response

"""

```
nidn = request.GET["nidn"]
```

```
periode = request.GET["periode"]
```

```
file_name = nidn + "_" + periode
```

```
files_path = MEDIA_URL + "berkas/" + nidn + "/" + periode +
```

```
"/"
```

```
path_to_zip = make_archive(files_path, "zip", files_path)
```

```
response = HttpResponse(
```

```
    FileWrapper(open(path_to_zip, "rb")),
```

```
    content_type="application/zip"
```

```
)
```

```
response["Content-Disposition"] = 'attachment;
```

```
filename="{filename}.zip".format(
```

```
    filename=file_name.replace(" ", "_")
```

```
)
```

```
return response
```

```
def pdf(request):
```

```
    periode = request.GET["periode"]
```

```
    template = "pdf_template.html"
```

```
    with connection.cursor() as con:
```

```
        con.execute(
```

```
            "SELECT
```

```
            p.nidn,d.nama_dosen,f.nama_fakultas,p.status,p.tgl_pengajuan
```

```
            FROM app_pengajuan p left JOIN app_dosen d ON p.nidn=d.nidn
```

```
            LEFT JOIN app_fakultas f ON f.id=d.fakultas WHERE
```

```
            p.periode=%s", [
```

```
                periode]
```

```
        )
```

```
        data = []
```



```

out = {}
for val in con.fetchall():
    nidn = val[0]
    nama = val[1]
    fakultas = val[2]
    if val[3] == "1":
        status = "Diproses"
    elif val[3] == "2":
        status = "Selesai"
    else:
        status = "Ditolak"
    tgl = val[4].strftime("%d/%m/%Y")
    out = {
        "nidn": nidn,
        "nama": nama,
        "fakultas": fakultas,
        "status": status,
        "tgl": tgl,
    }
    data.append(out)
context = {"data": data}
pdf = render_to_pdf(template, context)
if pdf:
    response = HttpResponse(pdf, content_type="application/pdf")
    filename = "Invoice_%s.pdf" # % inflow.student
    content = "inline; filename='%s'" % filename
    download = request.GET.get("download")
    if download:
        content = "attachment; filename='%s'" % filename
    response["Content-Disposition"] = content
    return response
return HttpResponse("Not found")

```



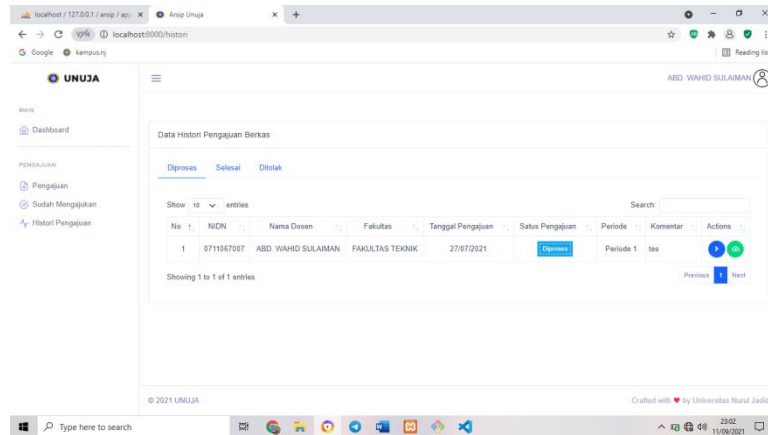
```

def render_to_pdf(template_src, context_dict={}):
    template = get_template(template_src)
    html = template.render(context_dict)
    result = BytesIO()
    pdf = pisa.pisaDocument(BytesIO(html.encode("ISO-8859-1")),
result)
    if not pdf.err:
        return HttpResponse(result.getvalue(),
content_type="application/pdf")
    return None

@csrf_exempt
def getperiode1(request):
    sql = Periode.objects.all()
    data = "<option value="">PILIH PERIODE</option>"
    for val in sql.iterator():
        data += (
            '<option value="'
            + str(val.kode_periode)
            + ">'
            + val.nama_periode.upper()
            + " ("
            + val.tgl_mulai.strftime("%Y")
            + ")"
            + "</option>"
        )
    return HttpResponse(data)

```

#### 24. Histori Pengajuan (dosen)



Gambar 4.44 Tampilan Histori pengajuan (dosen)  
**Segmen Program 4.23** Tampilan Histori Pengajuan (Dosen)

def histori(request):

    if request.session["nidn"] == None:

        return render(request, "index.html")

    else:

        return render(request, "histori.html")

@csrf\_exempt

def vhistori(request):

    status = request.POST.get("status")

    nidn = request.session["nidn"]

    if request.session["status"] == "A":

        query = Pengajuan.objects.filter(  
         status=status).all().order\_by("-tgl\_pengajuan")

    else:

        query = Pengajuan.objects.filter(  
         status=status, nidn=nidn).all().order\_by("-tgl\_pengajuan")

    data = []

    out = []

    no = 1

    pengajuan = ""

    for val in query.iterator():

        nom = "<center>" + str(no) + "</center>"

        dosen = Dosen.objects.filter(nidn=val.nidn).all()

```

for ds in dosen.iterator():
    nama = "<center>" + ds.nama_dosen + "</center>"
    fakul = Fakultas.objects.filter(id=ds.fakultas).all()
    for fl in fakul.iterator():
        fakultas = "<center>" + fl.nama_fakultas + "</center>"
    tgl = "<center>" + val.tgl_pengajuan.strftime("%d/%m/%Y") +
"</center>"
    if val.status == "0":
        pengajuan = '<center><button type="button" class="btn btn-
primary btn-square btn-outline-dashed btn-
sm">Pengajuan</button><center>'
    elif val.status == "1":
        pengajuan = '<center><button type="button" class="btn btn-
info btn-square btn-outline-dashed btn-
sm">Diproses</button><center>'
    elif val.status == "2":
        pengajuan = '<center><button type="button" class="btn btn-
success btn-square btn-outline-dashed btn-
sm">Selesai</button><center>'
    elif val.status == "3":
        pengajuan = '<center><button type="button" class="btn btn-
danger btn-square btn-outline-dashed btn-
sm">Ditolak</button><center>'
    period = Periode.objects.filter(kode_periode=val.periode)
    kodeperiode = ""
    for pr in period.iterator():
        periode = "<center>" + pr.nama_periode + "</center>"
        kodeperiode = pr.kode_periode
    if val.komentar == None:
        komentar = '<center>-<komentar>'
    else:
        komentar = val.komentar

```

```

button = (
    "<center>"
    + '<a href="javascript:vBerkas('
    + str(val.id)
    + ')" title = "Detail" class="btn btn-primary btn-icon-circle
btn-icon-circle-sm"><i class="fas fa-angle-right"></i></a>'
    + '&nbsp;<button id="download" data-nidn="'
    + val.nidn
    + '" data-periode="'
    + kodeperiode
    + '" title = "Download File" class="btn btn-success btn-icon-
circle btn-icon-circle-sm"><i class="mdi mdi-cloud-download-
outline"></i></button>'
)
button += "</a></center>"
out = [
    nom,
    "<center>" + val.nidn + "</center>",
    nama,
    fakultas,
    tgl,
    pengajuan,
    periode,
    komentar,
    button,
]
data.append(out)
no += 1
output = json.dumps({"data": data})
return HttpResponse(output, content_type="application/json")
@csrf_exempt
def showhistori(request):

```

```

kode = request.POST.get("kode")
nidn = request.POST.get("nidn")
with connection.cursor() as con:
    con.execute("SELECT * FROM app_berkas")
    data = []
    out = []
    no = 1
    for val in con.fetchall():
        nom = "<center>" + str(no) + "</center>"
        nama = val[1]
        with connection.cursor() as query:
            query.execute(
                "SELECT o.kode_periode,d.file FROM app_detailpengajuan d
                LEFT JOIN app_pengajuan p ON
                d.kode_pengajuan=p.kode_pengajuan LEFT JOIN app_periode o
                ON o.kode_periode=p.periode WHERE p.nidn="
                + nidn
                + "' AND p.kode_pengajuan="
                + str(kode)
                + "' AND d.berkas="
                + str(val[0])
                + "'"
            )
        if query.rowcount > 0:
            row = query.fetchone()
            status = (
                "<center><a href="
                + MEDIA_URL
                + "berkas/"
                + nidn
                + "/"

```



```

        + row[0]
        + "/"
        + str(row[1])
        + "' target='_blank' class='badge bg-primary'>View
Berkas</a></center>"
    )
    else:
        status = "<center class='badge bg-
danger'>Belum</center>"
        button = (
            '<center><a href="javascript:hBerkas('
            + str(val[0])
            + ")" title = "Delete" class="btn btn-danger btn-icon-circle
btn-icon-circle-sm"><i class="far fa-window-close"></i>'
        )
        button += "</a></center>"
        out = [nom, nama, status]
        data.append(out)
        no += 1
        output = json.dumps({"data": data})
        return HttpResponse(output, content_type="application/json")
def download(request):
    """
    A django view to zip files in directory and send it as
    downloadable response to the browser.
    Args:
        @request: Django request object
        @file_name: Name of the directory to be zipped
    Returns:
        A downloadable Http response
    """
    nidn = request.GET["nidn"]

```



```

periode = request.GET["periode"]
file_name = nidn + "_" + periode
files_path = MEDIA_URL + "berkas/" + nidn + "/" + periode +
"/"

path_to_zip = make_archive(files_path, "zip", files_path)
response = HttpResponse(
    FileWrapper(open(path_to_zip, "rb")),
    content_type="application/zip"
)
response["Content-Disposition"] = 'attachment;
filename="{filename}.zip".format(
    filename=file_name.replace(" ", "_")
)
return response

```

#### 4.4 Testing

Testing atau uji coba yang akan dilakukan ialah uji coba internal dan eksternal, uji coba internal menggunakan black box testing sedangkan uji coba eksternal ialah dengan melakukan pengujian langsung kepada user, yaitu dengan memberi angket kepada penguji dengan beberapa pertanyaan yang terdapat didalamnya dengan melihat kondisi sistem yang dibuat untuk melihat keberhasilan sistem dengan melihat setiap fitur yang terdapat didalam sistem apakah berjalan dengan baik atau tidak dan hasil dari angket tersebut yang akan ditetapkan sebagai nilai yang nantinya akan dihitung dengan menggunakan perhitungan skala likert untuk mengetahui hasil akhir dari proses nilai yang dihasilkan dari angket.

Berikut hasil dari uji coba internal dan eksternal :

#### 4.4.1 Pengujian Internal

Tabel 4.9 Tabel Pengujian Internal

No	Pengujian	Event	Hasil yang diharapkan	Hasil	
				Ya	Tidak
1	Form Login	Username dan Password benar	Menampilkan halaman utama (Dashboard)	✓	
		Username dan Password salah	Menampilkan notifikasi kesalahan dan tidak bisa menampilkan dashboard	✓	
2	Halaman Utama (Home)	Total data Dosen	Menampilkan jumlah data dalam tabel data dosen	✓	
		Total data Admin	Menampilkan jumlah data dalam tabel data admin	✓	
		Total Berkas	Menampilkan jumlah data kategori berkas dalam tabel berkas	✓	
		Total Pengajuan	Menampilkan jumlah data dalam tabel	✓	

Tabel 4.10 Tabel Pengujian Internal (Lanjutan)

		Garfik (Statistik Pengajuan )	Menampilkan prosentase dari pengajuan data inpassing	✓	
3	Master Data	Data Dosen	Menampilkan semua data dosen yang telah diinputkan	✓	
			Tambah Data: menampilkan form tambah data baru dan ditampilkan ditabel	✓	
			Edit Data: Menampilkan form edit data	✓	
			Delete Data: Menghapus data yang sudah tidak diperlukan	✓	
		Data Admin	Menampilkan semua data admin yang telah diinputkan	✓	
			Tambah Data: menampilkan form tambah data baru	✓	

Tabel 4.11 Tabel Pengujian Internal (Lanjutan)

			Edit Data: Menampilkan form edit data admin	✓	
			Delete Data: Menghapus data tidak diperlukan	✓	
		Kategori Berkas	Menampilkan semua kategori berkas pengajuan yang telah diinputkan	✓	
			Tambah Data: menampilkan form tambah data baru dan ditampilkan ditabel	✓	
			Edit Data: Menampilkan form edit berkas	✓	
			Delete Data: Menghapus data yang sudah tidak diperlukan	✓	
		Periode	Menampilkan periode pengajuan atau tanggal pengajuan	✓	

Tabel 4.13 Tabel Pengujian Internal (Lanjutan)

			Tambah Data: menampilkan form tambah data baru dan ditampilkan ditabel	✓	
			Edit Data: Menampilkan form edit periode	✓	
			Delete Data: Menghapus data	✓	
4	Pengolahan Data	Pengajuan	Menampilkan semua data pengajuan inpassing	✓	
			Tambah Data: menampilkan form tambah data baru dan ditampilkan ditabel	✓	
			Detail: Menampilkan detail data pengajuan atau berkas persyaratan yang telah diupload	✓	

Tabel 4.13 Tabel Pengujian Internal (Lanjutan)

			Status: Menampilkan form status pengajuan dan komentar	✓	
			Download File: Mengunduh file berkas persyaratan setiap dosen dalam bentuk pdf	✓	
5	View	Histori Pengajuan	Menampilkan data histori yang diinputkan di data pengajuan	✓	
			Detail: Menampilkan detail data pengajuan atau berkas persyaratan yang telah diupload dan diinputkan ditabel pengajuan	✓	
			Download File: Mengunduh file berkas	✓	



			persyaratan setiap dosen dalam bentuk pdf		
--	--	--	--	--	--

Tabel 4.14 Tabel Pengujian Internal (Lanjutan)

			Cetak PDF: Mencetak data histori menjadi PDF dan didownload	✓	
6	Profil		Menampilkan profil admin yang telah diinputkan ditabel admin	✓	
7	Logout		Keluar dari akun dan kembali ke halaman login	✓	

#### 4.4.2 Pengujian Eksternal

Tabel 4.15 Tabel Pengujian Eksternal

No	Pertanyaan	Keterangan				
		SB	B	C	TB	STB
1	Apakah sistem ini mudah dioperasikan?		3			
2	Apakah sistem ini membantu mempermudah petugas dalam menyimpan dan mengarsipkan data inpassing dosen?	3				

Tabel 4.16 Tabel Pengujian Eksternal (Lanjutan)

3	Apakah semua fitur dalam sistem ini sudah sesuai dengan yang diinginkan?		3			
4	Apakah semua fitur dalam sistem ini sudah berjalan dengan baik?		3			
5	Apakah Sistem ini sudah layak digunakan?		3			

Tabel 4.18 Tabel Perhitungan Skala Likert

No	Skala Jawaban	Skor
1	Sangat Baik (SB)	$5 \times 3 = 15$
2	Baik (B)	$4 \times 12 = 48$
3	Cukup (C)	$3 \times 0 = 0$
4	Tidak Baik (TB)	$2 \times 0 = 0$
5	Sangat Tidak Baik (STB)	$1 \times 0 = 0$
		Total = 63

Skor tertinggi (Y) = skor tertinggi x jumlah responden

$$Y = 5 \times 3 = 15$$

Skor terendah (X) = skor terendah x jumlah responden

$$X = 1 \times 3 = 3$$

Jumlah skor ideal = skor tertinggi x jumlah soal x jumlah responden  
 $5 \times 5 \times 3 = 75$

Berikut interval terendah (0%) hingga tertinggi (100%)

Tabel 4.19 Tabel Kriteria Interpretasi Persen

No	Presentase	Keterangan
1	0 – 19,99%	Sangat Tidak Baik
2	20 – 39,99%	Tidak Baik
3	40 – 59,99%	Cukup

4	60 – 79,99%	Baik
5	80 – 100%	Sangat Baik

Rumus penentuan nilai presentase dari kuisisioner yang dilakukan adalah penentuan nilai presentase = (Total skor yang diperoleh/skor ideal) x 100 sehingga didapat nilai presentase =  $(63/75) \times 100 = 84\%$ . Berdasarkan tabel presentase nilai diatas dapat diambil kesimpulan dari aplikasi pengarsipan data inpassing dosen Universitas Nurul Jadid berbasis web framework django dengan kategori sangat baik.

#### 4.5 Maintenance

Pada tahap terakhir yaitu maintenance atau perawatan perangkat lunak aplikasi pengarsipan data inpassing dosen dilakukan dengan cara mengupdate fitur dalam sistem jika ada perubahan fitur yang dibutuhkan dan dapat mengupdate username dan password admin jika diperlukan dan memperbaiki sistem jika suatu hari terjadi kesalahan pada sistem.