

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Metode Pengumpulan Data

Sebelum dilakukan uji coba, pengumpulan data adalah langkah awal yang harus dilakukan, yaitu dengan cara observasi dan wawancara kepada pihak terkait agar ditemukannya beberapa kebutuhan yang diinginkan. Berdasarkan hasil wawancara dengan petugas pengajuan jabatan fungsional dosen ialah sebagai berikut :

##### 4.1.1 Hasil Observasi dan Wawancara

Hasil dari observasi dan wawancara dengan pihak terkait ialah sebagai berikut:

##### 1. Observasi

Tabel 4.1 Habel Observasi

No	Uraian Kegiatan	Hasil Observasi
1	Mengamati proses pengajuan jabatan fungsional dosen	Masih dilakukan secara manual yaitu dosen menyerahkan berkas pengajuan jabatan fungsional dosen menggunakan <i>flashdisk</i> , semua berkas disimpan di dalam folder.
2	Mengamati petugas dalam proses pengecekan kelengkapan dokumen persyaratan yang diajukan oleh dosen	Pengecekan di lakukan dengan membuka berkas setiap yang diajukan dosen, jika terjadi kesalahan atau berkas tidak sesuai dosen mengajukan kembali berkas yang sesuai dengan persyaratan yang ditentukan.

2. Wawancara

Tabel 4.2 Hasil Wawancara

No	Bagian	Pertanyaan	Hasil Wawancara
1	31 Maret 2021 Dosen Anis Yusrotun N,S.Kom, M.MT (ex Tim PAK)	Bagaimana proses pengajuan jabatan fungsional dosen?	Proses pengajuan jabatan fungsional dosen masih dilakukan dengan manual yaitu dosen menyerahkan berkas persyaratan dengan menggunakan <i>flashdisk</i> yang diserahkan ke petugas lalu akan disalin ke komputer dan di simpan didalam folder.
		Bagaimana proses pengecekan kelengkapan dokumen persyaratan pengajuan jabatan fungsional dosen?	Pengecekan dilakukan dengan memeriksa dan membuka berkas persyaratan yang diajukan dosen yang di kumpulkan di folder, jika terdapat berkas yang kurang atau tidak sesuai maka dosen akan kembali menyerahkan berkas yang diperlukan ke petugas secara langsung.

Tabel 4.3 Hasil Wawancara (lanjutan)

2	17 Juni 2021 Ni'matur Rohim, S.Kom (Tim PAK)	Setelah pengajuan selesai dilakukan siapa yang memberi informasi ke dosen bahwa pengajuan sudah diterima?	Setelah pengajuan diterima maka petugas akan menyampaikan informasi tersebut lewat telegram yang didalamnya sudah terdapat grup terkait dosen yang mengajukan jabatan fungsional dosen
		Apa yang dilakukan jika ada data yang kurang atau tidak sesuai persyaratan?	Maka petugas akan menyampaikan lewat grup ditelegram dan selanjutnya dosen menyerahkan berkas persyaratan kembali ke petugas secara langsung.
		Apakah dosen boleh mengajukan kembali jabatan yang lebih tinggi?	Bisa tetapi harus mengajukannya harus urut, misalkan mengajukan asisten ahli dulu maka boleh selanjutnya mengajukan jabatan lektor.

### 3. Studi literatur

Teknik pengumpulan data pada tahap ini mengacu pada penelitian sebelumnya, disesuaikan dengan permasalahan yang sedang dilakukan sebagai bahan evaluasi terhadap sistem yang akan dirancang.

## 4.2 Hasil Analisis dan Desain

### 4.2.1 Analisis

Analisis merupakan bagian dari tahap pengumpulan data yang dilakukan untuk menganalisa sebuah system mengenai pengajuan jabatan fungsional di universitas nurul jadid. Analisis ini dilakukan untuk mengetahui bagaimana sistem lama untuk dibandingkan dengan sistem baru yang diinginkan dalam monitoring pengajuan jabatan fungsional dosen.

#### 1. Analisis sistem lama

Proses pengajuan jabatan fungsional di universitas nurul jadid masih dilakukan secara manual yaitu dosen mengumpulkan berkas persyaratan yang dibutuhkan menggunakan *flashdisk* kepada petugas yang selanjutnya akan disalin ke komputer. Petugas melakukan pengecekan berkas dosen dengan membuka berkas-berkas yang dikumpulkan untuk mengetahui kelengkapan persyaratan yang dibutuhkan, dengan sistem manual tersebut dapat mengakibatkan beberapa permasalahan yaitu pengumpulan berkas yang kurang efektif, membutuhkan waktu cukup lama dalam pengecekan kelengkapan berkas persyaratan pengajuan, dan informasi mengenai diterimanya pengajuan akan jadi dampak dari sistem ini.

#### 2. Analisis sistem baru

Proses sistem baru ini yaitu dosen melakukan pengajuan menggunakan aplikasi android dan akan diterima oleh petugas dengan sistem website yang sudah dibuat. Pada sistem ini petugas dapat lebih mudah mengecek kelengkapan berkas persyaratan dan setiap nama dosen yang mengajukan akan ada tombol detail yang didalamnya terdapat berkas yang diajukan dosen tersebut jadi berkas dosen tidak campur dengan berkas dosen lain, dan jika terdapat kesalahan atau ketidaksesuaian persyaratan maka petugas dapat mengirim komentar kepada dosen tentang kesalahan yang

harus diperbaiki. Sistem ini juga dilengkapi dengan adanya data dosen tetap dan juga menampilkan grafik pengajuan.

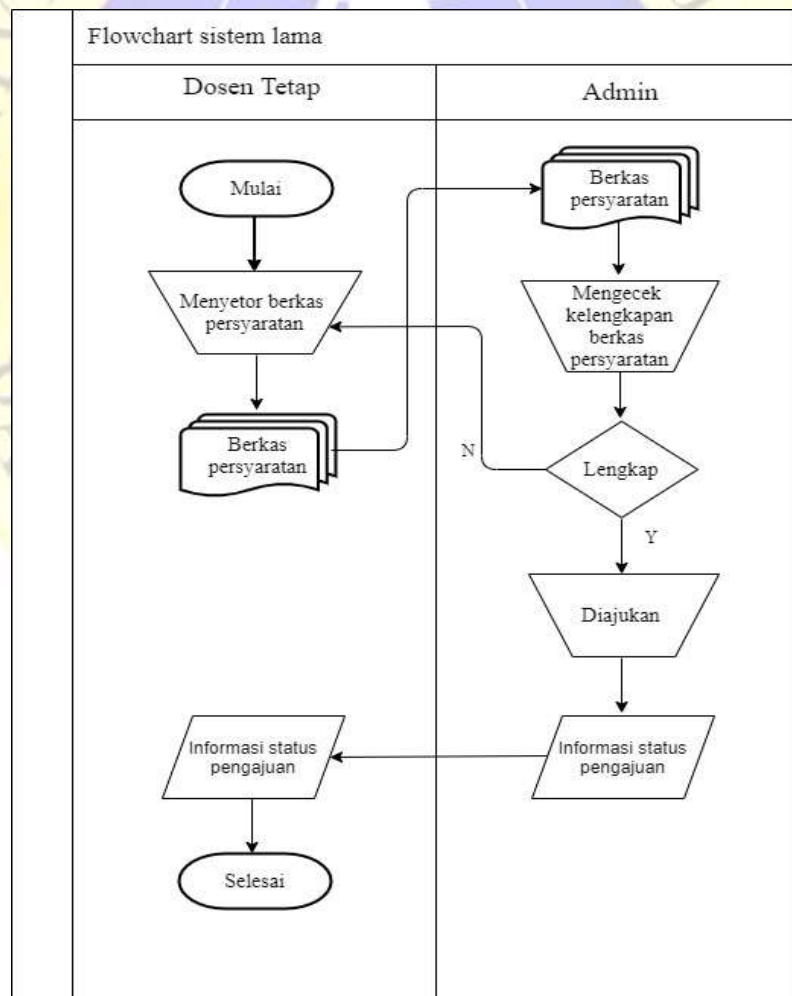
#### 4.2.2 Desain

Desain sistem ialah tahap setelah analisis data. Desain sistem bertujuan untuk memberikan gambaran tentang sistem yang akan dibuat. Pada tahap desain sistem akan menggunakan *Flowchart*, *Data Flow Diagram* dan *Entity Relationship Diagram*, dimana desain sistem ini yang akan digunakan pada tahap selanjutnya.

a. *Flowchart*

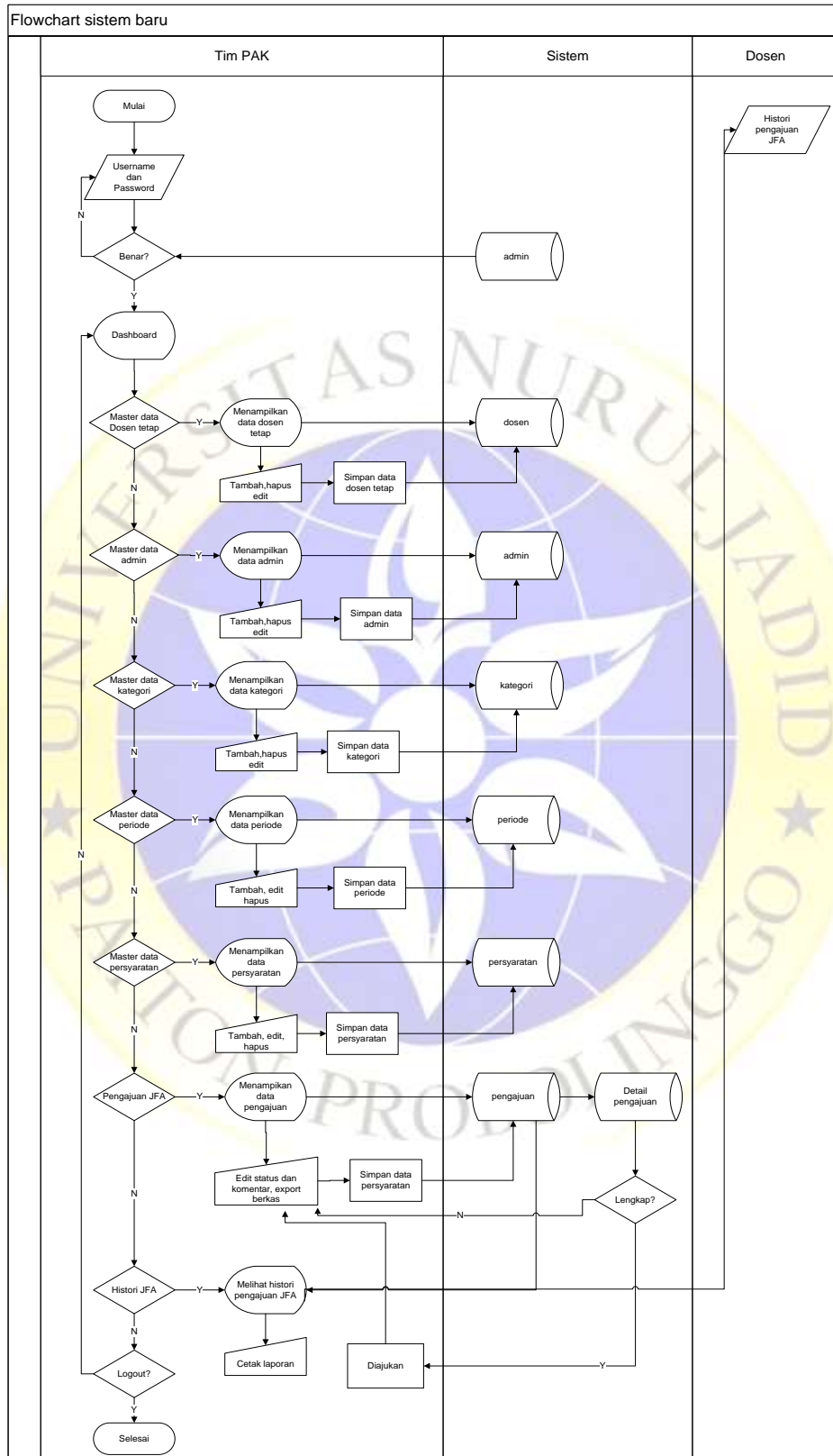
Gambaran *flowchart* monitoring pengajuan jabatan fungsional dosen sistem lama dan sistem baru ialah sebagai berikut:

1. *Flowchart* Sistem Lama



Gambar 4.1 Sistem Lama Pengajuan Jabatan Fungsional Dosen

## 2. Flowchart Sistem Baru



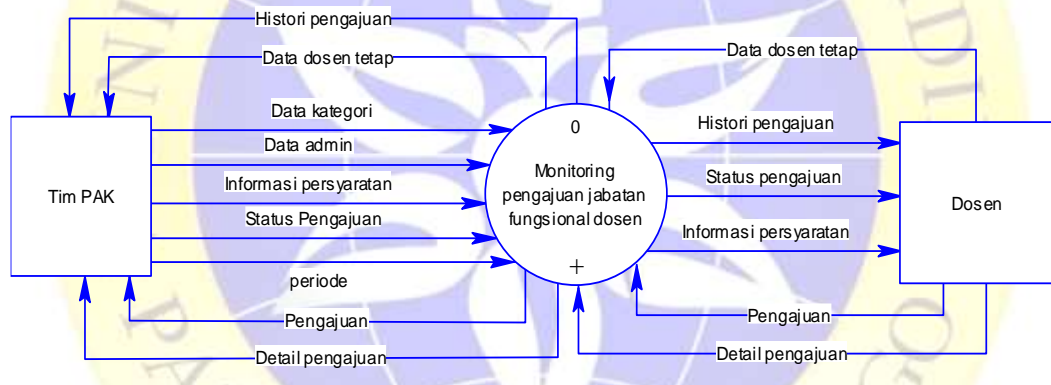
Gambar 4.2 Sistem Baru Monitoring Pengajuan Jabatan Fungsional Dosen

b. *Data Flow Diagram (DFD)*

*Data Flow Diagram (DFD)* digunakan untuk menggambarkan suatu sistem yang akan dikembangkan agar terstruktur dengan jelas, sehingga bisa dengan mudah dipahami secara logika. Berikut merupakan gambaran dari *Data Flow Diagram* sistem baru monitoring pengajuan jabatan fungsional dosen di Universitas Nurul Jadid:

1. *Context Diagram (DFD level 0)*

*Context Diagram* digambarkan untuk mendefinisikan dan memperjelas batas-batas sistem perangkat lunak, mengidentifikasi hubungan dengan entitas entitas diluar sistem itu sendiri, seperti sistem, kelompok organisasi, penyimpanan data eksternal lain. Seluruh sistem perangkat lunak ditampilkan sebagai satu proses.

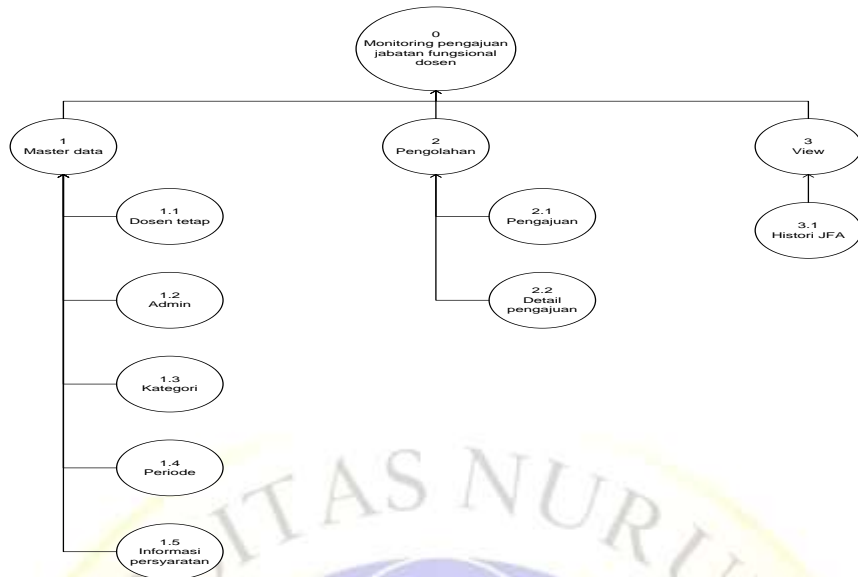


Gambar 4.3 Context Diagram

2. *Dekomposisi Diagram*

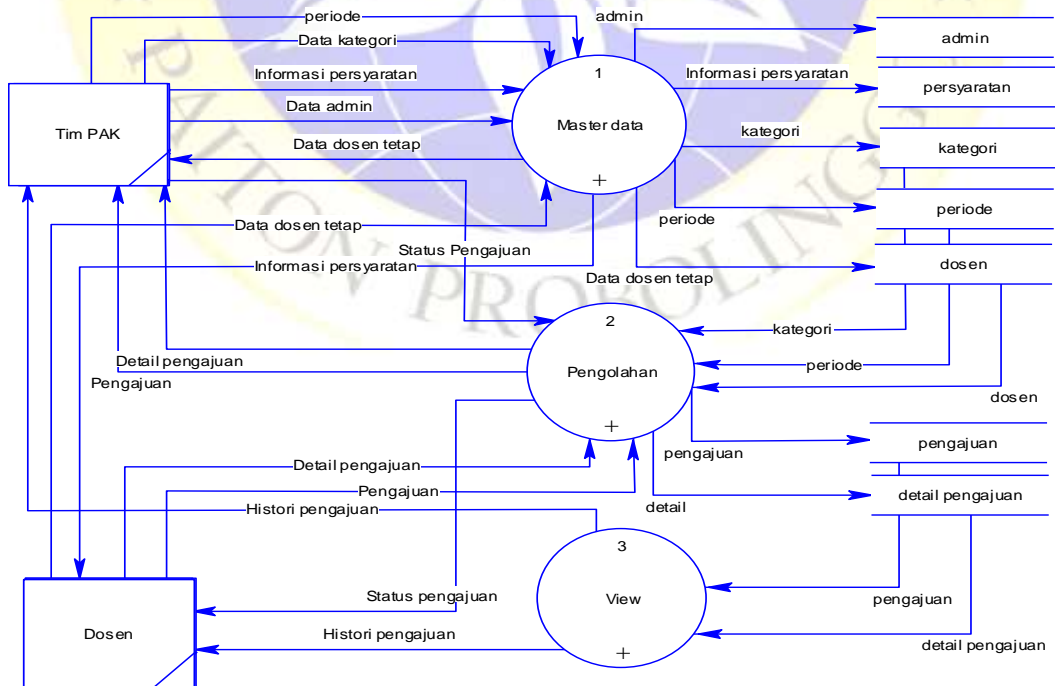
Dekomposisi Diagram adalah kegiatan menguraikan sistem kedalam subsistem, proses dan subproses komponennya. Tiap tingkatan abstraksi menampilkan detail lebih banyak atau lebih sedikit (sesuai keinginan) mengenai keseluruhan sistem atau subset sistem tersebut.

Diagram dekomposisi adalah diagram yang digunakan untuk menggambarkan dekomposisi sebuah sistem disebut juga bagan hierarki menunjukkan dekomposisi fungsional top-down dan struktur sistem. Diagram dekomposisi merupakan alat perencanaan untuk model proses yang lebih detail, yaitu diagram aliran data.



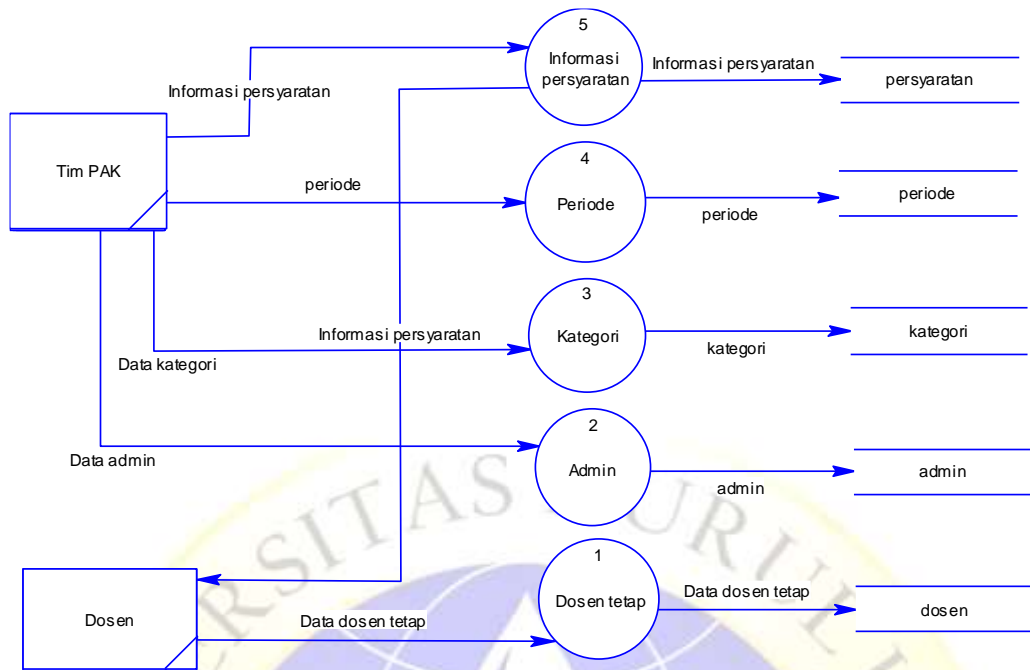
Gambar 4.4 Dekomposisi Diagram

3. *Data Flow Diagram (DFD) Level 1 dan 2* Setelah menggambarkan hubungan eksternal dengan sistem global dan membuat dekomposisi diagram, maka langkah selanjutnya ialah pembuatan DFD Level 1 dan DFD Level 2 sebagai pemecahan dari diagram konteks serta penyimpanan data.

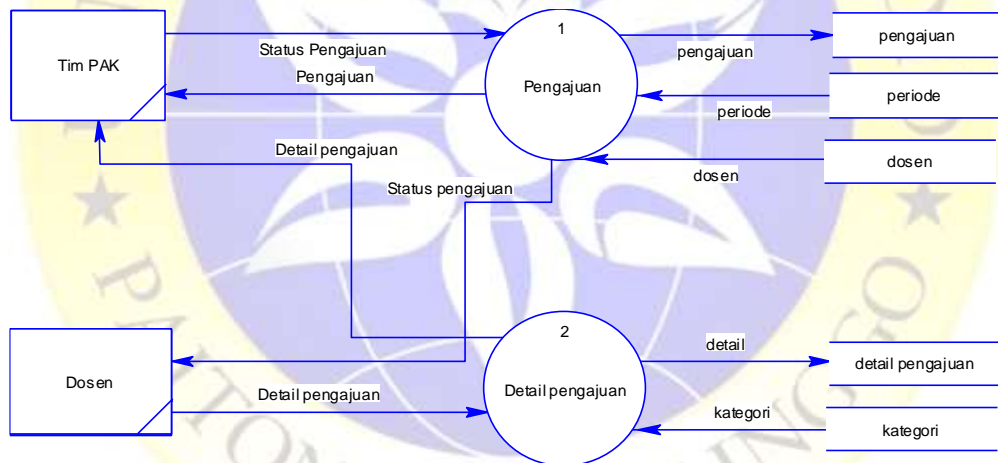


Gambar 4.5 DFD Level 1

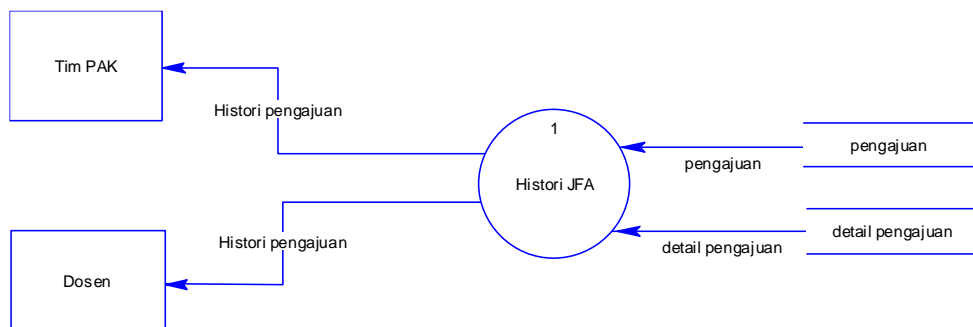




Gambar 4.6 proses 1 master data

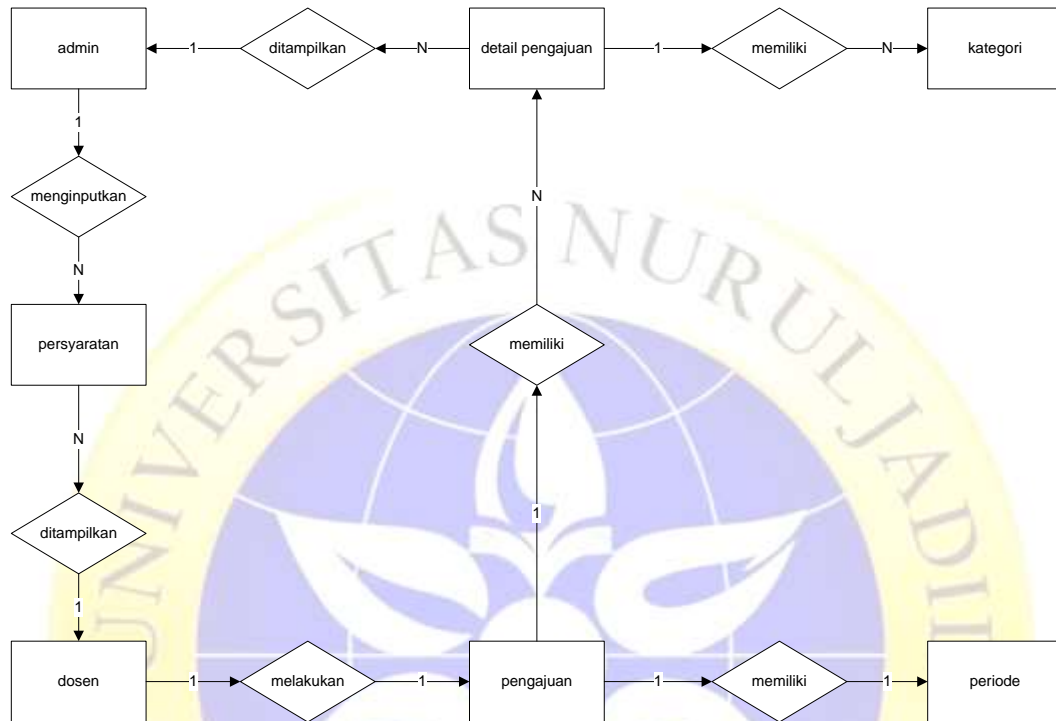


Gambar 4.7 proses 2 pengolahan



Gambar 4.8 proses 3 view

c. *Entity Relationship Diagram* (ERD) merupakan suatu model untuk menjelaskan hubungan antara satu entitas dengan entitas lainnya dalam basis data. Berikut gambaran dari ERD yang diusulkan untuk sistem baru pengajuan jabatan fungsional dosen universitas nurul jadid:



Gambar 4.9 *Entity Relationship Diagram* (ERD)

Kamus Data:

- a. Admin : id, nama, jns\_kelamin, tgl\_lahir, alamat, kontak, foto, username, password
- b. Data dosen tetap : id, nidn, nama\_dosen, jns\_kelamin, tgl\_lahir, alamat, kontak, fakultas, foto, username, password
- c. Pengajuan : id, nidn, prodi, tgl\_pengajuan, status usulan, status berkas, kode\_pengajuan, periode, komentar
- d. Periode : id, nama\_periode, tgl\_mulai, tgl\_selesai, status
- e. Detailpengajuan : id, kode\_pengajuan, kategori, berkas
- f. Kategori : id, nama\_kategori\_kategori
- g. Persyaratan : id, berkas.

d. Struktur Basis Data

Basis data adalah wadah untuk menyimpan data yang dapat dibuka, dikelola juga dicari pada saat dibutuhkan oleh pengguna. Berikut merupakan struktur basis data yang akan terhubung pada sistem informasi yang akan dibuat:

Tabel 4.4 Tabel dosen

No	Nama	Type
1	Id	bigint(20)
2	Nidn	Varchar(50)
3	nama_dosen	Varchar(100)
4	jns_kelamin	Varchar(1)
5	tgl_lahir	Date
6	Alamat	Varchar(100)
7	Kontak	Varchar(100)
8	Fakultas	Int(11)
9	Foto	Varchar(100)
10	Username	Varchar(50)
11	Password	Varchar(100)

Tabel 4.5 Tabel admin

No	Nama	Type
1	Id	bigint(20)
2	Nama	Varchar(100)
3	jns_kelamin	Varchar(1)
4	tgl_lahir	Date
5	Alamat	Varchar(100)
6	Kontak	Varchar(100)
7	Foto	Varchar(100)
8	Username	Varchar(50)
9	Password	Varchar(100)

Tabel 4.6 Tabel pengajuan

No	Nama	Type
1	Id	bigint(20)
2	Nidn	Varchar(100)
3	Prodi	Varchar(50)
4	Tgl_pengajuan	Date
5	Status_usulan	Varchar(100)
6	Status berkas	Varchar(100)
7	Kode_pengajuan	Varchar(25)
8	Periode	Varchar(3)
9	Komentar	Varchar(255)

Tabel 4.7 Tabel periode

No	Nama	Type
1	Id	bigint(20)
2	Nama_periode	Varchar(100)
3	Tgl_mulai	Date
4	Tgl_selesai	Date
5	Status	Int(11)

Tabel 4.8 Tabel detailpengajuan

No	Nama	Type
1	Id	bigint(20)
2	Kode_pengajuan	Varchar(100)
3	Kategori	Varchar(7)
4	Berkas	Varchar(255)

Tabel 4.9 Tabel kategori

No	Nama	Type
1	Id	bigint(20)
2	Nama_kategori	Varchar(100)
3	Kategori	Varchar(25)

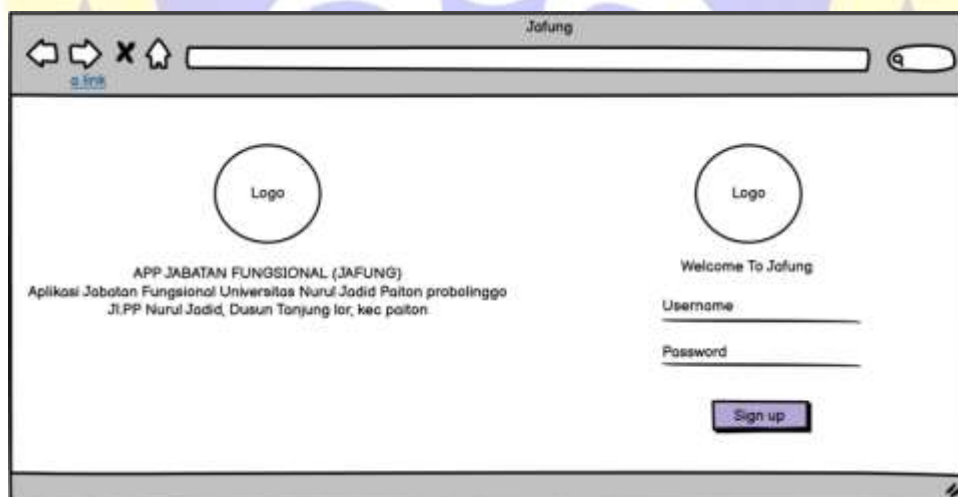
Tabel 4.10 Tabel Persyaratan

No	Nama	Type
1	Id	int(11)
2	Berkas	Varchar(255)

e. Desain interface

a. Desain login

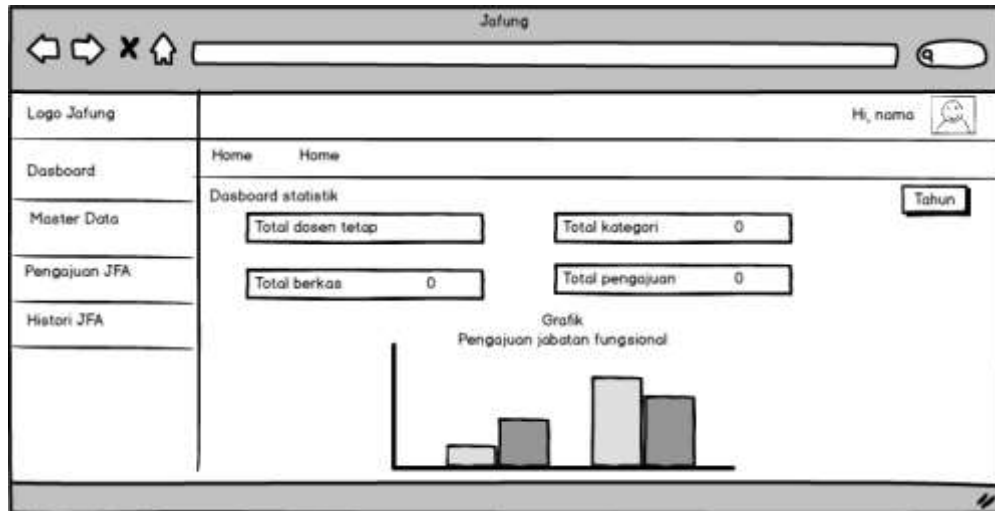
Login ialah menu untuk masuk ke website dengan memasukkan username dan password untuk masuk ke halaman website dimana tampilan desain login dapat dilihat pada gambar 4.10



Gambar 4.10 Desain login

b. Desain halaman dashboard

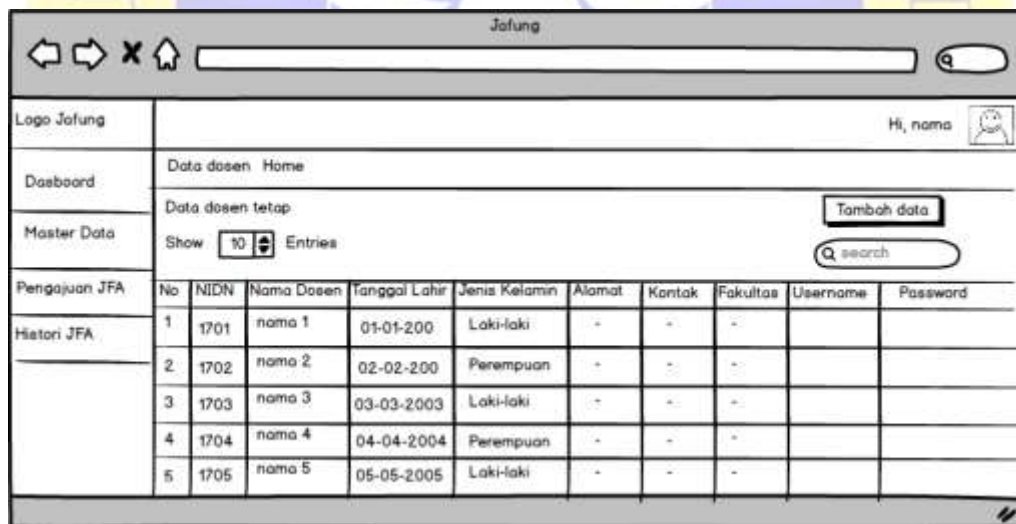
Setelah melakukan login maka tampilan awal yang akan di tampilkan adalah halam dashboard yang terdapa pada gambar 4.11



Gambar 4.11 Desain halaman utama

c. Desain Master data dosen tetap

Pada tombol master data terdapat beberapa menu diantaranya menu dosen tetap yang menampilkan data dosen tetap universitas nurul jadid dapat dilihat pada gambar 4.12



Gambar 4.12 Desain master data dosen tetap

d. Desain Tambah data dosen tetap

Form tambah data digunakan untuk melakukan penambahan data dosen tetap jika terdapat data dosen yang belum tercantum pada halaman data dosen dapat dilihat pada gambar 4.13

Gambar 4.13 Desain tambah data dosen tetap

e. Desain master data kategori

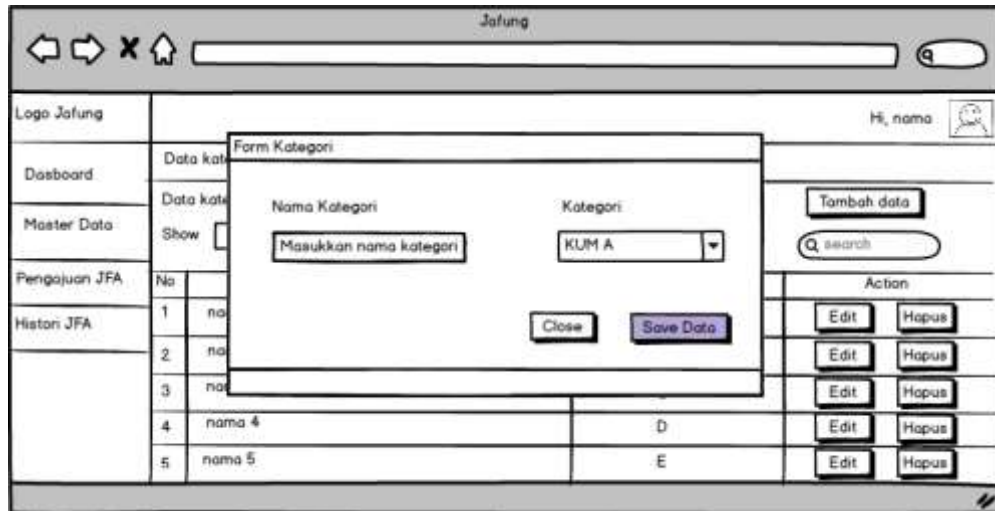
Halaman kategori ialah halaman yang menampilkan beberapa kategori/berkas-berkas persyaratan yang akan diupload oleh dosen pada saat melakukan pengajuan kenaikan jabatan yang dapat dilihat pada gambar 4.14

No	Nama Kategori	Kategori	Action
1	nama 1	A	Edit Hapus
2	nama 2	B	Edit Hapus
3	nama 3	C	Edit Hapus
4	nama 4	D	Edit Hapus
5	nama 5	E	Edit Hapus

Gambar 4.14 Desain master data kategori

f. Desain tambah data kategori

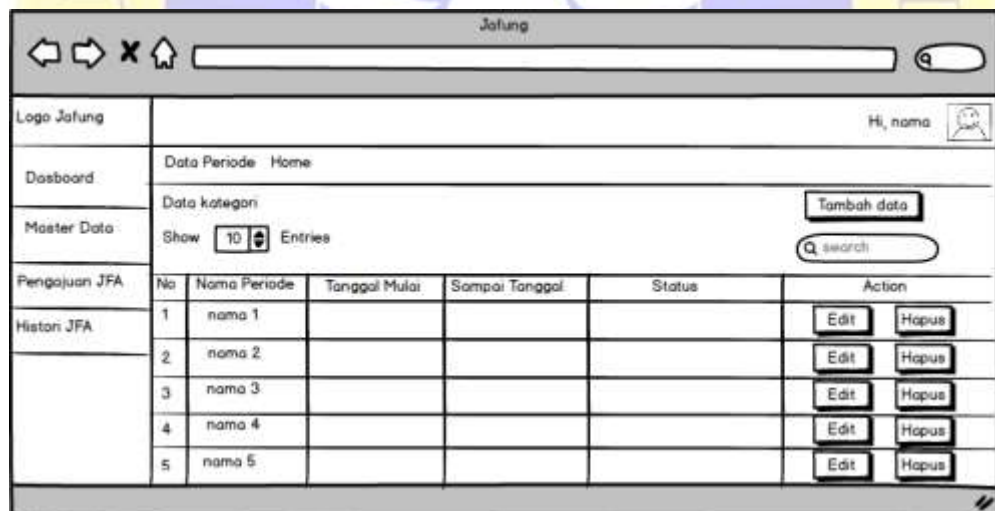
Form tambah kategori berfungsi untuk menbah kategori jika terdapat penambahan berkas persyaratan untuk pengajuan kenaikan jabatan dosen yang dapat dilihat pada gambar 4.14



Gambar 4.15 Desain tambah data kategori

g. Desain master data periode

Halaman periode menampilkan periode yang akan ditentukan dalam pengajuan kenaikan jabatan dosen dan melengkapi persyaratan yang sudah ditentukan yang dapat dilihat pada gambar 4.16

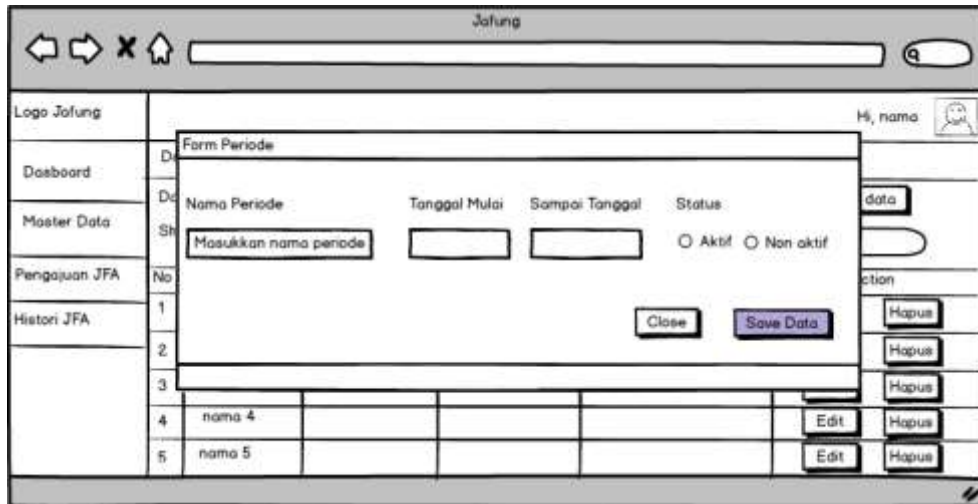


Gambar 4.16 Desain master data periode

h. Desain tambah data periode

Form tambah periode digunakan jika ada penambahan periode untuk pengajuan kenaikan jabatan dosen dengan ditentukan tanggal mulai dan akhir pengajuan yang dapat dilihat pada gambar 4.17

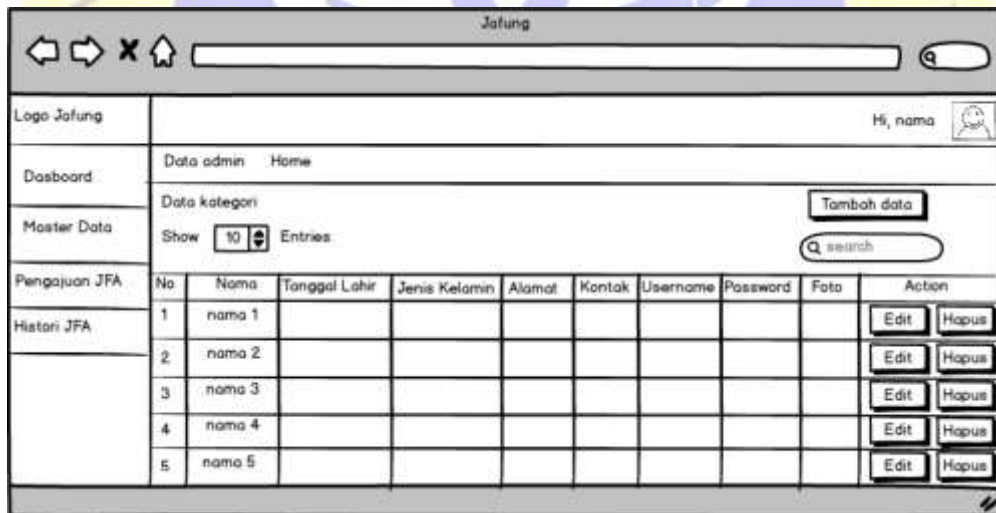




Gambar 4.17 Desain tambah data periode

i. Desain master data admin

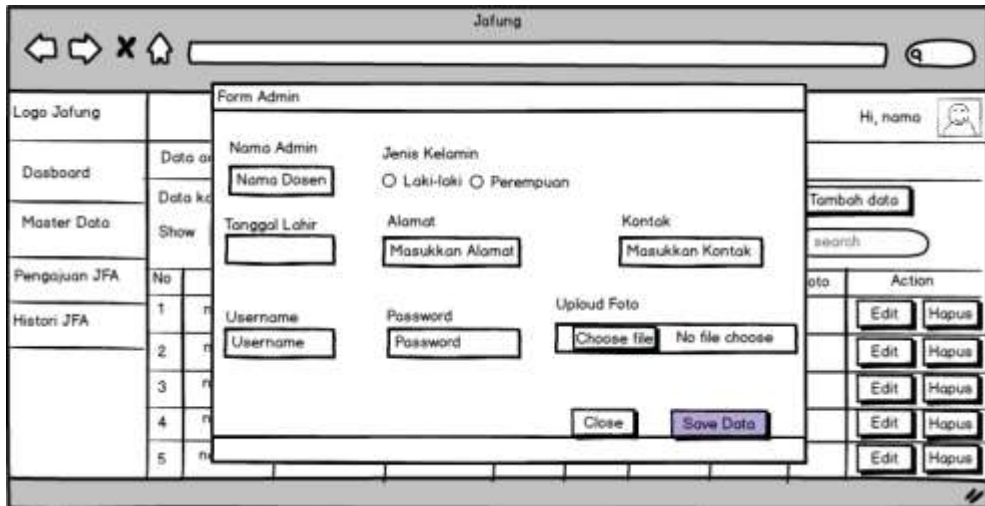
Halaman data admin ialah tampilan data admin yang dapat menggunakan website ini yang dapat dilihat pada gambar 4.18



Gambar 4.18 Desain Master data admin

j. Desain tambah data admin

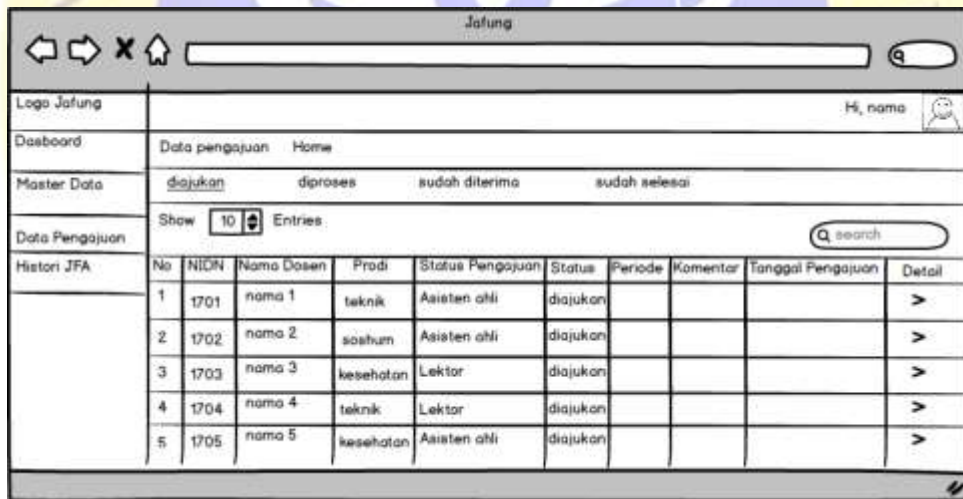
Form tambah data admin digunakan jika ada penambahan admin untuk mengoperasikan website yang fungsinya sama dengan register tetapi hanya admin yang dapat menambahkan admin lain dan dapat dilihat pada gambar 4.19



Gambar 4.19 Desain Tambah data admin

k. Desain Pengajuan

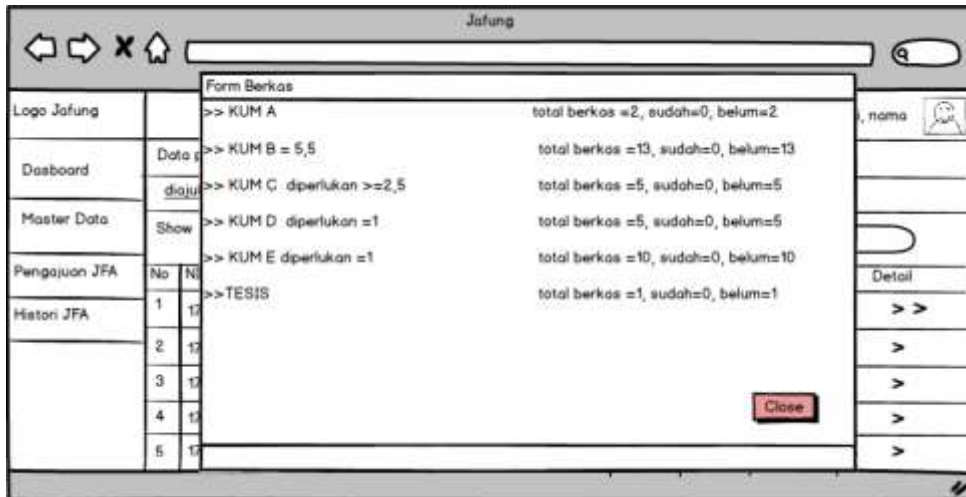
Halaman pengajuan adalah biodata dosen yang yang mengajukan pengajuan kenaikan jabatan dapat dilihat pada gambar 4.20



Gambar 4.20 Desain Pengajuan JFA

l. Desain detail pengajuan

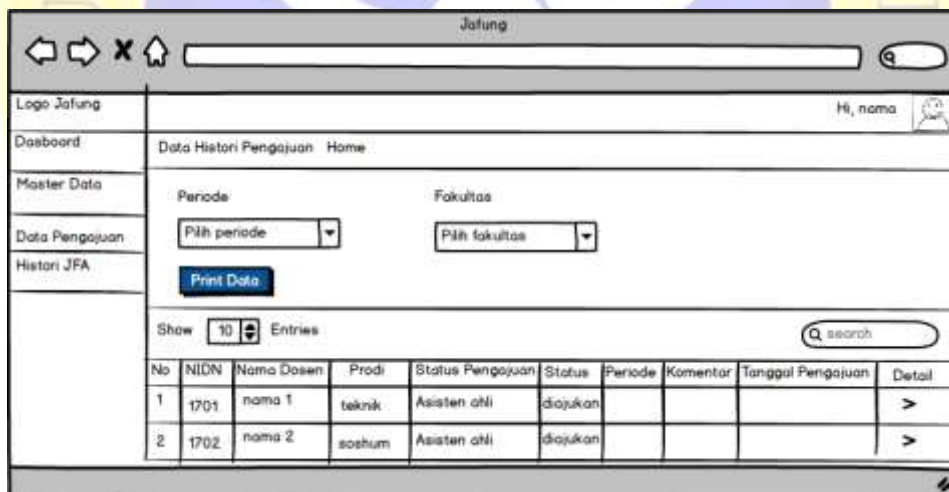
Halaman detail pengajuan berisi berkas-berkas persyaratan yang dibutuhkan dalam pengajuan kenaikan jabatan dosen yang dapat diperiksa dan mengirim komentar di halaman pengajuan jika terdapat kesalahan berkas atau berkas tidak lengkap yang dapat dilihat pada gambar 4.21



Gambar 4.21 Desain detail berkas pengajuan

m. Desain histori JFA

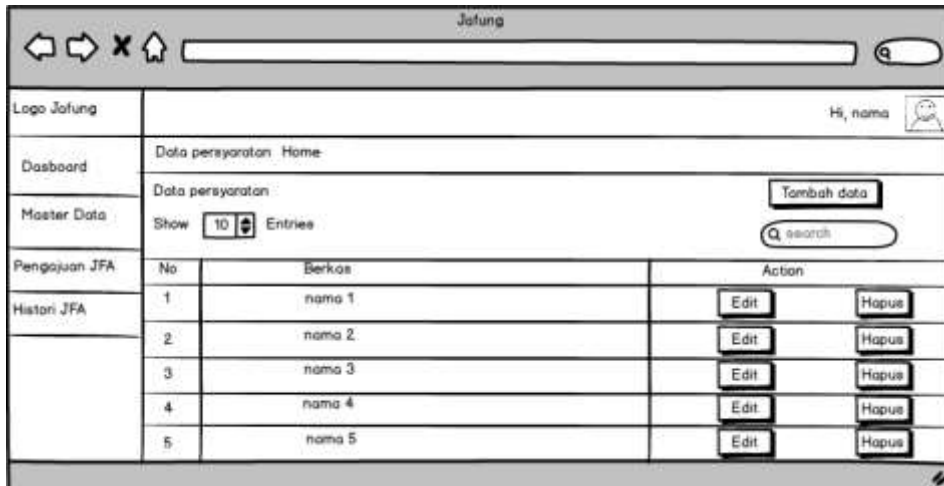
Halaman histori JFA berisi data dan berkas pengajuan dan juga dapat mencetak laporan berdasarkan data yang ada dengan memilih periode ataupun berdasarkan fakultas laporan yang ingin dicetak



Gambar 4.22 Desain histori JFA

n. Desain master data Persyaratan

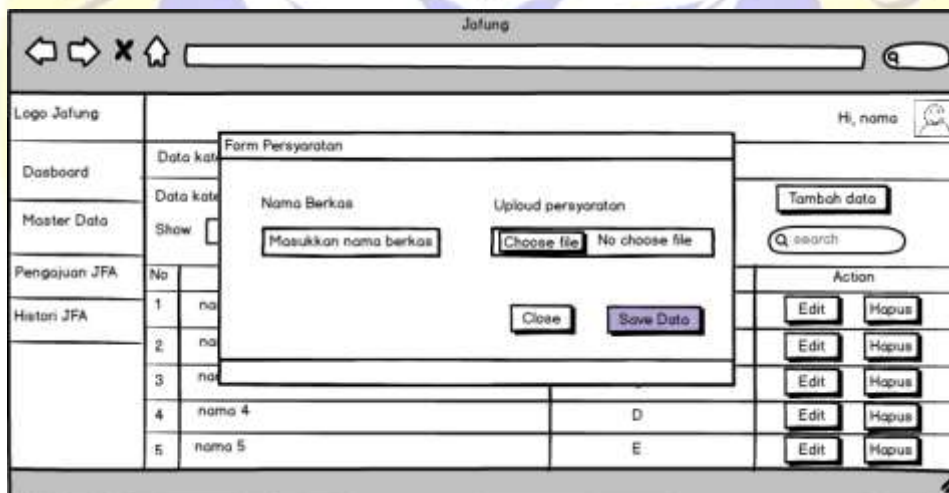
Desain data persyaratan berisi persyaratan seriap jabatan yang akan di tampilkan atau dikirim ke aplikasi jafung android yang digunakan dosen untuk informasi berkas yang harus disiapkan untuk pengajuan jabatan fungsional dosen



Gambar 4.23 Desain data persyaratan

- o. Desain master data tambah persyaratan

Desain tambah persyaratan digunakan untuk menambahkan persyaratan yang sesuai ketentuan



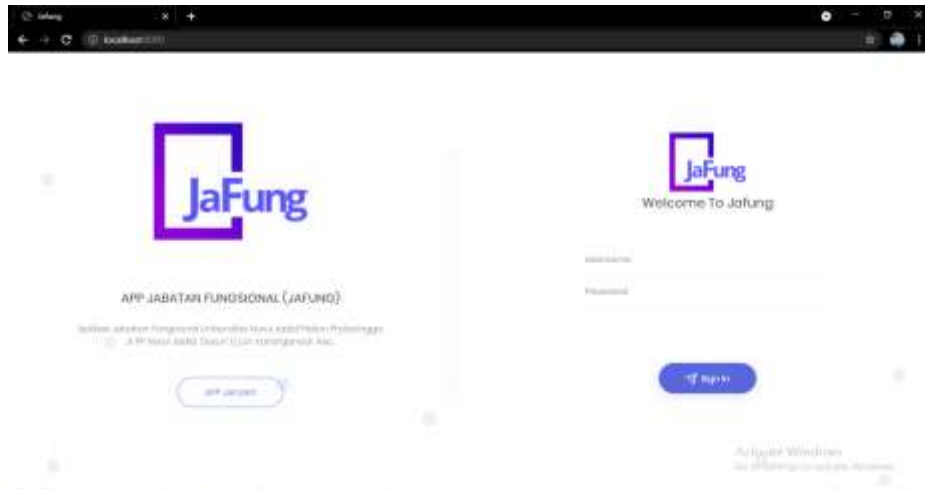
Gambar 4.24 Desain tambah persyaratan

### 4.3 Implementation

Setelah tahap desain dan perancangan alur sistem dilakukan, tahapan selanjutnya yaitu mengimplementasikan desain yang dibuat pada aplikasi yang akan digunakan. Pada penelitian ini menggunakan aplikasi *Visual Studio Code* untuk mengimplementasikan desain sistem dan untuk desain database menggunakan *MySql*. Aplikasi yang dikembangkan ialah monitoring

pengajuan jabatan fungsional dosen. Berikut ialah hasil dari implementasi monitoring pengajuan jabatan fungsional dosen :

#### 4.3.1 Halaman login



Gambar 4.25 Halaman Login

#### Segmen program 4.1 Halaman login

```
@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
```

```

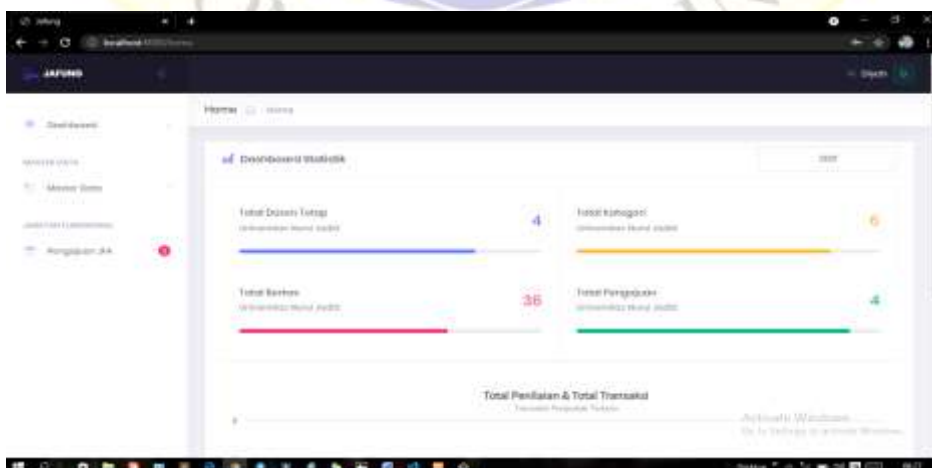
request.session["nidn"] = val.nidn
request.session["nama"] = val.nama_dosen
request.session["tgl"] = val.tgl_lahir.strftime("%Y-%m-%d")
request.session["gender"] = val.jns_kelamin
request.session["alamat"] = val.alamat
request.session["kontak"] = val.kontak
request.session["fakultas"] = val.fakultas
request.session["foto"] = ANDROID_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)

```

#### 4.3.2 Halaman home



Gambar 4.26 Halaman home

## Segmen program 4.2 Halaman home

```
def home(request):
```

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

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

```
    else:
```

```
        dosen = Dosen.objects.filter(status="D").all().count()
```

```
        kategori = ""
```

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

```
            con.execute("SELECT * FROM app_kategori GROUP BY  
kategori")
```

```
            kategori = con.rowcount
```

```
            berkas = Kategori.objects.all().count()
```

```
            permintaan = Pengajuan.objects.all().count()
```

```
            count = {
```

```
                "dosen": dosen,
```

```
                "kategori": kategori,
```

```
                "berkas": berkas,
```

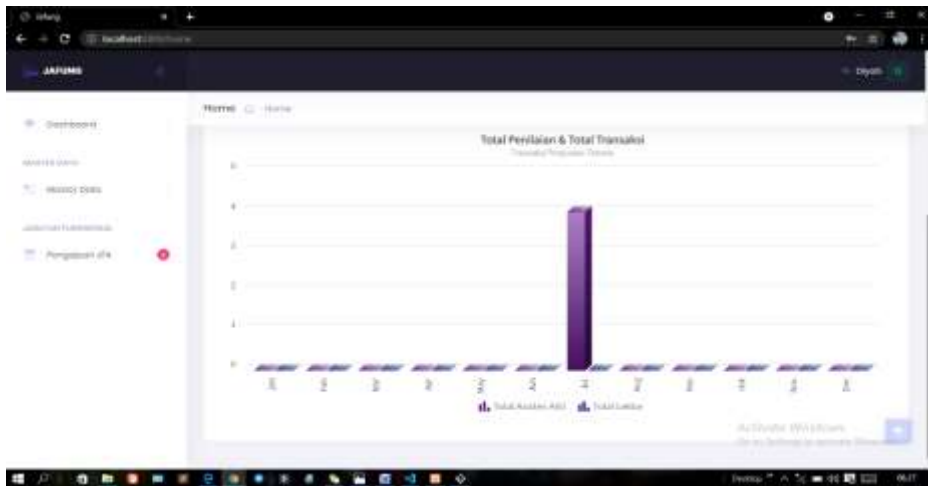
```
                "permintaan": permintaan,
```

```
                "url": "home",
```

```
            }
```

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

### 4.3.3 Halaman home grafik pengajuan



Gambar 4.27 Halaman home grafik pengajuan

### 4.3.4 Halaman master data dosen tetap

The screenshot shows a web dashboard with a sidebar on the left containing navigation items like 'Dashboard', 'Master Data', 'History Data', 'Jumlah Transaksi', and 'Pengajuan'. The main content area features a table titled 'Data Dosen Tetap' with a subtitle 'Terdapat Data Dosen Tetap'. The table has columns for 'No', 'NIP', 'Nama', 'Tanggal Lahir', 'Jenis Kelamin', 'Agama', 'Status', 'Pendidikan', 'Gelar', and 'Masa Kerja'. There are four rows of data. The dashboard footer includes 'A&B Prolog' and '© 2020 Prolog'.

No	NIP	Nama	Tanggal Lahir	Jenis Kelamin	Agama	Status	Pendidikan	Gelar	Masa Kerja
1	197003010121011001	121011	19700301	Laki-laki	-	-	TAMBAH DATA	MAK	MAK
2	197003010121011002	121012	19700301	Laki-laki	ISLAM	REKREANER/STAF	TAMBAH DATA	MAK	MAK
3	197003010121011003	121013	19700301	Perempuan	-	-	TAMBAH DATA	MAK	MAK
4	197003010121011004	121014	19700301	Perempuan	ISLAM	REKREANER/STAF	TAMBAH DATA	MAK	MAK

Gambar 4.28 Halaman master data dosen tetap

### Segmen Program 4.3 data dosen tetap

```
def dosen(request):
```

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

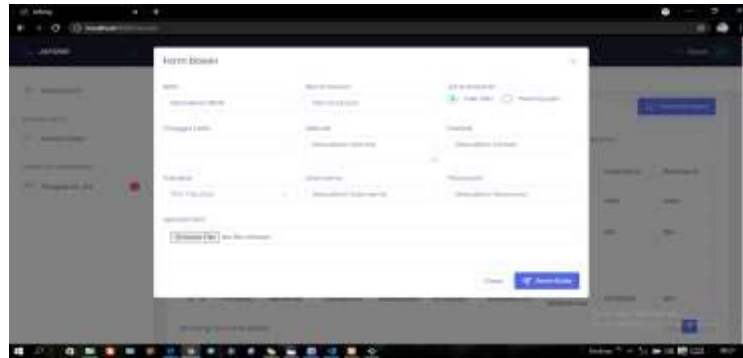
```
        return render(request, "index.html")
```

```
    else:
```

```
        return render(request, "dosen.html", {"url": "dosen"})
```



#### 4.3.5 Halaman tambah data dosen tetap



Gambar 4.29 Halaman tambah data dosen tetap

#### Segmen program 4.4 Tambah 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")
```

```

foto = request.POST.get("foto1 ")

# foto = request.FILES("foto")

if pk != "":

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

# upload file folder

path = "foto/"

file = request.FILES["foto"].name

ext = file.split(".")[1]

filename = str(uuid.uuid4())

full_filename = os.path.join(

    ANDROID_URL, path, filename + "." + ext)

cek = Dosen.objects.filter(pk=pk).all()

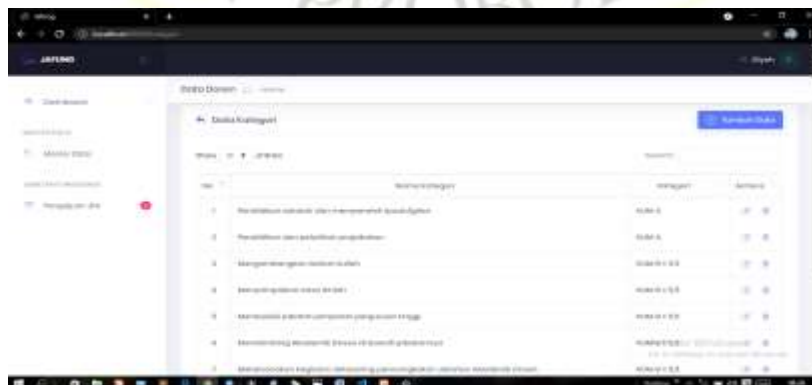
for gmb in cek.iterator():

    del full_filename = os.path.join(

        ANDROID_URL, path, str(gmb.foto))

```

#### 4.3.6 Halaman master data kategori



Gambar 4.30 Halaman master data kategori

### Segmen program 4.5 data kategori

```
def kategori(request):
```

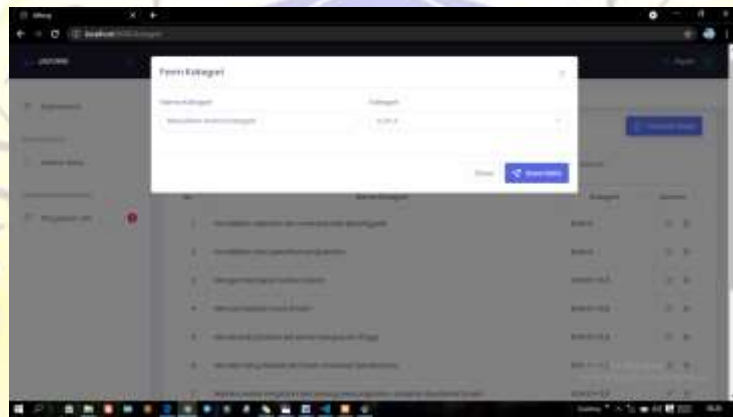
```
    if request.session["id"] == None:
```

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

```
    else:
```

```
        return render(request, "kategori.html", {"url": "kategori"})
```

#### 4.3.7 Halaman tambah data kategori



Gambar 4.31 Halaman tambah data kategori

### Segmen Program 4.6 Tambah data kategori

```
@csrf_exempt
```

```
def svkategori(request):
```

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

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

```
        if pk != "":
```

```
            editdata = Kategori.objects.get(pk=pk)
```

```
            save = KategoriForm(request.POST, instance=editdata)
```

```
else:
```

```
    save = KategoriForm(request.POST or None)
```

```
if save.is_valid:
```

```
    sql = save.save()
```

```
    if sql:
```

```
        arr = 1
```

```
    else:
```

```
        arr = 0
```

```
    else:
```

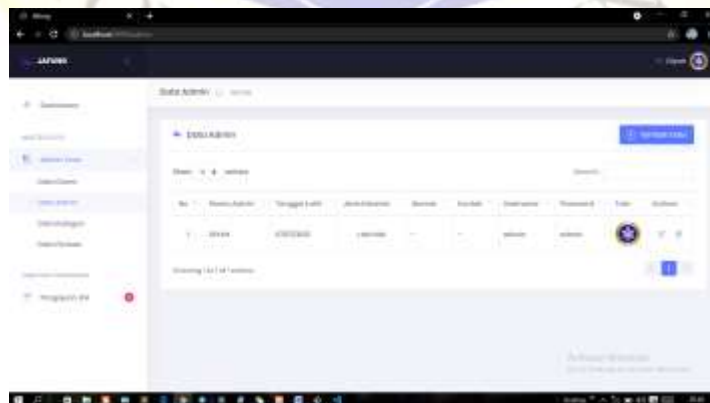
```
        arr = 0
```

```
    else:
```

```
        arr = 0
```

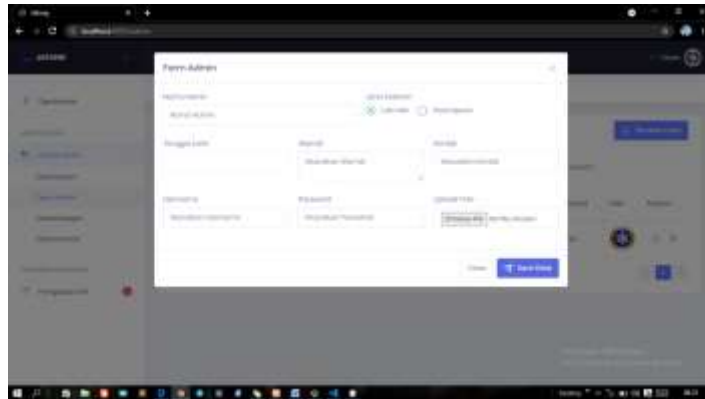
```
    return HttpResponse(arr)
```

#### 4.3.8 Halaman master data admin



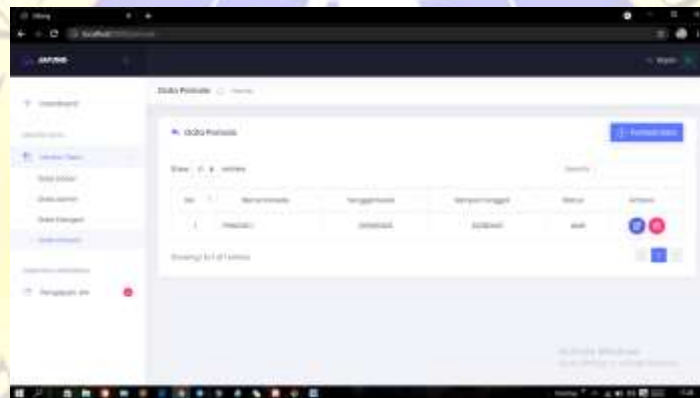
Gambar 4.32 Halaman master data admin

#### 4.3.9 Halaman Tambah data admin



Gambar 4.33 Halaman tambah data admin

#### 4.3.10 Halaman master data periode



Gambar 4.34 Halaman master data periode

#### **Segmen program 4.7** Data periode

```
def periode(request):
```

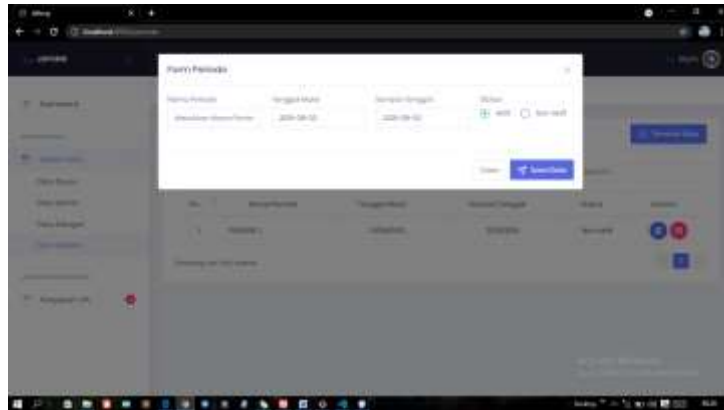
```
    if request.session["id"] == None:
```

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

```
    else:
```

```
        return render(request, "periode.html", {"url": "periode"})
```

#### 4.3.11 Halaman tambah data periode



Gambar 4.35 Halaman tambah data periode

#### Segmen program 4.8 Tambah data periode

```
@csrf_exempt
```

```
def svperiode(request):
```

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

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

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

```
            con.execute("UPDATE app_periode SET status='0' WHERE 1")
```

```
        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:
```

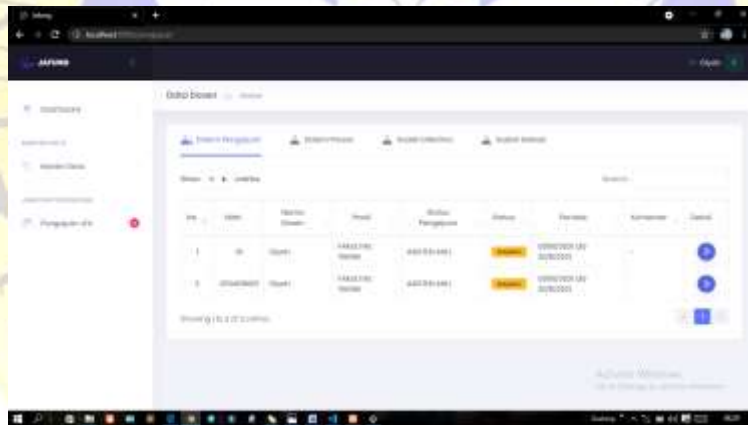
```
            sql = save.save()
```

```

if sql:
    arr = 1
else:
    arr = 0
else:
    arr = 0
else:
    arr = 0
return HttpResponse(arr)

```

#### 4.3.12 Halaman pengajuan JFA



Gambar 4.36 Halaman pengajuan JFA

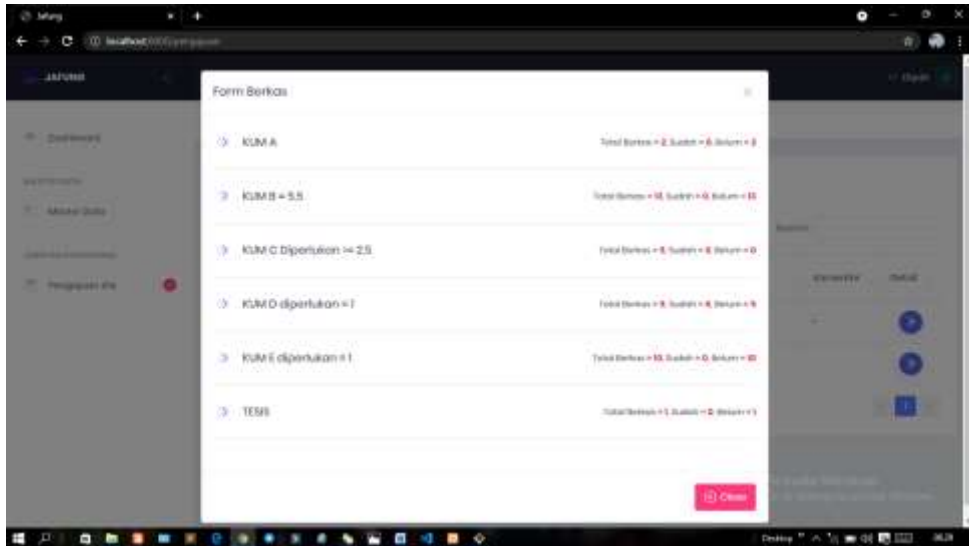
#### Segmen program 4.9 Pengajuan JFA

```

def pengajuan(request):
    if request.session["id"] == None:
        return render(request, "index.html")
    else:
        return render(request, "pengajuan.html", {"url": "pengajuan"})

```

### 4.3.13 Halaman detail pengajuan



Gambar 4.37 Halaman detail pengajuan

#### Segmen program 4.10 Detail pengajuan

@csrf\_exempt

```
def vadddetail(request):
```

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

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

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

```
        data = []
```

```
        out = []
```

```
        no = 1
```

```
        status = ""
```

```
        berkas = ""
```

```
        sql = Kategori.objects.filter(kategori=kat).all()
```

```
        for value in sql.iterator():
```



```

dtl = (
    DetailPengajuan.objects.filter(kategori=value.id)
    .filter(kode_pengajuan=kode)
    .exists()
)

nama = value.nama_kategori.upper()
kat = value.kategori

if dtl:
    status = '<center><span class="kt-badge kt-badge--inline kt-
badge--success">Sudah</span></center>'

    query = (
        DetailPengajuan.objects.filter(kategori=value.id)
        .filter(kode_pengajuan=kode)
        .all()
    )

    for v in query.iterator():

        # berkas = v.berkas

        berkas = (

            '<center><a href="'

            + ANDROID_URL

            + "pdf/"

```

```

        + v.berkas

        + "' target='_blank'>View Berkas</a></center>'

    )

else:

    status = '<center><span class="kt-badge kt-badge--inline kt-
badge--danger">Belum</span></center>'

    berkas = '<center><input type="file" class="form-control"
id="upload" data-id="' + \

str(value.id)+'" accept="application/pdf"></center>'

    out = [

        no,

        kat,

        nama,

        berkas,

        status,

    ]

    data.append(out)

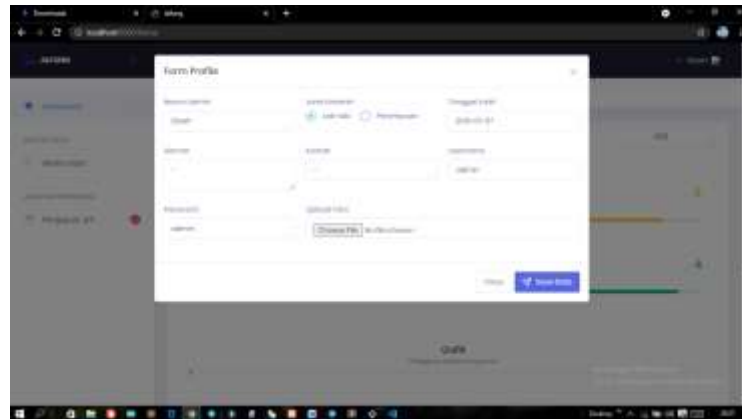
    no += 1

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

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

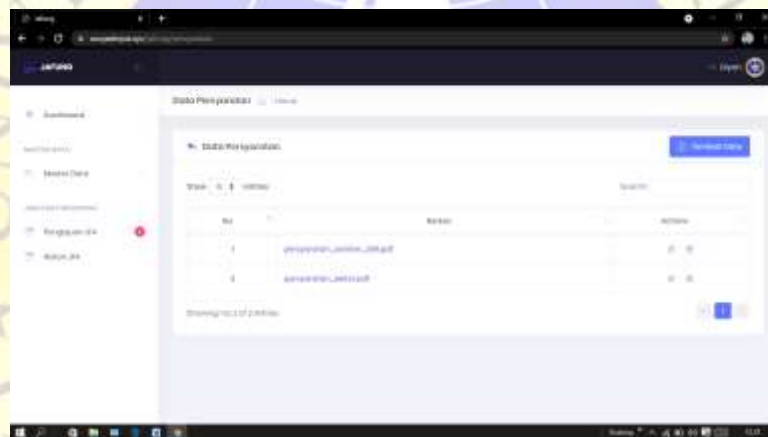
```

#### 4.3.14 Halaman profil



Gambar 4.38 Halaman profil

#### 4.3.15 Halaman master data persyaratan



Gambar 4.39 Halaman master data persyaratan

#### Segmen program 4.11 Data persyaratan

```
def persyaratan(request):
```

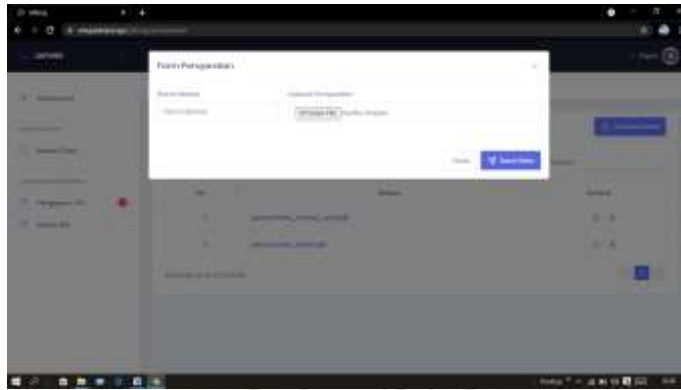
```
    if request.session["id"] == None:
```

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

```
    else:
```

```
        return render(request, "persyaratan.html", {"url": "persyaratan"})
```

#### 4.3.16 Halaman tambah data persyaratan



Gambar 4.40 Halaman tambah data persyaratan

#### Segmen program 4.12 Tambah data persyaratan

```
@csrf_exempt
```

```
def svpersyaratan(request):
```

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

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

```
        berkas = request.POST.get("nama")
```

```
        foto = request.POST.get("file1")
```

```
        if pk != "":
```

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

```
            # upload file folder
```

```
            path = "persyaratan/"
```

```
            file = request.FILES["file"].name
```

```
            ext = file.split(".")[-1]
```

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

```
                "/", "_").replace(".", "_")
```

```

full_filename = os.path.join(
    ANDROID_URL, path, filename + "." + ext)

cek = Persyaratan.objects.filter(pk=pk).all()

for gmb in cek.iterator():

    delfull_filename = os.path.join(
        ANDROID_URL, path, str(gmb.berkas))

    if foto == "":
        data.berkas = None
    else:
        data.berkas = filename+'.'+ext

    data.save()

    if foto != "":
        # hapus File
        if os.path.isfile(delfull_filename):
            os.remove(delfull_filename)

        # 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()

arr = 1

else:

    if request.POST.get("file1") == "":

        save = Persyaratan(
            berkas="0"
        )

    else:

        # upload file folder

        path = "persyaratan/"

        file = request.FILES["file"].name

        ext = file.split(".")[1]

        filename = berkas.replace(" ", "_").replace(
            "/", "_").replace(".", "_")

        full_filename = os.path.join(

            ANDROID_URL, path, filename + "." + ext)

    if foto != "":

        # 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()

save = Persyaratan(

    berkas=filename+'.'+ext,

)

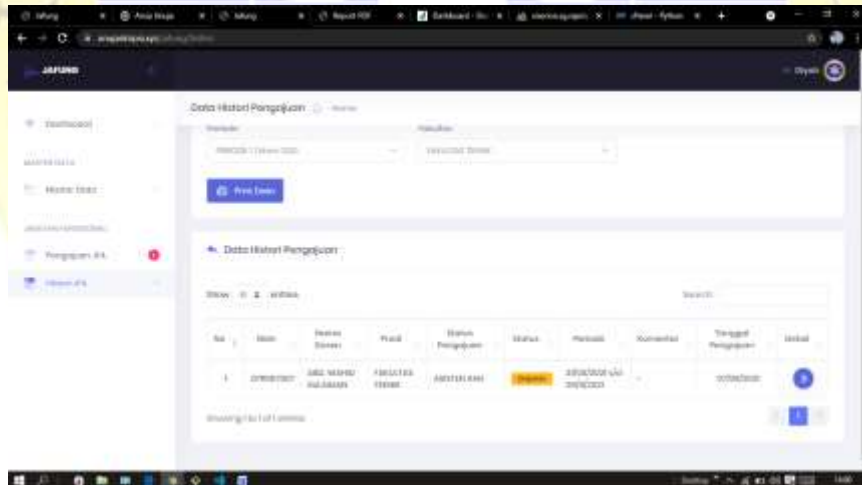
save.save()

arr = 1

return HttpResponse(arr)

```

#### 4.3.17 Halaman histori JFA



Gambar 4.41 Gambar halaman histori JFA

#### Segmen program 4.13 Histori JFA

```

def histori(request):

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

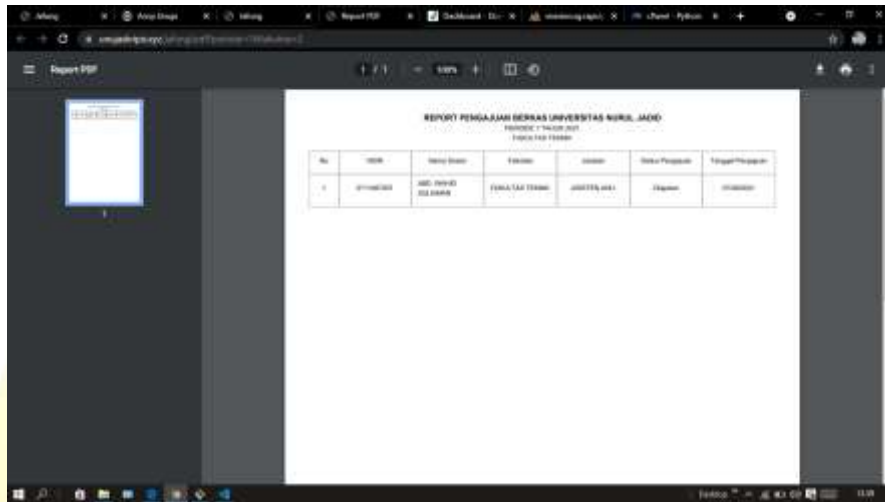
```

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

else:

```
return render(request, "histori.html", {"url": "histori"})
```

#### 4.3.18 Laporan pdf



Gambar 4.42 Gambar laporan pdf

## 4.4 Testing

Uji coba yang akan dilakukan untuk pengujian internal menggunakan black box testing dan pengujian eksternal melakukan pengujian langsung kepada user. Pada pengujian eksternal ini penguji diberi angket, dimana nilai dari hasil angket tersebut akan dihitung menggunakan skala likert. dengan diberikan beberapa pertanyaan yang akan dijawab langsung oleh user. Dari perhitungan skala likert tersebut akan diketahui keberhasilan sistem yang diuji dengan melihat output yang dihasilkan dari kondisi input yang sudah dirancang sesuai dengan fungsi yang sudah ditentukan. Berikut merupakan perolehan hasil dari uji coba dengan keterangan sebagai berikut :

### 4.4.1 Pengujian Internal

Pengujian internal dilakukan secara langsung kepada Bapak Abu Tholib, M.Kom selaku dosen di fakultas teknik Universitas Nurul Jadid. Tampilan hasil uji dari pengujian internal dapat dilihat pada tabel berikut



Tabel 4.11 Tabel pengujian internal

No	Form	Fungsi	Hasil yang diharapkan	Hasil Uji	
				Sesuai	Tidak
1	Login	Username dan password	Jika benar maka akan menampilkan halaman utama/beranda Jika salah tidak dapat masuk ke website	√	
2	Dashboard		Menampilkan halaman dashboard	√	
3	Master Data	Dosen tetap	Menampilkan semua data dosen tetap	√	
		Data kategori	Menampilkan data kategori berkas persyaratan pengajuan	√	
		Data admin	Menampilkan data admin	√	
		Data periode	Menampilkan data periode	√	
4	pengajuan JFA		Menampilkan data dosen yang mengajukan pengajuan jabatan	√	

			fungsi dosen beserta dapat menentukan status pengajuan dan mengirim komentar		
		Detail	Menampilkan berkas-berkas persyaratan yang diajukan oleh dosen	√	
5	Logout		Keluar dari akun dan kembali ke halaman login	√	
6	My profil		Menampilkan profil user dan dapat mengedit profil	√	

Berdasarkan hasil pengujian menggunakan metode balckbox yang dilakukan kepada bapak Abu Tholib, M.Kom dapat disimpulkan bahwa hasil input dan output program pada sistem sudah sesuai dan juga terdapat komentar pada aplikasi yang dikhususkan untuk admin sehingga admin memonitoring kelengkapan persyaratan pengajuan jabatan fungsional dosen.

#### 4.4.2 Pengujian eksternal

Pengujian yang dilakukan langsung kepada Tim PAK sebanyak 3 dengan dibagikan kuesioner secara langsung. Terdapat 6 pernyataan pada tabel pengujian ini yang dapat dilihat pada tabel dibawah ini :

Tabel 4.12 Tabel pengujian eksternal

No	Pertanyaan	Jawaban				
		SB	B	C	TB	STB
	Skor nilai	5	4	3	2	1
1	Apakah aplikasi ini sudah sesuai dengan kebutuhan?	3				
2	Apakah aplikasi ini sudah layak digunakan?	1	2			
3	Apakah desain tampilan aplikasi ini sudah memudahkan pengguna?		3			
4	Apakah aplikasi ini dapat membantu dalam memonitoring data persyaratan pengajuan jabatan fungsional?	3				
5	Apakah ada kesulitan dalam pengoprasian aplikasi ini?		3			
6	Apakah aplikasi ini dapat menjawab permasalahan dalam mengajukan jabatan fungsional ?		2	1		

Tabel 4.13 Perhitungan skala likert

No	Skala Jawaban	Skor
1	Sangat Baik (SB)	$5 \times 7 = 35$
2	Baik (B)	$4 \times 10 = 40$
3	Cukup(C)	$3 \times 1 = 3$
4	Tidak Baik (TB)	$2 \times 0 = 0$
5	Sangat Tidak Baik (STB)	$1 \times 0 = 0$
	Total Skor	78

Jumlah skor ideal = skor tertinggi x Jumlah soal x Jumlah responden

$$5 \times 6 \times 3 = 90$$

Tabel 4.14 Kriteria Interpretasi persen

No	Persentase	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 kuesioner yang peneliti lakukan adalah Penentuan nilai presentase = (Total Skor yang diperoleh / Skor ideal) x 100, Sehingga di dapat nilai presentase =  $(78/90) \times 100 = 86\%$ . Berdasarkan tabel presentase nilai diatas dapat diambil kesimpulan bahwa “Aplikasi monitoring pengajuan jabatan fungsional dosen universitas nurul jadid berbasis web framework django” mendapatkan persentase 86,66% yang menurut tabel kriteria interpretasi menunjukkan bahwa aplikasi ini mendapatkan kategori sangat baik.

#### 4.5 Maintenance

Dalam tahap ini maintenance/perawatan perangkat lunak aplikasi monitoring pengajuan jabatan fungsional dosenakan di lakukan dengan *update* sistem jika terdapat penambahan fitur yang dibutuhkan. *Update* nama pengguna dan kata sandi jika terdapat perubahan petugas untuk mencegah terjadinya *hacking* pada sistem dan memperbaiki ketika terjadi kesalahan pada sistem.