

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Hasil Pengumpulan Data**

Hasil pengumpulan data dalam penelitian ini diperoleh dari kegiatan observasi dan wawancara kepada pihak yang bersangkutan dengan pengurus pondok pesantren.

##### **4.1.1 Hasil Observasi**

Berdasarkan observasi yang dilakukan di pondok pesantren Mambaul Ulum Paiton menghasilkan bahwa dalam proses pendaftaran dan administrasi oleh pengurus pesantren masih dilakukan secara konvensional dimana dalam proses pencatatan ditulis kedalam buku besar lalu diketik kembali ke excel sebagai salinan dari buku besar tadi. Hal ini dapat menyebabkan sulitnya pencarian dan pengecekan data sehingga dapat terjadi kehilangan data, tidak adanya data cadangan apabila mengalami kehilangan data, sulitnya merekap keuangan pesantren yang memicu terhadap seringnya terjadi kesalahan dalam perhitungan data keuangan dalam pembuatan laporan, dan juga pada proses rekapitulasi tanggungan santri sehingga banyak santri yang tidak tahu terkait jumlah tanggungan yang dimilikinya. Kedua, kegiatan pendaftaran santri baru yang masih dilakukan dengan cara manual yakni dengan datang langsung ke pesantren kemudian mengisi formulir. Terakhir, data santri bukan hanya data santri yang ada di pondok pesantren melainkan juga data santri dari luar pesantren yang sekolah di Lembaga dibawah naungan pondok pesantren.

##### **4.1.2 Hasil Wawancara**

Hasil wawancara yang dilakukan pada tanggal 17 Maret 2021 kepada pengurus pesantren bagian sekretaris dan bendara, menghasilkan bahwa pondok pesantren Mambaul Ulum memiliki 3 Wilayah yaitu wilayah Al-Khoirot (dalem selatan), Abdullah (dalem timur), Al-Hikmah (dalem utara) dengan memiliki 530 santri dari 3 wilayah tersebut. Ada beberapa jenis administrasi pembayaran yang ada di pesantren meliputi pembayaran sumbangan pembinaan pesantren (SPP), pembayaran loker/lemari, pembayaran ekstra, pembayaran uang kantin (kos

makan) dan lain-lain. Lalu pendaftaran santri baru yang masih dilakukan dengan cara manual yakni datang langsung ke bagian pendaftaran untuk mengisi formulir dan memenuhi persyaratan pendaftaran seperti foto copy KK, KTP orang tua/ wali santri, Akta Kelahiran santri.

## **4.2 Hasil Analisis dan Desain**

### **4.2.1 Analisis Sistem Lama**

Dalam proses administrasi pembayaran sumbangan pembinaan pesantren (SPP), pembayaran loker/lemari, pembayaran ekstra, pembayaran uang kantin (kos makan). Santri atau wali santri diwajibkan membawa kartu pembayaran kemudian diserahkan kepada bendahara pesantren untuk diproses kembali sesuai dengan jenis pembayaran yang akan dilakukan. Setelah selesai diproses kartu pembayaran tersebut diserahkan kembali kepada santri atau wali santri untuk disimpan sebagai tanda bukti pembayaran, kemudian petugas menginput data pembayaran tersebut ke dalam buku besar pembayaran dan *Microsoft Excel*. Kedua, proses pendaftaran santri baru, santri atau wali santri harus datang ke pesantren untuk mengambil dan mengisi formulir pendaftaran, setelah santri sudah siap untuk berangkat mondok di pesantren, santri tersebut wajib membawa formulir pendaftaran yang telah diisi dan persyaratan yang harus dipenuhi kemudian diserahkan kepada petugas pendaftaran. Lalu petugas menginputkan data santri baru tersebut ke dalam buku besar data santri dan *Microsoft Excel*.

### **4.2.2 Analisis Sistem Baru**

Aplikasi ini mampu melakukan proses administrasi pembayaran dan pendaftaran santri baru secara efektif dan efisien. Petugas tidak perlu mencatat proses pembayaran dan pendaftaran kedalam buku besar melainkan santri baru yang sudah melakukan proses pendaftaran secara online datanya langsung masuk kedalam data pendaftaran santri baru dan akan diproses oleh pengurus pesantren untuk menginput ID santri, wilayah, gang, kamar sesuai dengan yang ditempati santri tersebut. Begitupun dengan administrasi pembayaran, santri datang kepada pengurus pesantren bagian keuangan atau bendahara, kemudian petugas bendahara tersebut menginputkan form pembayaran yang akan santri lakukan, disana ada status santri itu mau melakukan pembayaran lunas, dispen atau belum

bayar. ID yang di inputkan pengurus tersebut digunakan untuk login santri disana akan menampilkan semua santri yang melakukan pembayaran, kemudian klik di tombol cari kemudian ketikan ID santri yang mau mengecek tunggakannya

#### 4.2.3 Desain Sistem

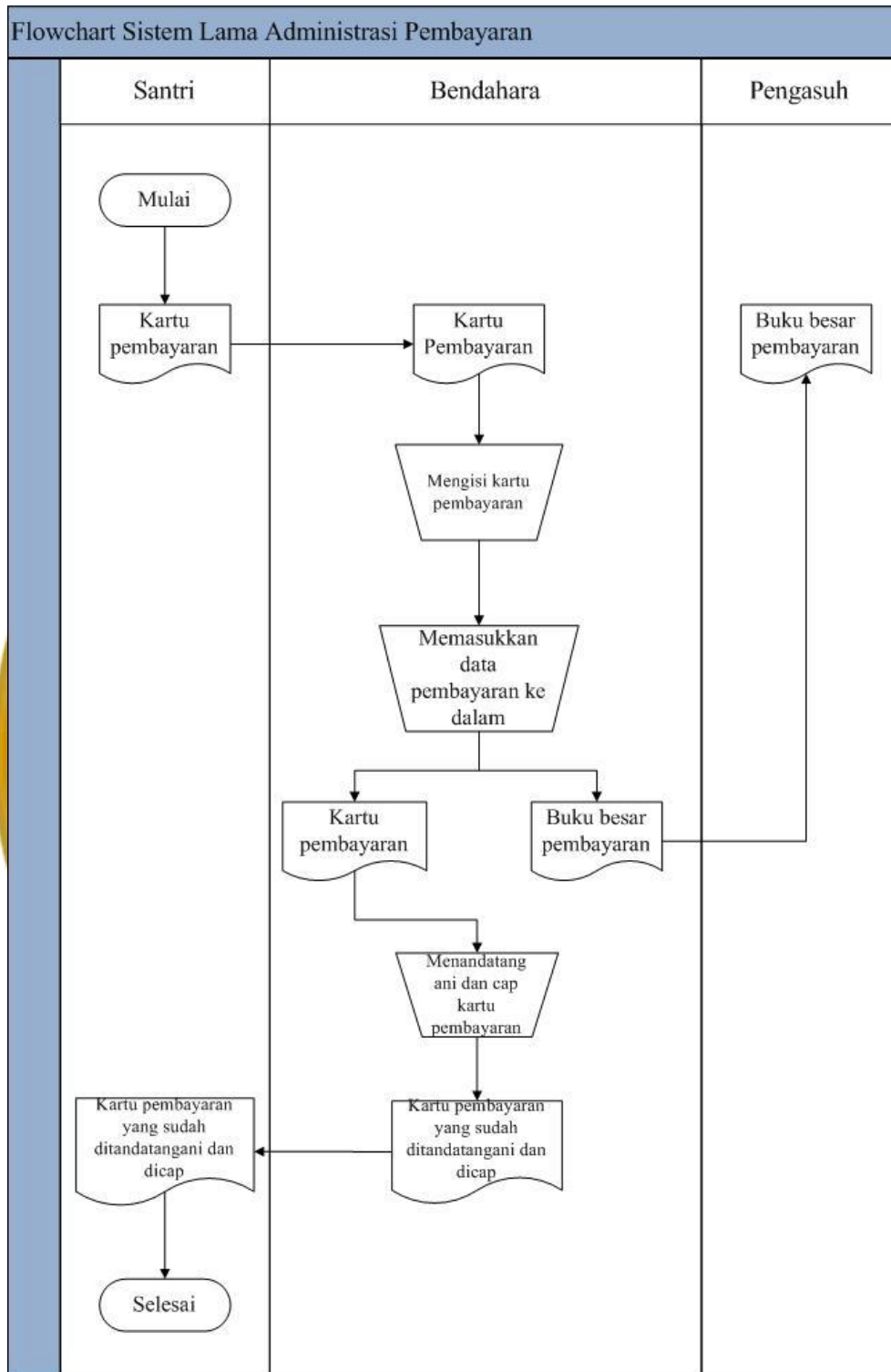
Dalam penelitian ini, desain sistem dibuat dengan perancangan bagan alur (*flowchart*), *Data Flow Diagram (DFD)*, *Entity Relayionship Diagram (ERD)*, serta dilengkapi dengan desain database dan desain program.

##### 1. Bagan Alur (*flowchart*)

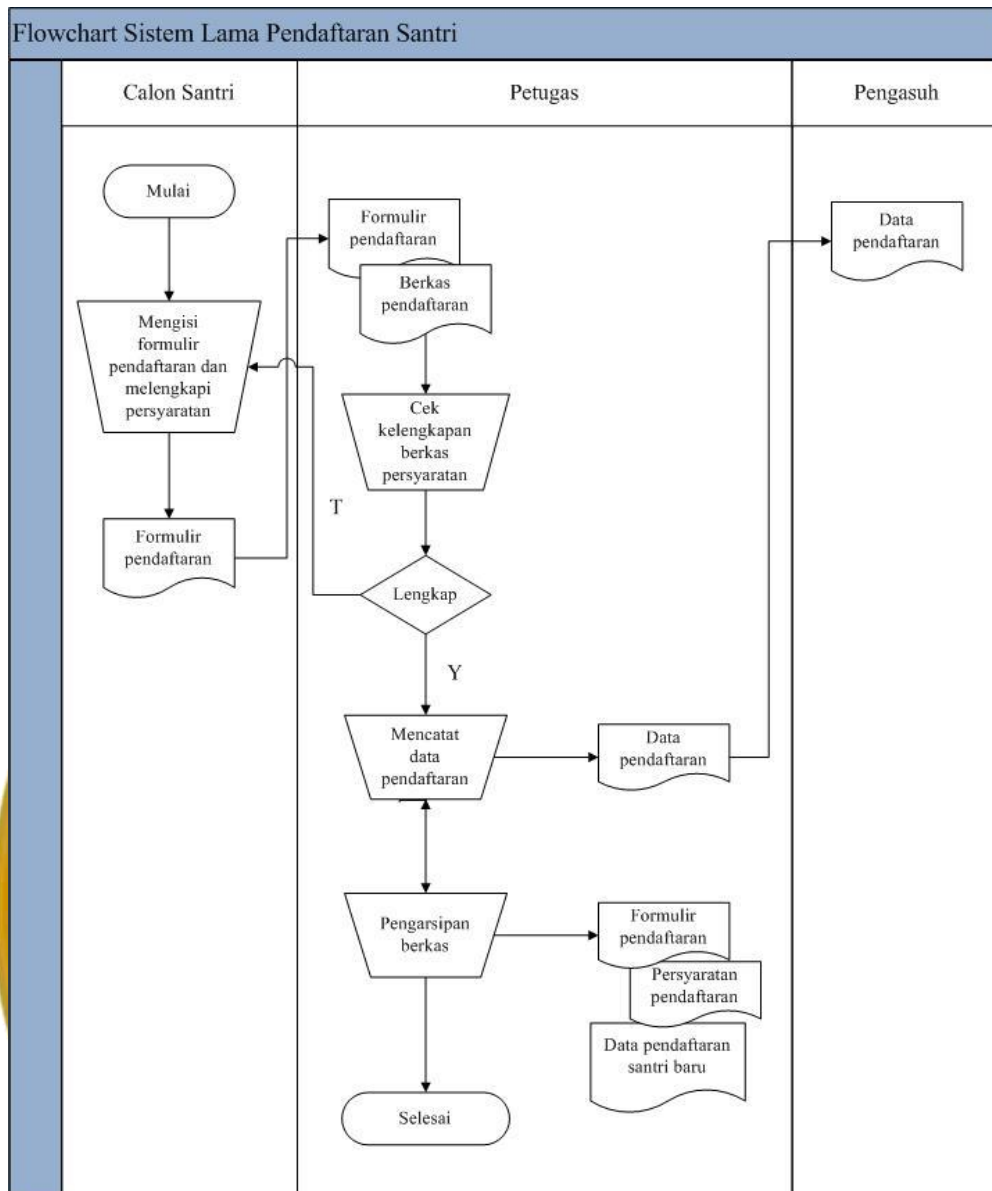
*Flowchart* sistem lama dan sistem baru pada proses administrasi pembayaran dan pendaftaran santri baru adalah sebagai berikut :



a. Flowchart sistem lama

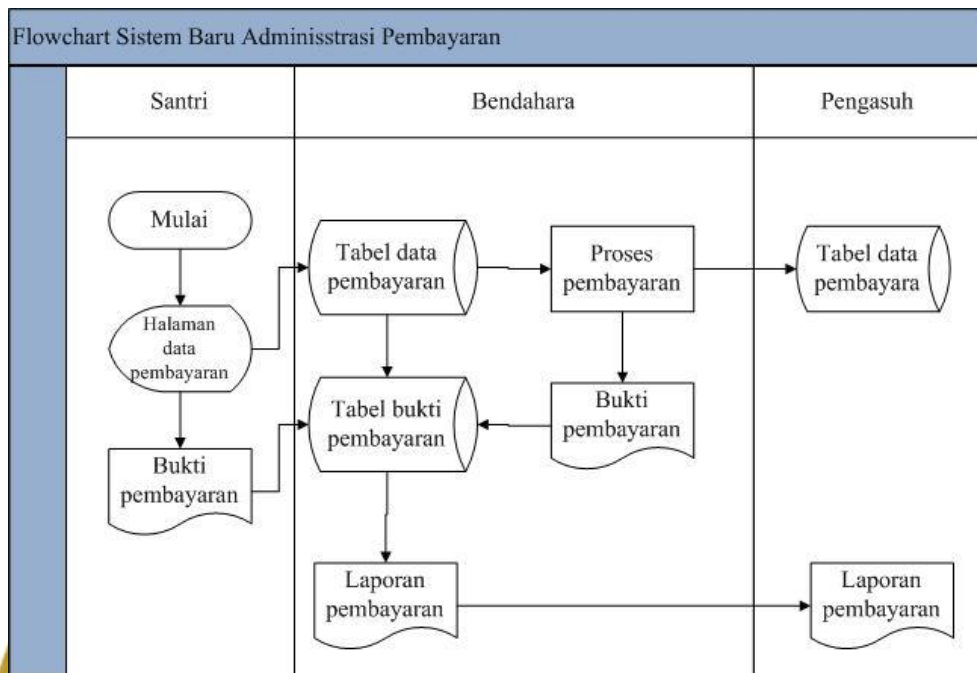


Gambar 4. 1. Flowchart sistem lama administrasi



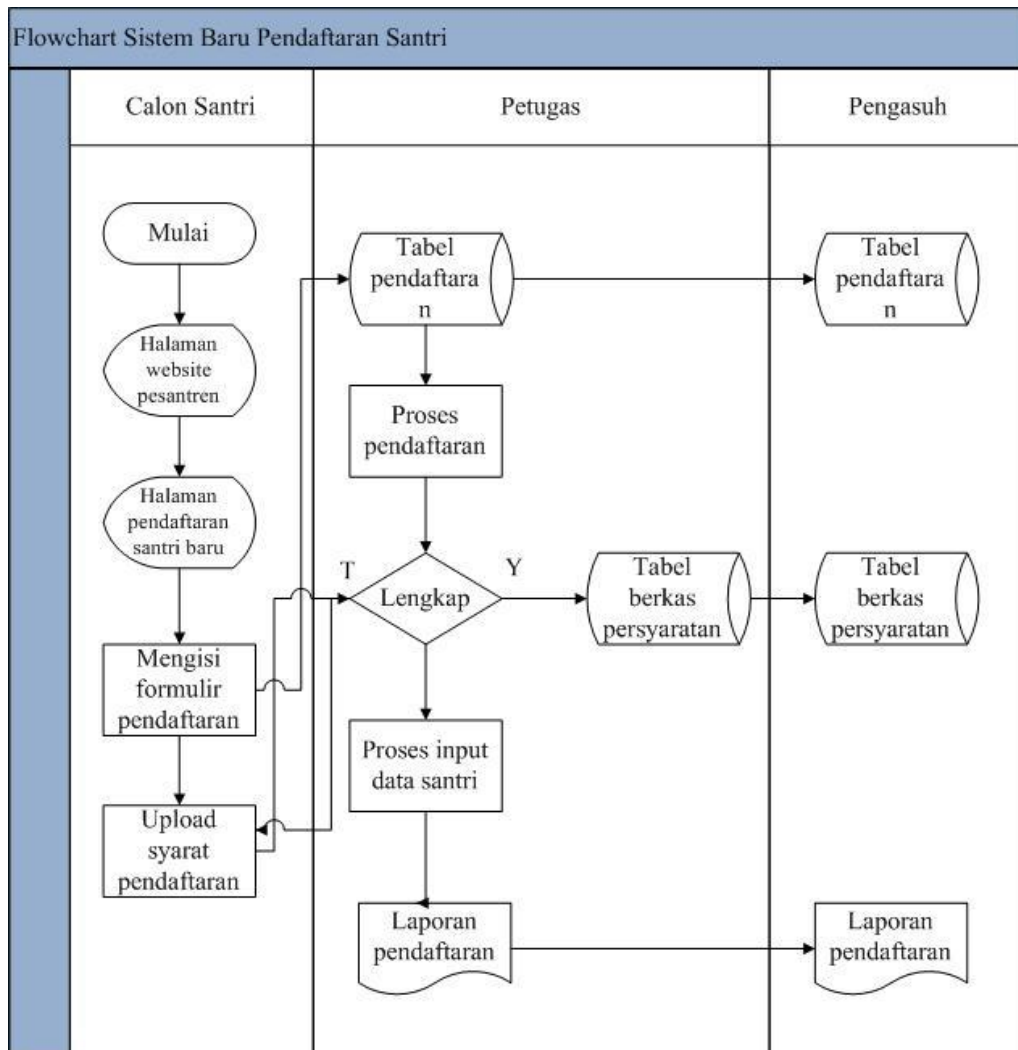
Gambar 4.2. Flowchart sistem lama pendaftaran

a. *Flowchart* sistem baru



Gambar 4. 3. Flowchart sistem baru administrasi





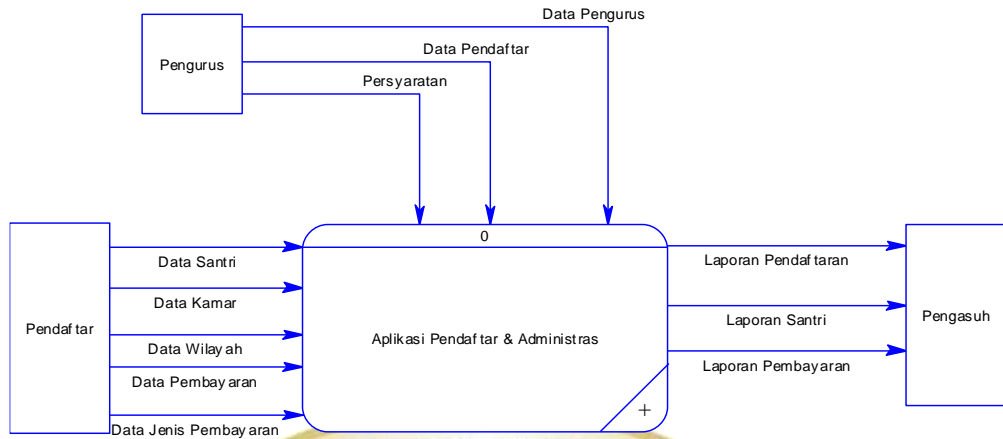
**Gambar 4. 4.** Flowchart sistem baru pendaftaran

## 2. Data Flow Diagram (DFD)

*Data Flow Diagram* adalah salah satu perancangan yang dapat menentukan bagaimana sistem akan berjalan sehingga dari suatu sistem aplikasi dapat memudahkan dalam pengolahan pendaftaran dan administrasi pembayaran di pondok pesantren mambaul ulum.

### a. Diagram Konteks

*Diagram* konteks merupakan suatu sistem perancangan yang menggambarkan alur suatu hubungan entitas ke entitas lain sehingga dalam alur suatu sistem dapat diketahui.

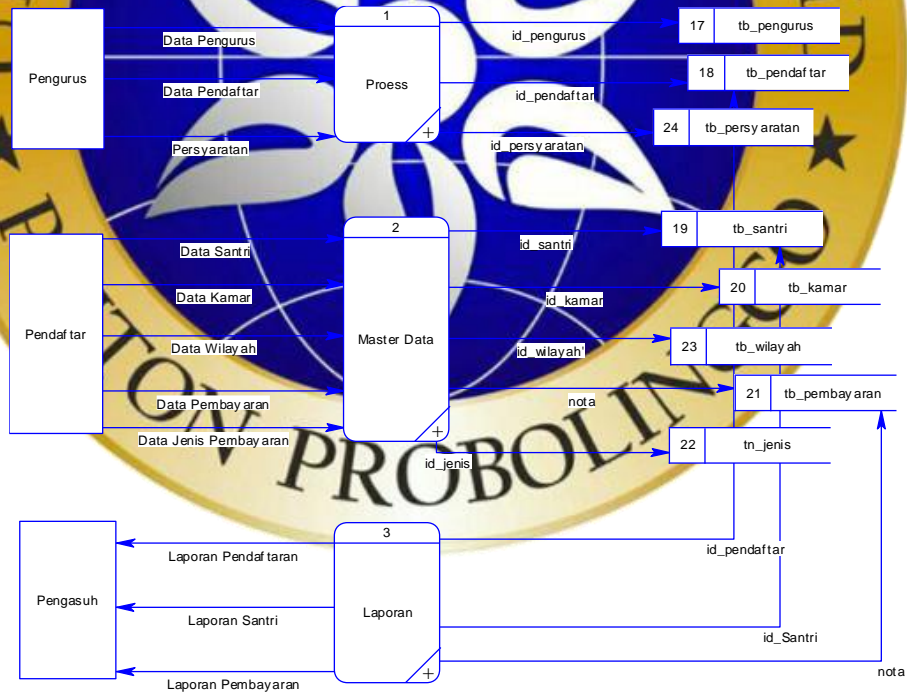


Gambar 4. 5. Diagram Konteks

b. Data Flow Diagram

1) DFD level 1

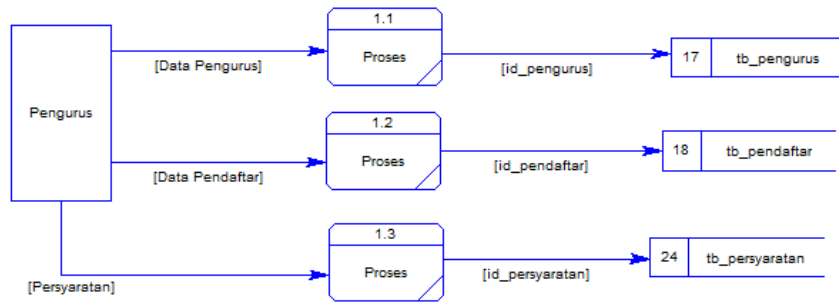
DFD level 1 akan menjelaskan secara lebih rinci konteks diagram aplikasi administrasi pembayaran dan pendaftaran santri baru ke dalam beberapa proses sesuai dengan kebutuhan sistem.



Gambar 4. 6. DFD Level 1

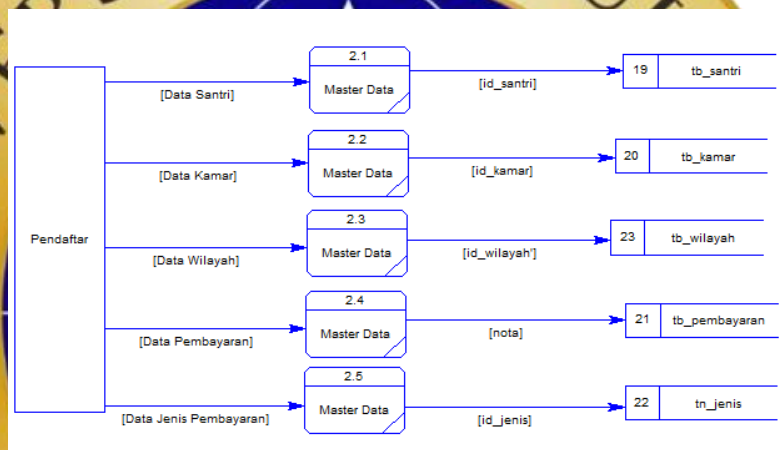


2) DFD level 1 Proses 1



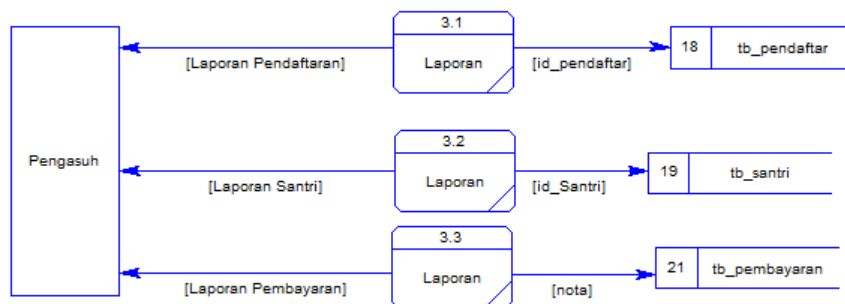
Gambar 4.7. DFD Level 1 Proses 1

3) DFD level 1 Proses 2



Gambar 4.8. DFD Level 1 Proses 2

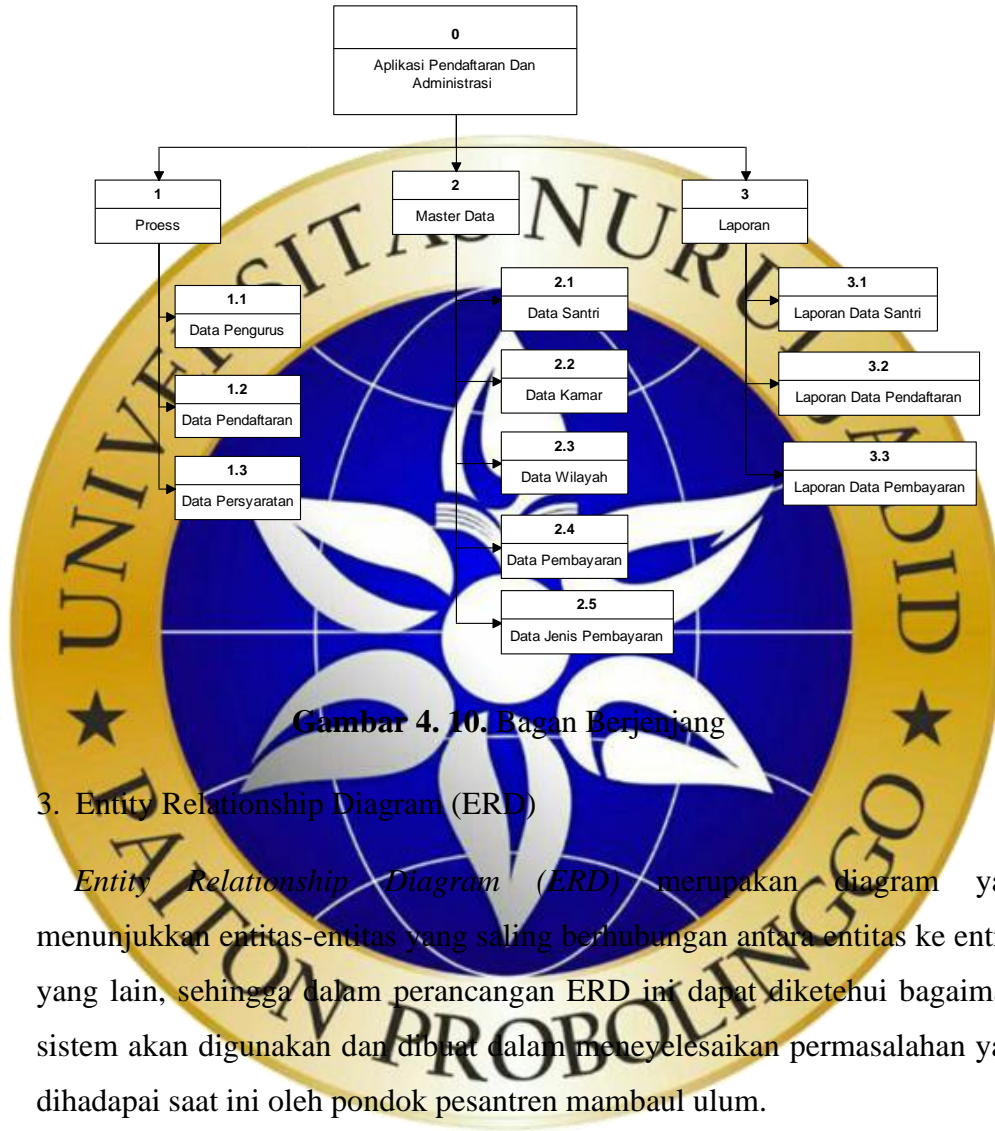
4) DFD level 1 Proses 3



Gambar 4.9. DFD Level 1 Proses 3

## 5) Bagan Berjenjang

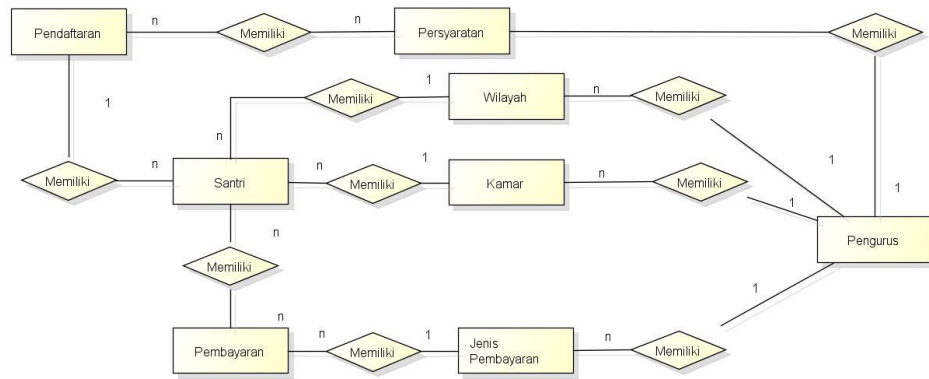
Bagan berjenjang adalah perancangan yang menggambarkan seperti struktur data, dan bagan berjenjang ini bagian dari perancangan sistem DFD.



Gambar 4. 10. Bagan Berjenjang

## 3. Entity Relationship Diagram (ERD)

*Entity Relationship Diagram (ERD)* merupakan diagram yang menunjukkan entitas-entitas yang saling berhubungan antara entitas ke entitas yang lain, sehingga dalam perancangan ERD ini dapat diketahui bagaimana sistem akan digunakan dan dibuat dalam menyelesaikan permasalahan yang dihadapi saat ini oleh pondok pesantren mambaul ulum.



Gambar 4.11. ERD

Kamus Data:

Data pengurus : id, username, password.

Data kamar : nama\_kamar, gang, kapasistas.

Data wilayah : nama\_wilayah, keterangan.

Data santri : noktp, idsantri, nama, alamat, tempat\_lahir, tgl\_lahir, jk, pendidikan, kamar, gang, wilayah, nama\_ortu, bulan\_pendaftaran.

Data pendaftaran : noktp, namasantri, alamat, tempat\_lahir, tgl\_lahir, jk, nohp, nama\_ortu, bulan\_pendaftaran.

Data pembayaran : idsantri, namasantri, alamat, kamar, gang, wilayah, nama\_jenis, pembayaran, bayar, status, tanggal\_pembayaran, bulan\_pembayaran.

Jenis pembayaran : nama\_jenis, pembayaran, tanggal.

#### 4.2.4 Desain Database

Desain database adalah merupakan suatu tempat dalam penyimpanan data dan memberikan ruang untuk menyimpan suatu data yang dikelola oleh pengurus, dalam desain database ini dibuat menggunakan sistem django. Berikut tabel-tabel.

a. Tabel Petugas

Tabel ini berisi data petugas untuk digunakan menyimpan data petugas. Atribut pada tabel petugas bisa dilihat dibawah ini

**Tabel 4. 1.** Tabel Petugas

<b>Tabel Petugas</b>		
<b>No</b>	<b>Nama Field</b>	<b>Type</b>
<b>1</b>	<b>Id</b>	<b>Varchar(12)</b>
<b>2</b>	<b>Username</b>	<b>CharFiled(120)</b>
<b>3</b>	<b>Password</b>	<b>CharFiled(120)</b>

b. Tabel Pendaftaran

Tabel pendaftaran ini akan memberikan ruang penyimpanan data pendaftaran, sehingga pengurus dapat mengecek data-data pendaftaran pada tabel ini. Berikut tabelnya.

**Tabel 4. 2.** Tabel Pendaftaran

<b>Tabel Pendaftaran</b>		
<b>No</b>	<b>Nama Field</b>	<b>Type</b>
<b>1</b>	<b>Id</b>	<b>Varchar(12)</b>
<b>2</b>	<b>Noktp</b>	<b>CharFiled(120)</b>
<b>3</b>	<b>Namasantri</b>	<b>CharFiled(120)</b>
<b>4</b>	<b>Alamat</b>	<b>CharFiled(120)</b>
<b>5</b>	<b>Tempat_lahir</b>	<b>CharFiled(120)</b>
<b>6</b>	<b>Tgl_lahir</b>	<b>CharFiled(120)</b>
<b>7</b>	<b>Jk</b>	<b>CharFiled(120)</b>
<b>8</b>	<b>Nohp</b>	<b>CharFiled(120)</b>
<b>9</b>	<b>Nama_ortu</b>	<b>CharFiled(120)</b>
<b>10</b>	<b>Bulan_pendaftaran</b>	<b>CharFiled(120)</b>

c. Tabel Santri

Tabel santri ini akan memberikan ruang penyimpanan data santri, sehingga pengurus dapat mengecek data-data santri pada tabel ini. Berikut tabelnya.

**Tabel 4. 3.** Tabel Santri

Tabel Santri		
No	Nama Field	Type
1	Id	Varchar(12)
2	Noktp	CharFiled(120)
3	Idsantri	CharFiled(120)
4	Namasantri	CharFiled(120)
5	Alamat	CharFiled(120)
6	Tempat_lahir	CharFiled(120)
7	Tgl_lahir	CharFiled(120)
8	Jk	CharFiled(120)
9	Pendidikan	CharFiled(120)
10	Nama_ortu	CharFiled(120)

d. Tabel Alumni

Tabel alumni ini akan memberikan ruang penyimpanan data lumni, sehingga pengurus dapat mengecek data-data lumni pada tabel ini. Berikut tabelnya.

**Tabel 4. 4.** Tabel Alumni

Alumni		
No	Nama Field	Type
1	Id	Varchar(12)
2	Noktp	CharFiled(120)
3	Idsantri	CharFiled(120)
4	Namasantri	CharFiled(120)
5	Alamat	CharFiled(120)

6	Tempat_lahir	CharFiled(120)
7	Tgl_lahir	CharFiled(120)
8	Jk	CharFiled(120)
9	Pendidikan	CharFiled(120)
10	Nama_ortu	CharFiled(120)

e. Tabel Pembayaran

Tabel pembayaran ini akan memberikan ruang penyimpanan data pembayaran, sehingga pengurus dapat mengecek data-data pembayaran pada tabel ini. Berikut tabelnya.

**Tabel 4.5.** Tabel Pembayaran

Tabel Pembayaran		
No	Nama Field	Type
1	Id	Varchar(12)
2	Nama_santri	CharFiled(120)
3	Alamat	CharFiled(120)
4	Kamar	CharFiled(120)
5	Gang	CharFiled(120)
6	Wilayah	CharFiled(120)
7	Nama_jenis	CharFiled(120)
8	Pembayaran	CharFiled(120)
9	Bayar	CharFiled(120)
10	Status	CharFiled(120)
11	Bulan_pembayaran	CharFiled(120)

f. Tabel Jenis Pembayaran

Tabel jenis pembayaran ini akan memberikan ruang penyimpanan data jenis pembayaran, sehingga pengurus dapat mengecek data-data jenis pembayaran pada tabel ini. Berikut tabelnya.

**Tabel 4. 6.** Tabel Jenis Pembayaran

<b>Tabel Jenis Pembayaran</b>		
<b>No</b>	<b>Nama Field</b>	<b>Type</b>
<b>1</b>	<b>Id</b>	<b>Varchar(12)</b>
<b>2</b>	<b>Nama_jenis</b>	<b>CharFiled(120)</b>
<b>3</b>	<b>Pembayaran</b>	<b>CharFiled(120)</b>
<b>4</b>	<b>Tanggal</b>	<b>CharFiled(120)</b>

g. Tabel Kamar

Tabel kamar ini akan memberikan ruang penyimpanan data kamar, sehingga pengurus dapat mengecek data-data kamar pada tabel ini. Berikut tabelnya.

**Tabel 4. 7.** Tabel Kamar

<b>Tabel Kamar</b>		
<b>No</b>	<b>Nama Field</b>	<b>Type</b>
<b>1</b>	<b>Id</b>	<b>Varchar(12)</b>
<b>2</b>	<b>Nama_kamar</b>	<b>CharFiled(120)</b>
<b>3</b>	<b>Gang</b>	<b>CharFiled(120)</b>
<b>4</b>	<b>Kapasistas</b>	<b>CharFiled(120)</b>

h. Tabel Wilayah

Tabel wilayah ini akan memberikan ruang penyimpanan data wilayah, sehingga pengurus dapat mengecek data-data wilayah pada tabel ini. Berikut tabelnya.

**Tabel 4. 8.** Tabel Wilayah

<b>Tabel Wilayah</b>		
<b>No</b>	<b>Nama Field</b>	<b>Type</b>
<b>1</b>	<b>Id</b>	<b>Varchar(12)</b>
<b>2</b>	<b>Nama_wilayah</b>	<b>CharFiled(120)</b>
<b>3</b>	<b>Keterangan</b>	<b>CharFiled(120)</b>

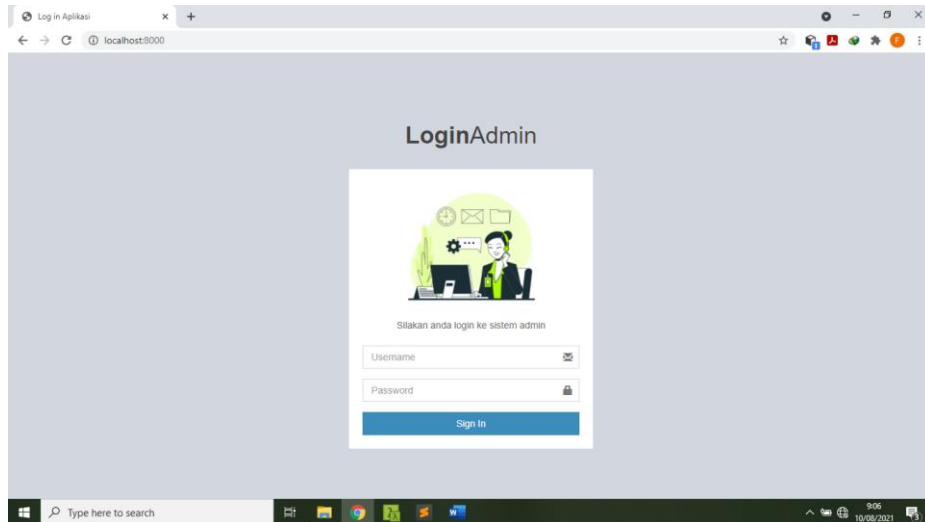
#### **4.2.5 Desain Input Dan Output**

Tahapan desain input dan output akan digunakan sebagai acuan dalam membuat sistem serta menghubungkan antara program dengan pengguna (user). Desain input digunakan sebagai interface antar pengguna untuk menginputkan data dan kemudian disimpan ke dalam tabel database. Salah satu desain yang termasuk ke dalam desain input yang ada dalam aplikasi administrasi dan pendaftaran pesantren online ini ialah halaman input login, halaman input petugas, halaman input pendaftaran dan halaman input administrasi. Desain output adalah hasil dari proses yang diinputkan oleh pengguna yang berisikan informasi administrasi dan pendaftaran pesantren. Aplikasi ini memiliki beberapa desain input dan output meliputi :

##### **1. Tampilan Halaman Login Petugas**

Pada tampilan login ini pengurus dapat memasukkan username dan password untuk dapat mengelola data pada sub menu utama, jika username dan password benar. Berikut gambaran dari login.

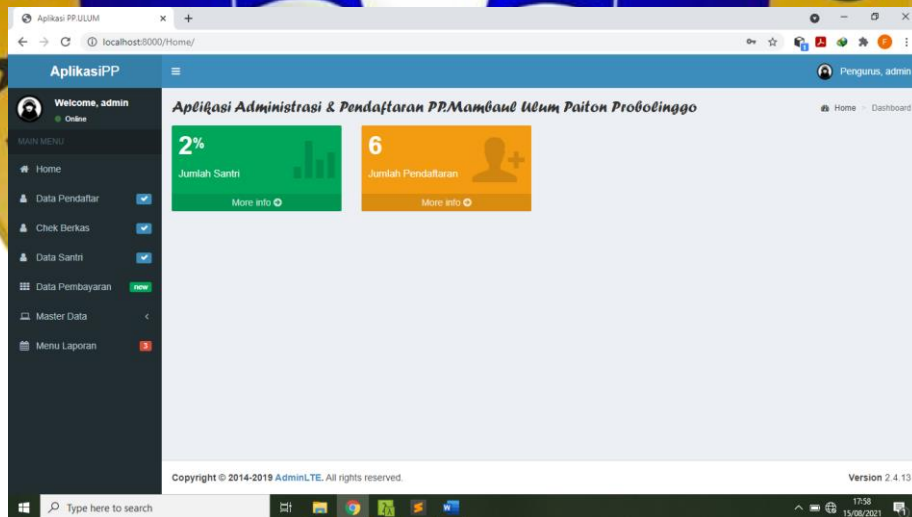




**Gambar 4. 12.** Tampilan Halaman login petugas

## 2. Tampilan Menu Utama

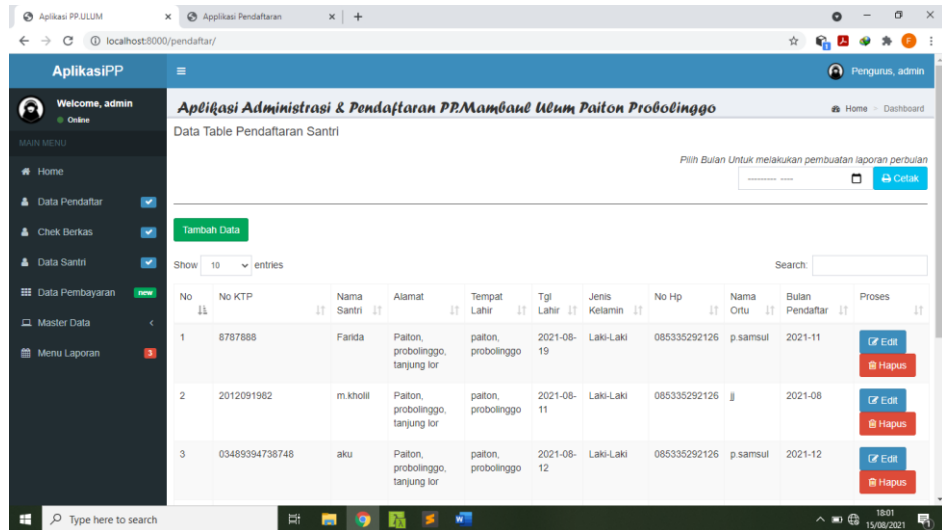
Ketika petugas sudah berhasil memasukkan username dan password dengan benar maka pengurus dapat mengelola data halaman menu utama. Berikut gambaran dari menu utama.



**Gambar 4. 13.** Tampilan Menu Utama

## 3. Tampilan Data Pendaftaran

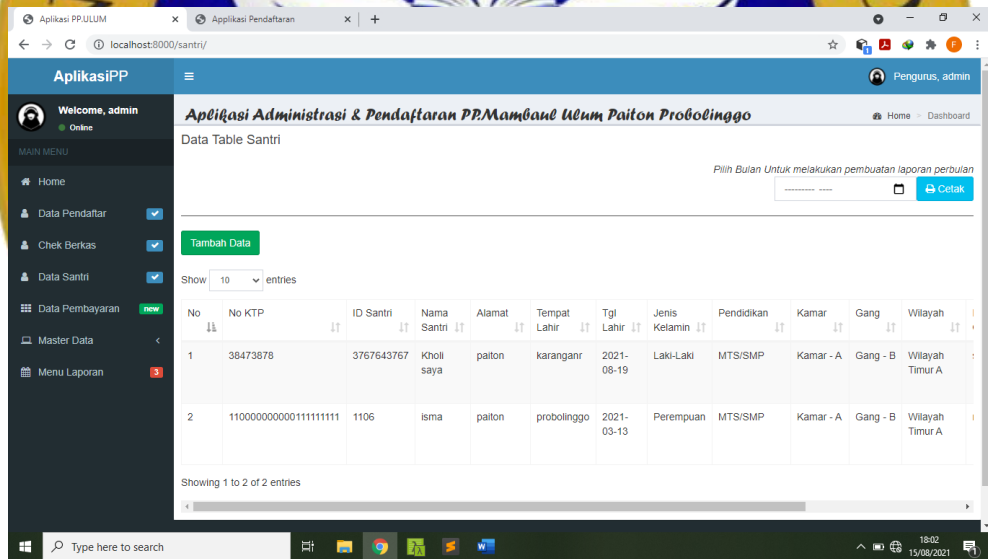
Pada tampilan ini, pengurus dapat melihat data-data pendaftaran yang sudah melakukan pendaftaran santri baru. Berikut gambarannya.



Gambar 4. 14. Tampilan Data Pendaftaran

#### 4. Tampilan Data Santri

Pada tampilan ini, pengurus dapat melihat data-data santri yang sudah dikelola oleh pengurus sebelumnya. Berikut gambarannya.



Gambar 4. 15. Tampilan Data Santri

#### 5. Tampilan Data Alumni

Pada tampilan ini, pengurus dapat melihat data-data alumni yang diinputkan oleh pengurus sebelumnya. Berikut gambarannya.

**AplikasiPP**  
 Aplikasi Administrasi & Pendaftaran PP Mambaul Ulum Paiton Probolinggo  
 Data Tabel Alumni Santri

Pilih Tahun Untuk melakukan pembuatan laporan pertahun  
 Pilih Tahun

Show 10 entries

No	NIK	ID Santri	Nama Santri	Alamat	Tempat Lahir	Tgl Lahir	Jenis Kelamin	Pendidikan	Kamar	Gang	Wilayah	Nama Ortu
1	3513115002050002	100205	Mia Febrianti	Desa Kotaanyar Kecamatan Kotaanyar	Probolinggo	2005-02-10	Perempuan	SMK	A2	Wilayah Abdullah	Wilayah Abdullah	Butro

Showing 1 to 1 of 1 entries

Copyright © 2014-2019 AdminLTE. All rights reserved. Version 2.4.13

**Gambar 4. 16.** Tampilan Data Alumni

#### 6. Tampilan Data Pembayaran

Pada tampilan ini, pengurus dapat melihat data-data pembayaran yang sudah melakukan pembayaran yang diinputkan oleh pengurus sebelumnya. Berikut gambarannya.

**AplikasiPP**  
 Aplikasi Administrasi & Pendaftaran PP Mambaul Ulum Paiton Probolinggo  
 Data Tabel Pembayaran Santri

Pilih Bulan untuk melakukan pembuatan laporan perbulan

Show 10 entries

No	Print	ID Santri	Nama Santri	Alamat	Kamar	Gang	Wilayah	Nama Jenis Pembayaran	Pembayaran	Bayar	Status	Tgl Pembayaran
1	<input type="button" value="Cetak"/>	3767643767	Kholi saya	paiton	Gang - B	Kamar - A	Wilayah Timur A	SPP 2	150000	50000	Dispen	2021-08-2

Showing 1 to 1 of 1 entries

Copyright © 2014-2019 AdminLTE. All rights reserved. Version 2.4.13

**Gambar 4. 17.** Tampilan Data Pembayaran

#### 7. Tampilan Data Jenis Pembayaran

Pada tampilan ini, pengurus dapat melihat data-data jenis pembayaran yang sudah dikelola oleh pengurus sebelumnya. Berikut gambarannya.

**ApikasiPP**  
Welcome, admin  
Online

**Apikasi Administrasi & Pendaftaran PP/Mambaul Ulem Paiton Probolinggo**

Data Table Jenis Pembayaran

Tambah Data

Show 10 entries

No	Nama Jenis	Pembayaran	Tanggal	Proses
1	Loker / Lemari	15000	2021-08-31	<a href="#">Edit</a> <a href="#">Hapus</a>
2	SPP	30000	2021-08-31	<a href="#">Edit</a> <a href="#">Hapus</a>
3	Ekstra	15000	2021-08-31	<a href="#">Edit</a> <a href="#">Hapus</a>
4	Hafatul Imithan	200000	2021-08-31	<a href="#">Edit</a> <a href="#">Hapus</a>
5	Kalender tahunan	25000	2021-08-31	<a href="#">Edit</a> <a href="#">Hapus</a>
6	Kantin (Liang Makan)	250000	2021-08-31	<a href="#">Edit</a> <a href="#">Hapus</a>

Showing 1 to 6 of 6 entries

Previous 1 Next

**Gambar 4. 18.** Tampilan Data Jenis Pembayaran

8. Tampilan Data kamar

Pada tampilan ini, pengurus dapat melihat data-data kamar yang sudah dikelola oleh pengurus sebelumnya. Berikut gambarannya.

**ApikasiPP**  
Welcome, admin  
Online

**Apikasi Administrasi & Pendaftaran PP/Mambaul Ulem Paiton Probolinggo**

Data Table Kamar

Tambah Data

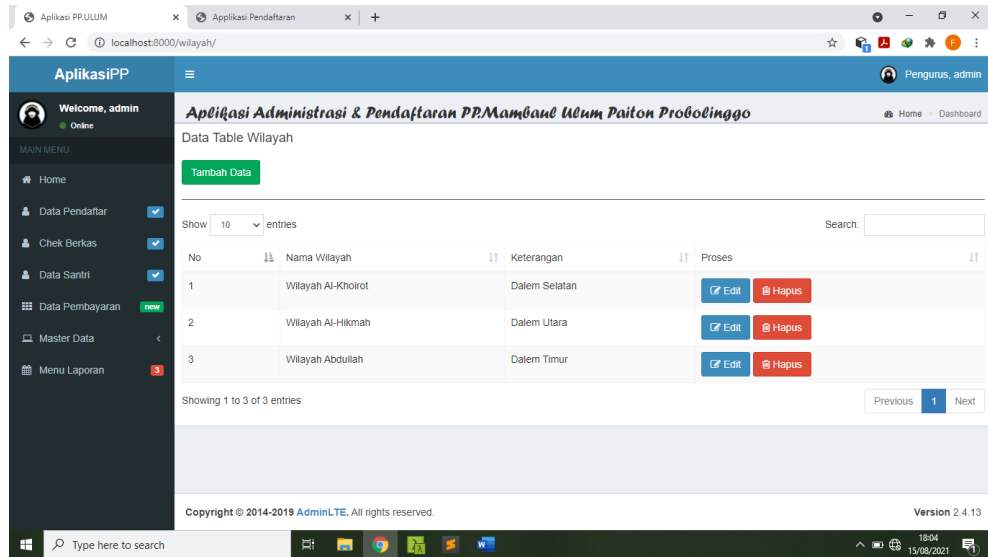
Show 10 entries

No	Nama Kamar	Gang	Kapasitas	Proses
1	Lainnya /Luar	lainnya /Luar	0	<a href="#">Edit</a> <a href="#">Hapus</a>
2	A1	Abu Hasan (Putra Al-Khoirot)	K 06 - G 04	<a href="#">Edit</a> <a href="#">Hapus</a>
3	B1	Abdullah (Putra Al-Khoirot)	K 06 - G 04	<a href="#">Edit</a> <a href="#">Hapus</a>
4	C1	Moh Khoiri (Putra Al-Khoirot)	K 05 - G 04	<a href="#">Edit</a> <a href="#">Hapus</a>
5	D1	Moh As'ad (Putra Al-Khoirot)	K 05 - G 04	<a href="#">Edit</a> <a href="#">Hapus</a>
6	Aula Dipojoyo	Aula Dipojoyo (Putra)	0	<a href="#">Edit</a> <a href="#">Hapus</a>
7	A2	Abu Hasan (Putra Al-Khoirot)	K 06 - G 04	<a href="#">Edit</a> <a href="#">Hapus</a>

**Gambar 4. 19.** Tampilan Data kamar

9. Tampilan Data Wilayah

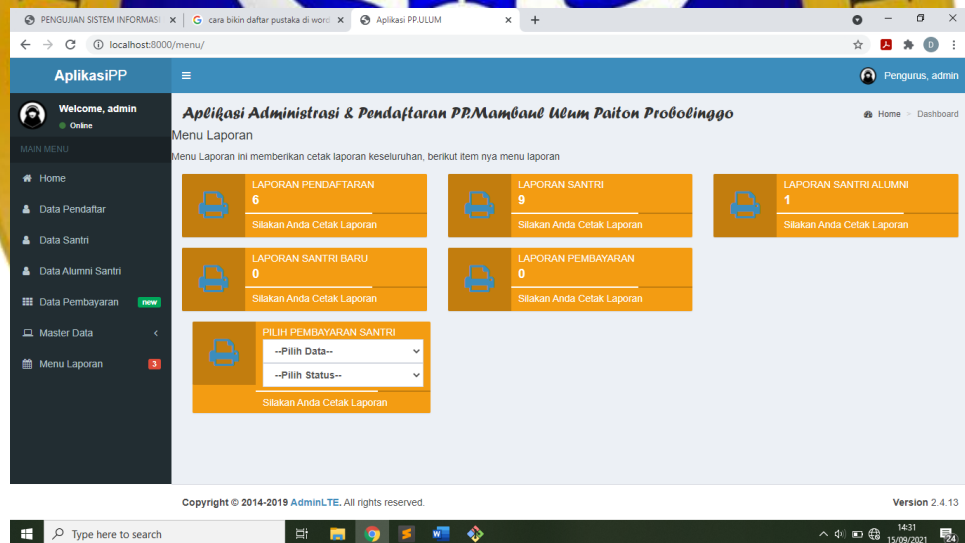
Pada tampilan ini, pengurus dapat melihat data-data wilayah yang sudah dikelola oleh pengurus sebelumnya. Berikut gambarannya.



**Gambar 4. 20.** Tampilan Data Wilayah

#### 10. Tampilan Menu Laporan

Pada tampilan ini, pengurus dapat melihat data-data wilayah yang sudah dikelola oleh pengurus sebelumnya. Berikut gambarannya.

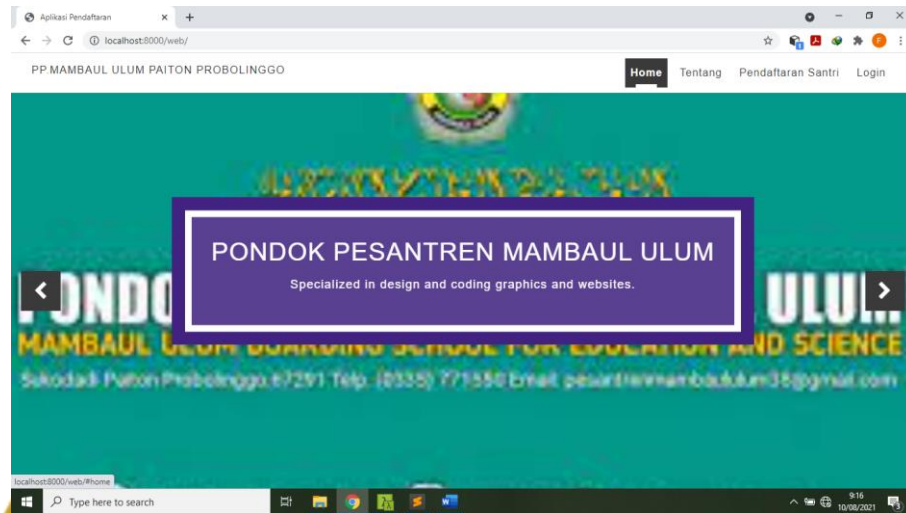


**Gambar 4. 21.** Tampilan Menu Laporan

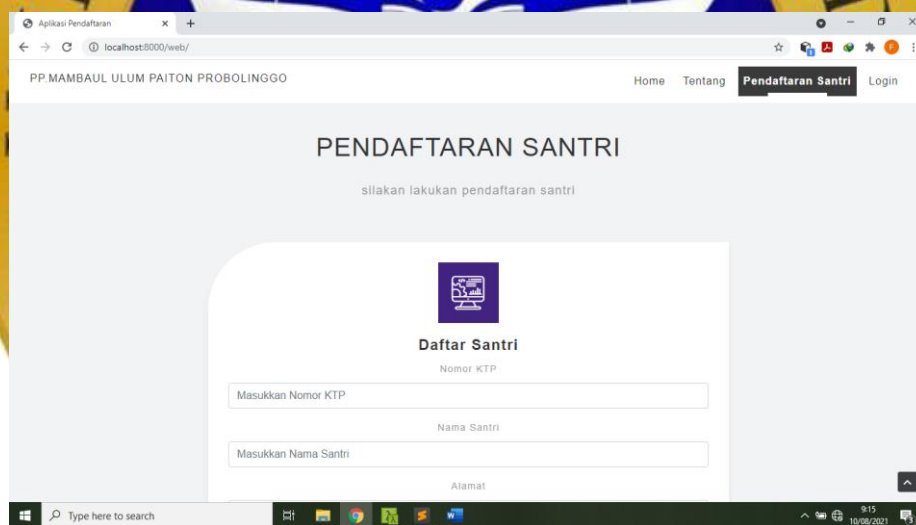
#### 11. Tampilan Halaman Website

Halaman website pendaftaran ini, yaitu dapat memberikan pelayanan pendaftaran santri secara online sehingga calon santri baru, dapat melakukan pendaftaran secara online melalui website ini. Website online memberikan

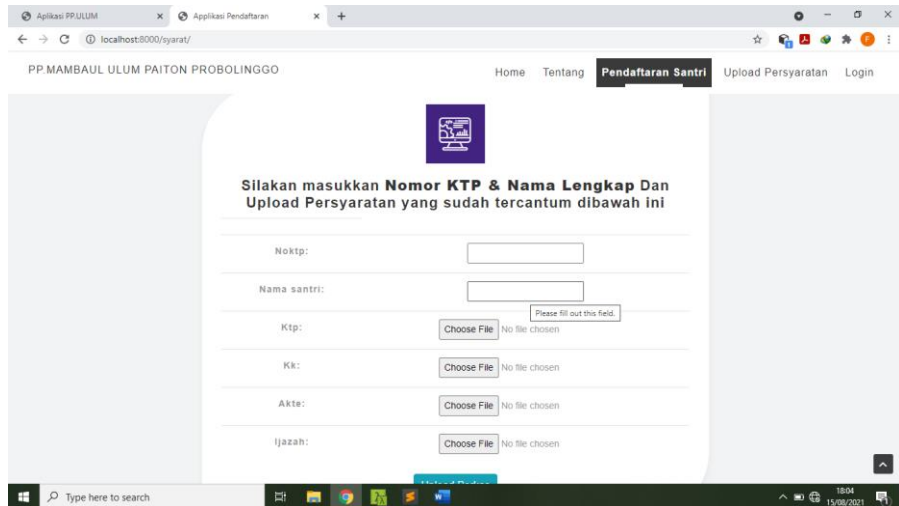
informasi dalam pembayaran santri sehingga santri dapat mengecek data pembayaran melalui website ini.



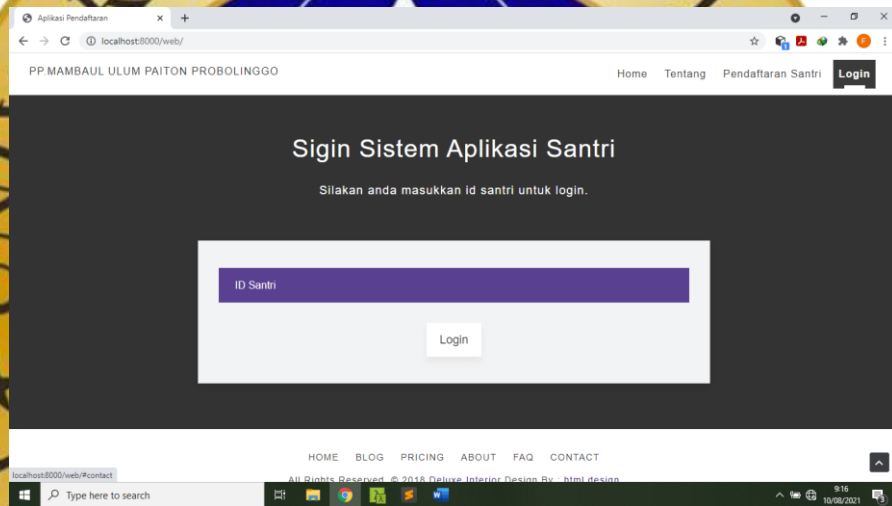
Gambar 4. 22. Tampilan Halaman Website



Gambar 4. 23. Tampilan Form Pendaftaran



Gambar 4. 24. Tampilan Upload Berkas Persyaratan



Gambar 4. 25. Tampilan Login santri

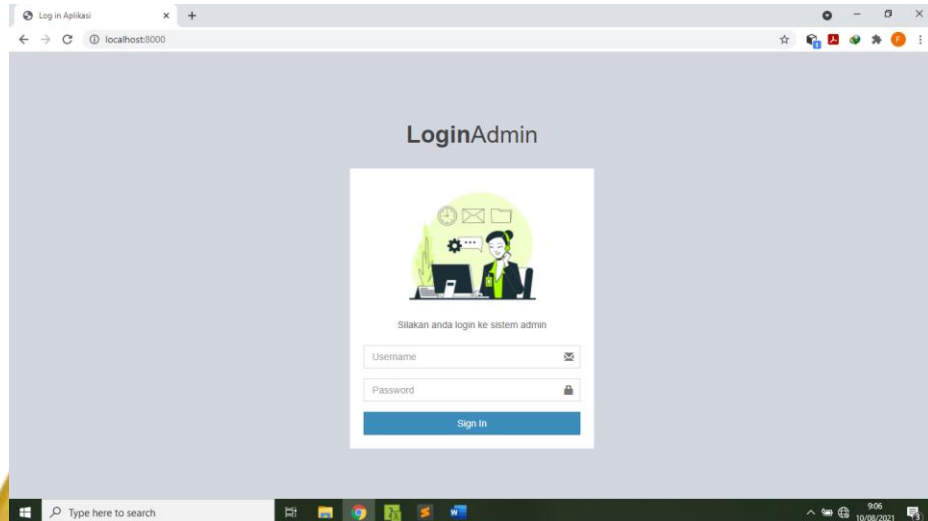
### 4.3 Pemograman (Coding)

Setelah berhasil mendesain aplikasi administrasi dan pendaftaran pondok pesantren, tahap selanjutnya adalah mengimplementasikan desain ke dalam kode program. Dibawah ini adalah implementasi dari Aplikasi Administrasi dan Pendaftaran Pesantren Berbasis Web Menggunakan *Framework Django* StudiKasus di PP. Mambaul Ulum meliputi.

#### 1. Halaman Login Petugas

Halaman login ini digunakan oleh petugas atau pengurus dengan memasukkan nama pengguna dan kata sandi. Halaman login ini digunakan

untuk masuk ke halaman utama sistem. Berikut halaman login pada aplikasi administrasi dan pendaftaran terlihat pada gambar 4.26 Halaman Login Petugas.



Gambar 4. 26. Halaman Login Petugas

```
Kode : def index(request):
    context = {
        'page_title':'Login',
    }
    #print(request.user)
    if request.method == "POST":
        username = request.POST['username']
        password = request.POST['password']

        user = authenticate(request, username=username, password=password)
        if user is not None:
            login(request, user)
            return redirect('Home')
        else:
            return redirect('index')

    return render(request, 'index.html', context)
```

#### Segmen Program 4. 1. Halaman Login Petugas

#### 2. Tampilan Menu Utama

Setelah petugas login, pengguna akan diarahkan ke halaman menu utama sebagai hak akses petugas. Pada halaman ini mengelola menu-menu diantaranya, data pendaftar mengelola data pendaftar, data santri mengelola data santri, data alumni mengelola data alumni, data pembayaran mengelola



data pembayaran, data jenis pembayaran mengelola data jenis pembayaran, data kamar mengelola data kamar, data wilayah mengelola data wilayah, menu laporan mengelola menu laporan pendaftaran, laporan santri baru, laporan santri, laporan alumni, dan laporan pembayaran yang sudah dibuat. Berikut Gambar 4.27 Halaman Menu Utama



Gambar 4. 27. Halaman Menu Utama

```
Kode : def HomeView(request):
    jml_pendaftaran = ModelPendaftar21.objects.count()
    jml_santri = ModelSantri12.objects.count()
    context = {
        'page_title':'Home',
        'jml_pendaftaran': jml_pendaftaran,
        'jml_santri': jml_santri
    }
    return render(request, 'Home.html', context)
```

```
def LogoutView(request):
    context = {
        'page_title':'logout',
    }
    if request.method == "POST":
        if request.POST["logout"] == "Submit":
            logout(request)

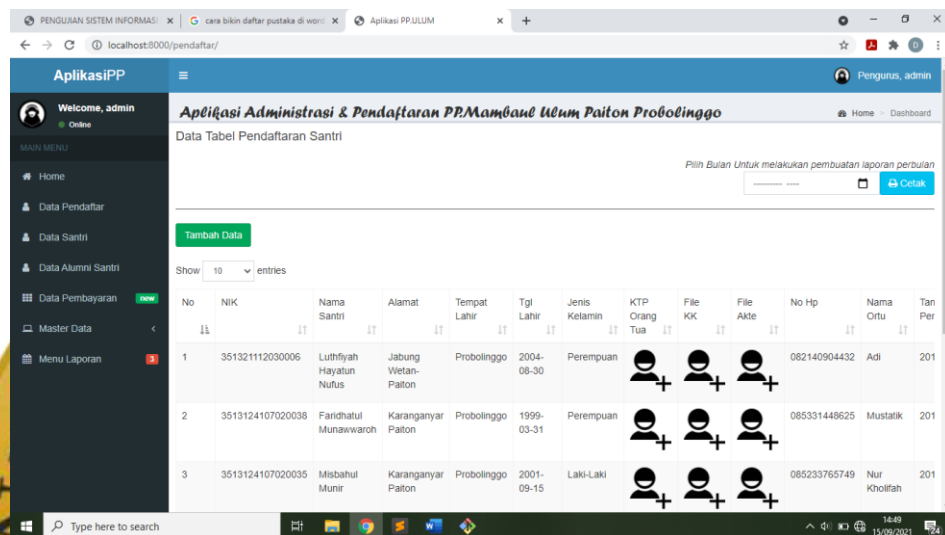
    return redirect('index')

return render(request, 'logout.html', context)
```

#### Segmen Program 4. 2. Halaman Menu Utama

### 3. Tampilan Data Pendaftar

Data pendaftar ini dikelola oleh petugas. Pada halaman data pendaftar terdapat beberapa fitur meliputi, tambah data pendaftar, cetak laporan pendaftar perbulan, dan edit data. Berikut Gambar 4.28 Tampilan Data Pendaftar



No	NIK	Nama Santri	Alamat	Tempat Lahir	Tgl Lahir	Jenis Kelamin	KTP Orang Tua	File KK	File AKTE	No Hp	Nama Ortu	Tan Per
1	35132112030006	Luthfiah Hayatun Nufus	Jabung Wetan-Paiton	Probolinggo	2004-08-30	Perempuan				082140904432	Adi	201
2	3513124107020038	Faridhatul Munawwaroh	Karanganyar Paiton	Probolinggo	1999-03-31	Perempuan				085331448625	Mustalik	201
3	3513124107020035	Misbahul Munir	Karanganyar Paiton	Probolinggo	2001-09-15	Laki-Laki				085233765749	Nur Khoifah	201

Gambar 4.28. Tampilan Data Pendaftar

```
Kode : def PendaftarV(request):
    data_pendaftar = ModelPendaftar21.objects.all()
    context = {
        'data_pendaftar': data_pendaftar,
    }
    return render(request, 'Master/data_pendaftar/tabel.html', context)

def T_PendaftarV(request):
    if 'noktp' in request.GET:
        c_santri=request.GET['noktp']
        post_santri1 = ModelPendaftar21.objects.filter(noktp=c_santri)
    else:
        post_santri1 = ModelPendaftar21.objects.filter(noktp=None)

    if request.method == 'POST':
        ModelPendaftar21.objects.create(
            noktp = request.POST['noktp'],
            nama_santri = request.POST['nama_santri'],
            alamat = request.POST['alamat'],
            tempat_lahir = request.POST['tempat_lahir'],
            tgl_lahir = request.POST['tgl_lahir'],
            jk = request.POST['jk'],
            KTP_Ortu= request.FILES['upload'],
```

```

        KK= request.FILES['kk'],
        AKTE= request.FILES['akta'],
        nohp = request.POST['nohp'],
        nama_ortu = request.POST['nama_ortu'],
        tanggal_pendaftaran = request.POST['tanggal_d'],
        bulan = request.POST['bulan'],
    )
    messages.info(request, 'Data Berhasil Di Simpan..?')
    return HttpResponseRedirect("/pendaftar/")
context = {
    'page': 'page',
    'post_santri1': post_santri1,
}
return render(request, 'Master/data_pendaftar/input.html', context)

def HapusPendaftarV(request, hapusp_id):
    ModelPendaftar21.objects.filter(id=hapusp_id).delete()
    return redirect('pendaftar')

def EditPendaftarV(request, id_p):
    data = ModelPendaftar21.objects.get(id=id_p)
    if request.method == 'POST':
        data.noktp = request.POST.get('noktp')
        data.nama_santri = request.POST.get('nama_santri')

        data.alamat = request.POST.get('alamat')

        data.tempat_lahir = request.POST.get('tempat_lahir')

        data.tgl_lahir = request.POST.get('tgl_lahir')

        data.jk = request.POST.get('jk')
        data.KTP_Ortu = request.FILES['upload']
        data.KK = request.FILES['kk']
        data.AKTE = request.FILES['akta']
        data.nohp = request.POST.get('nohp')

        data.nama_ortu = request.POST.get('nama_ortu')

        data.tanggal_pendaftaran = request.POST.get('tanggal_d')

        data.bulan = request.POST.get('bulan')

        data.save()
        return redirect('pendaftar')

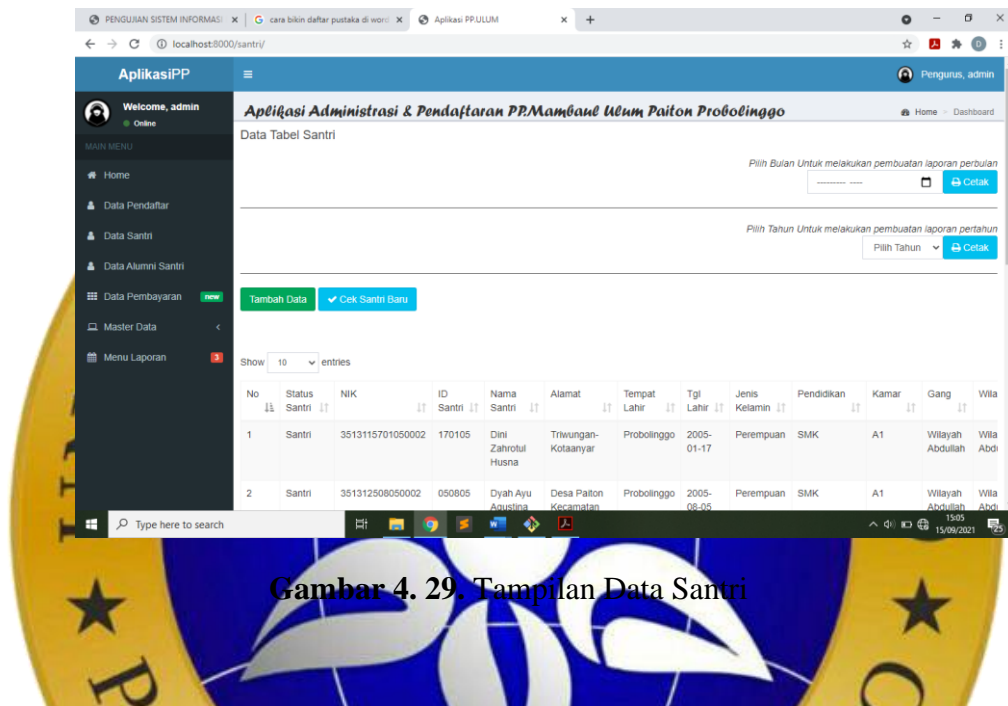
context = {'data': data}
return render(request, 'Master/data_pendaftar/edit.html', context)

```

### Segmen Program 4. 3. Tampilan Data Pendaftar

#### 4. Tampilan Data Santri

Data santri ini dikelola oleh petugas. Pada halaman data santri terdapat beberapa fitur meliputi, tambah data santri, data santri baru, cetak laporan santri perbulan, cetak laporan santri pertahun dan edit data. Berikut Gambar 4.29 Tampilan Data Santri



Gambar 4. 29. Tampilan Data Santri

```
Kode : def SantriV(request):
    data_santri = ModelSantri12.objects.filter(status_santri='Santri')
    context = {
        'data_santri': data_santri,
    }
    return render(request, 'Master/data_santri/tabel.html', context)
```

```
def SantriBaruV(request):

    data_santri1 = ModelSantri12.objects.filter(status_santri='Santri Baru')
    context = {
        'data_santri1': data_santri1,
    }
    return render(request, 'Master/data_santri/tabel_baru.html', context)
```

```
def T_santriV(request):
    select_kamar = ModelKamar.objects.all()
    select_wilayah = Modelwilayah.objects.all()
    # cek santri perkamar
    K1=ModelSantri12.objects.filter(kamar = 'A1').count()
```

```

# cek setiap kamar
A1=ModelKamar.objects.filter(nama_kamar = 'A1')
A2=ModelKamar.objects.filter(nama_kamar = 'A2')
A3=ModelKamar.objects.filter(nama_kamar = 'A3')
A4=ModelKamar.objects.filter(nama_kamar = 'A4')
A5=ModelKamar.objects.filter(nama_kamar = 'A5')
A6=ModelKamar.objects.filter(nama_kamar = 'A6')
if 'cari' in request.GET:
    c_santri=request.GET['cari']
    post_santri = ModelPendaftar21.objects.filter(noktp=c_santri)
else:
    post_santri = ModelPendaftar21.objects.filter(noktp=None)

if request.method == 'POST':
    ModelSantri12.objects.create(
        noktp = request.POST['noktp'],
        idsantri = request.POST['idsantri'],
        nama_santri = request.POST['nama_santri'],
        alamat = request.POST['alamat'],
        tempat_lahir = request.POST['tempat_lahir'],
        tgl_lahir = request.POST['tgl_lahir'],
        jk = request.POST['jk'],
        pendidikan = request.POST['pendidikan'],
        nama_ortu = request.POST['nama_ortu'],
        kamar = request.POST['kamar'],
        gang = request.POST['gang'],
        wilayah = request.POST['wilayah'],
        status_santri = request.POST['status'],
        bulan = request.POST['bulan'],
        tahun = request.POST['tahun'],
    )
    c_d=request.POST['noktp']
    ModelPendaftar21.objects.filter(noktp=c_d).delete()
    messages.info(request, 'Data Berhasil Di Simpan..?')
    return HttpResponseRedirect("/santri/")

context = {
    'page': 'page',
    'select_kamar': select_kamar,
    'select_wilayah': select_wilayah,
    'post_santri': post_santri,
    'K1': K1,
    'A1': A1,
    'A22': A22,
    'A3': A3,
    'A4': A4,
    'A5': A5,
    'A6': A6
}

return render(request, 'Master/data_santri/input.html', context)
def HapusSantriV(request, hapuss_id):
    ModelSantri12.objects.filter(id=hapuss_id).delete()
    return redirect('santri')

```

```

def EditSantriV(request, id_s):
    A1=ModelKamar.objects.filter(nama_kamar = 'A1')
    A2=ModelKamar.objects.filter(nama_kamar = 'A2')
    A3=ModelKamar.objects.filter(nama_kamar = 'A3')
    A4=ModelKamar.objects.filter(nama_kamar = 'A4')
    A5=ModelKamar.objects.filter(nama_kamar = 'A5')
    A6=ModelKamar.objects.filter(nama_kamar = 'A6')
    data = ModelSantri12.objects.get(id=id_s)
    if request.method == 'POST':
        data.noktp = request.POST.get('noktp')
        data.idsantri = request.POST.get('idsantri')
        data.nama_santri = request.POST.get('nama_santri')

        data.alamat = request.POST.get('alamat')

        data.tempat_lahir = request.POST.get('tempat_lahir')

        data.tgl_lahir = request.POST.get('tgl_lahir')

        data.jk = request.POST.get('jk')
        data.pendidikan = request.POST.get('pendidikan')

        data.nama_ortu = request.POST.get('nama_ortu')

        data.kamar = request.POST.get('kamar')

        data.gang = request.POST.get('gang')

        data.wilayah = request.POST.get('wilayah')

        data.status_santri = request.POST.get('status')

        data.bulan = request.POST.get('bulan')

        data.tahun = request.POST.get('tahun')

        data.save()
        return redirect('santri')

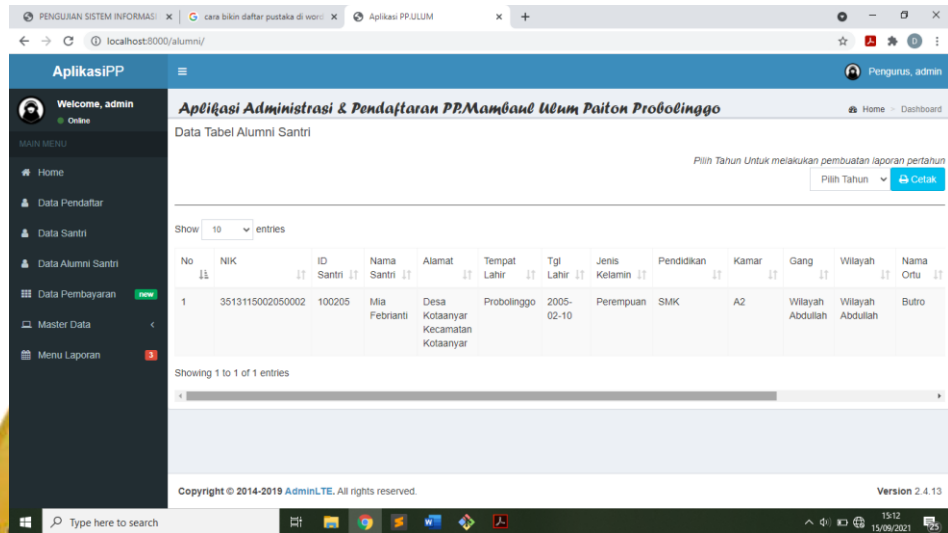
    context = {'data': data,
               'A1': A1,
               'A2': A2,
               'A3': A3,
               'A4': A4,
               'A5': A5,
               'A6': A6
              }
    return render(request, 'Master/data_santri/edit.html', context)

```

#### Segmen Program 4. 4. Tampilan Data Santri

## 5. Tampilan Data Alumni

Data alumni ini dikelola oleh petugas. Pada halaman data alumni terdapat beberapa fitur meliputi, tambah data alumni, cetak laporan santri pertahun dan edit data. Berikut Gambar 4.30 Tampilan Data Alumni



No	NIK	ID Santri	Nama Santri	Alamat	Tempat Lahir	Tgl Lahir	Jenis Kelamin	Pendidikan	Kamar	Gang	Wilayah	Nama Ortu
1	3513115002050002	100205	Mia Febrianti	Desa Kotaanyar Kecamatan Kotaanyar	Probolinggo	2005-02-10	Perempuan	SMK	A2	Wilayah Abdullah	Wilayah Abdullah	Butro

**Gambar 4. 30.** Tampilan Data Alumni

```
Kode def LP_Alumni_SantriTahun(request):
    if 'cari_tahun' in request.GET:
        cetak=request.GET['cari_tahun']
        lp_santri = ModelSantri12.objects.filter(tahun=cetak)
    else:
        lp_santri = ModelSantri12.objects.filter(tahun=None)

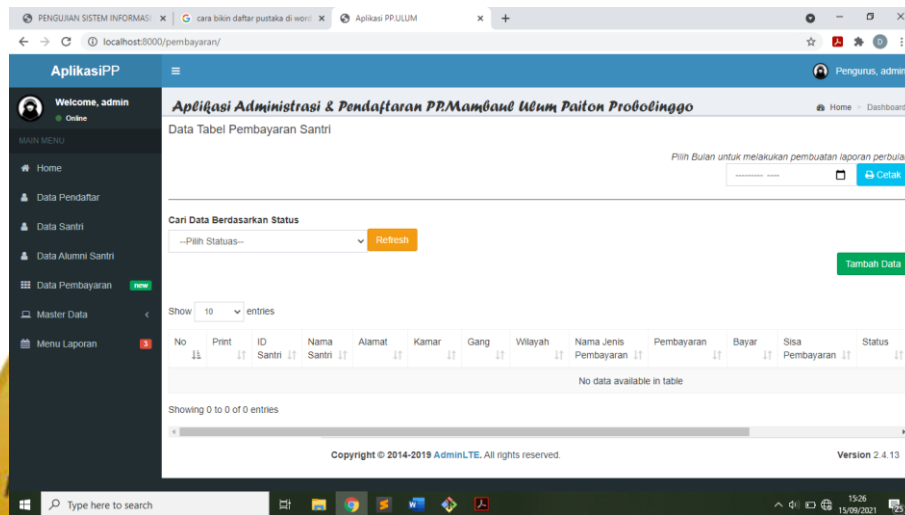
    if 'cari_lumni' in request.GET:
        cetak=request.GET['cari_lumni']
        lp_santri = ModelSantri12.objects.filter(status_santri=cetak)
    else:
        lp_santri = ModelSantri12.objects.filter(status_santri=None)

    context = {
        'lp_santri': lp_santri,
    }
    return render(request, 'Master/data_alumni/c_tahun.html', context)
```

### **Segmen Program 4. 5.** Tampilan Data Alumni

## 6. Tampilan Data Pembayaran

Data pembayaran ini dikelola oleh petugas. Pada halaman data pembayaran terdapat beberapa fitur meliputi, tambah data pembayaran, cetak laporan pembayaran perbulan dan pilih status. Berikut Gambar 4.31 Tampilan Data Pembayaran



Gambar 4. 31. Tampilan Data Pembayaran

```
Kode : def PembayaranV(request):
    # pem = Modelpembayaran12.objects.all()
    if 'c_status' in request.GET:
        c_status=request.GET['c_status']
        data_pembayaran =
        Modelpembayaran12.objects.filter(status=c_status)
    else:
        data_pembayaran = Modelpembayaran12.objects.all()
    context = {
        'data_pembayaran': data_pembayaran,
    }
    return render(request, 'Master/data_pembayaran/tabel.html', context)

def T_PembayaranV(request):
    select_jenis = ModeljenisP.objects.all()
    if 'chek' in request.GET:
        c_santri=request.GET['chek']
        post_santri = ModelSantri12.objects.filter(idsantri=c_santri)
    else:
        post_santri = ModelSantri12.objects.filter(idsantri=None)

    if 'cari_jenis' in request.GET:
        cari_jenis=request.GET['cari_jenis']
        post_jenis = ModeljenisP.objects.filter(nama_jenis=cari_jenis)
```



```

else:
    post_jenis = ModeljenisP.objects.filter(nama_jenis=None)

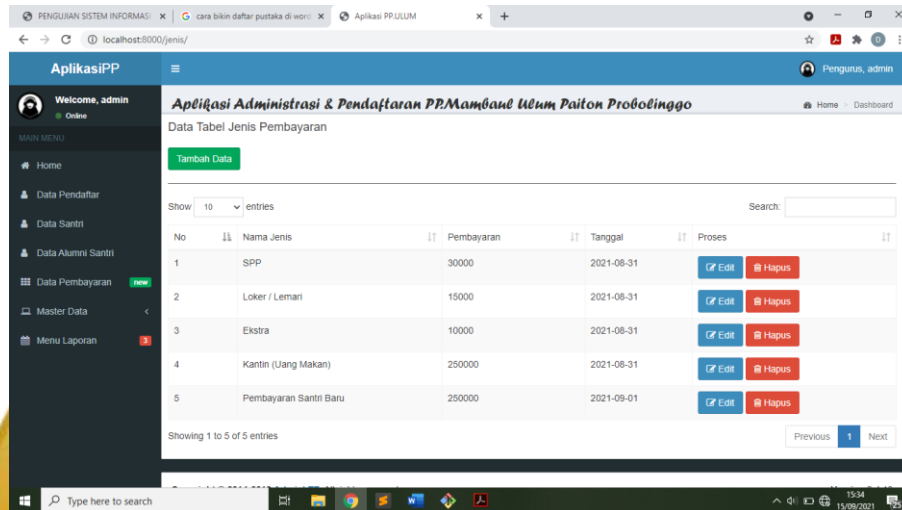
if request.method == 'POST':
    Modelpembayaran12.objects.create(
        idsantri = request.POST['idsantri'],
        nama_santri = request.POST['nama_santri'],
        alamat = request.POST['alamat'],
        kamar = request.POST['kamar'],
        gang = request.POST['gang'],
        wilayah = request.POST['wilayah'],
        status_santri = request.POST['status_santri'],
        nama_jenis = request.POST['nama_jenis'],
        pembayaran = request.POST['pembayaran'],
        bayar = request.POST['bayar'],
        kembalian = request.POST['kembalian'],
        status = request.POST['status'],
        tanggal = request.POST['tanggal'],
        bulan = request.POST['bulan'],
    )
    #messages.info(request, 'Data Berhasil Di Simpan..?')
    return HttpResponseRedirect("/pembayaran/")
context = {
    'post_santri': post_santri,
    'post_jenis': post_jenis,
    'select_jenis': select_jenis
}
return render(request, 'Master/data_pembayaran/input.html', context)
def HapuspembayaranV(request, hapuspem_id):
    Modelpembayaran12.objects.filter(id=hapuspem_id).delete()
    return redirect('pembayaran')
def EditPembayaranV(request, id_pem):
    data = Modelpembayaran12.objects.get(id=id_pem)
    if request.method == 'POST':
        data.pembayaran = request.POST.get('pembayaran')
        data.status = request.POST.get('status')
        data.tanggal = request.POST.get('tanggal')
        data.bulan = request.POST.get('bulan')
        data.save()
        return redirect('pembayaran')
context = {'data': data,}
return render(request, 'Master/data_pembayaran/edit.html', context)

```

#### Segmen Program 4. 6. Tampilan Data Pembayaran

## 7. Tampilan Data Jenis Pembayaran

Tampilan ini dikelola oleh petugas. Pada halaman ini terdapat beberapa fitur meliputi, tambah, edit dan hapus data jenis pembayaran. Berikut Gambar 4.32 Tampilan Data Jenis Pembayaran



No	Nama Jenis	Pembayaran	Tanggal	Proses
1	SPP	30000	2021-08-31	<a href="#">Edit</a> <a href="#">Hapus</a>
2	Loker / Lemari	15000	2021-08-31	<a href="#">Edit</a> <a href="#">Hapus</a>
3	Ekstra	10000	2021-08-31	<a href="#">Edit</a> <a href="#">Hapus</a>
4	Kantin (Uang Makan)	250000	2021-08-31	<a href="#">Edit</a> <a href="#">Hapus</a>
5	Pembayaran Santan Baru	250000	2021-09-01	<a href="#">Edit</a> <a href="#">Hapus</a>

Gambar 4. 32. Tampilan Data Jenis Pembayaran

```
Kode : def JenisV(request):
    data_jenis = ModeljenisP.objects.all()
    context = {
        'data_jenis': data_jenis,
    }
    return render(request, 'Master/data_jenis/tabel.html', context)

def T_JenisV(request):
    if request.method == 'POST':
        ModeljenisP.objects.create(
            nama_jenis = request.POST['nama_jenis'],
            pembayaran = request.POST['pembayaran'],
            tanggal = request.POST['tanggal'],
        )
        messages.info(request, 'Data Berhasil Di Simpan..?')
        return HttpResponseRedirect("/jenis/")
    context = {
        'page': 'page'
    }
    return render(request, 'Master/data_jenis/input.html', context)

def HapusJenisV(request, hapusj_id):
    ModeljenisP.objects.filter(id=hapusj_id).delete()
    return redirect('jenis')
```

```

def EditJenisV(request, id_j):
    data = ModeljenisP.objects.get(id=id_j)
    if request.method == 'POST':
        data.nama_jenis = request.POST.get('nama_jenis')
        data.pembayaran = request.POST.get('pembayaran')

        data.tanggal = request.POST.get('tanggal')

    data.save()
    return redirect('jenis')

context = {'data': data}
return render(request, 'Master/data_jenis/edit.html', context)

```

#### Segmen Program 4. 7. Tampilan Data Jenis Pembayaran

#### 8. Tampilan Data Kamar

Data kamar ini dikelola oleh petugas. Pada halaman data kamar terdapat beberapa fitur meliputi, tambah data kamar. Gambar 4.33 Tampilan Data Pembayaran

No	Nama Kamar	Gang	Kapasitas Kamar	Proses
1	A1	Wilayah Abdullah	10	[Edit] [Hapus]
2	A2	Wilayah Abdullah	10	[Edit] [Hapus]
3	A3	Wilayah Abdullah	10	[Edit] [Hapus]
4	A4	Wilayah Abdullah	10	[Edit] [Hapus]
5	A5	Wilayah Abdullah	10	[Edit] [Hapus]
6	A6	Wilayah Abdullah	10	[Edit] [Hapus]
7	A5	Abu Hasan (Putra Al-Khoirot)	10	[Edit] [Hapus]

Gambar 4. 33. Tampilan Data Kamar

```

Kode : #kamar
def KamarV(request):
    data_kamar = ModelKamar.objects.all()
    context = {
        'data_kamar': data_kamar,
    }
    return render(request, 'Master/data_kamar/tabel.html', context)
def T_KamarV(request):
    if request.method == 'POST':
        ModelKamar.objects.create(
            nama_kamar = request.POST['nama_kamar'],
            gang = request.POST['gang'],
            kapasistas = request.POST['kapasitas'],
        )
        return HttpResponseRedirect("/kamar/")
    context = {
        'Tambah Kamar':'tambah'
    }
    return render(request, 'Master/data_kamar/input.html', context)

def HapusKamarV(request, hapusk_id):
    ModelKamar.objects.filter(id=hapusk_id).delete()
    return redirect('kamar')

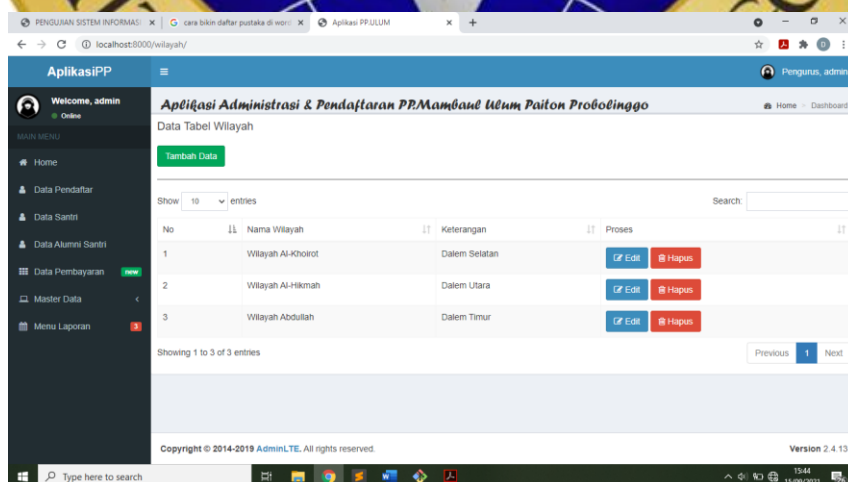
def EditKamarV(request, id_k):
    edit_kamar = ModelKamar.objects.get(id=id_k)

```

### Segmen Program 4. 8. Tampilan Data Kamar

#### 9. Tampilan Data Wilayah

Data wilayah ini dikelola oleh petugas. Pada halaman data wilayah terdapat beberapa fitur meliputi, tambah data wilayah. Berikut Gambar 4.34 Tampilan Data Wilayah



Gambar 4. 34. Tampilan Data Wilayah

```

Kode : def WilayahV(request):
    data_wilayah = Modelwilayah.objects.all()
    context = {
        'data_wilayah': data_wilayah,
    }
    return render(request, 'Master/data_wilayah/tabel.html', context)

def T_WilayahV(request):
    if request.method == 'POST':
        Modelwilayah.objects.create(
            nama_wilayah = request.POST['nama_wilayah'],
            keterangan = request.POST['keterangan'],
        )
        messages.info(request, 'Data Berhasil Di Simpan..?')
        return HttpResponseRedirect("/wilayah/")
    context = {
        'wialayah': 'wilayah'
    }
    return render(request, 'Master/data_wilayah/input.html', context)

def HapusWilayahV(request, hapusw_id):
    Modelwilayah.objects.filter(id=hapusw_id).delete()
    return redirect('wilayah')

def EditWilayahV(request, id_w):
    data = Modelwilayah.objects.get(id=id_w)
    if request.method == 'POST':
        data.nama_wilayah = request.POST.get('nama_wilayah')
        data.keterangan = request.POST.get('keterangan')
        data.save()
        return redirect('wilayah')

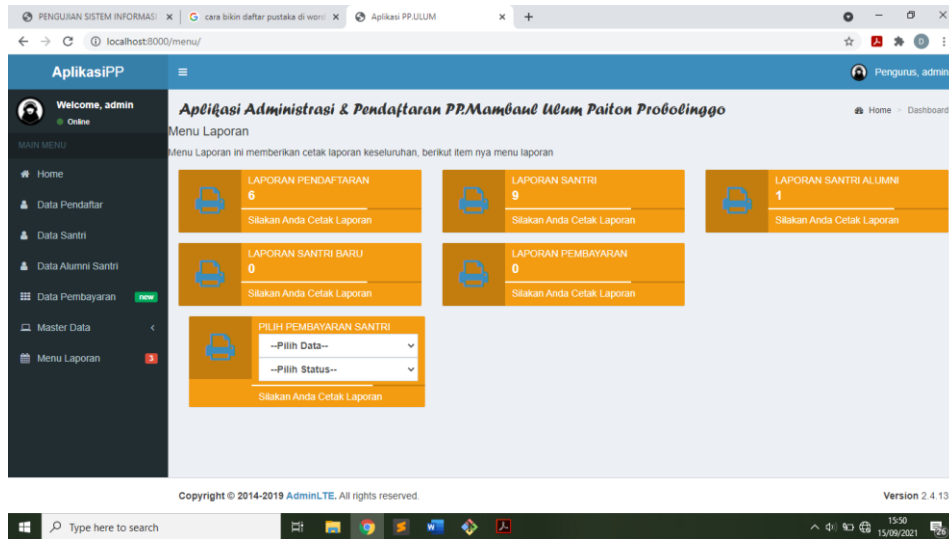
    context = {'data': data}
    return render(request, 'Master/data_wilayah/edit.html', context)

```

#### Segmen Program 4. 9. Tampilan Data Wilayah

##### 10. Tampilan Menu Laporan

Pada halaman ini menampilkan beberapa jenis laporan yang dapat di cetak dalam bentuk pdf, kemudian petugas dapat menyimpan hasil cetak pdf dan dapat mencetak langsung hasil pdf. Berikut Gambar 4.35 Halaman Menu Utama



Gambar 4. 35. Tampilan Menu Laporan

```

Kode : def Menu_laporan(request):
    jml_pendaftaran = ModelPendaftar21.objects.count()
    jml_santri = ModelSantri12.objects.filter(status_santri = 'Santri').count()
    jml_pembayaran = Modelpembayaran12.objects.count()
    jml_alumni = ModelSantri12.objects.filter(status_santri =
'Alumni').count()
    jml_baru = ModelSantri12.objects.filter(status_santri = 'Santri
Baru').count()
    context = {
        'page_title':'Home',
        'jml_pendaftaran': jml_pendaftaran,
        'jml_santri': jml_santri,
        'jml_pembayaran': jml_pembayaran,
        'jml_baru': jml_baru,
        'jml_alumni': jml_alumni
    }
    return render(request, 'Master/laporan/menu.html', context)
def lp_pendaftaran(request):
    lp_pendaftar = ModelPendaftar21.objects.all()
    context = {
        'lp_pendaftar': lp_pendaftar,
    }
    return render(request, 'Master/laporan/lp_pendaftar.html', context)

def lp_santri(request):
    lp_santri = ModelSantri12.objects.filter(status_santri='Santri')
    context = {
        'lp_santri': lp_santri,
    }
    return render(request, 'Master/laporan/lp_santri.html', context)
def lp_santri_alumni(request):

```

```

lp_santri = ModelSantri12.objects.filter(status_santri='Alumni')
context = {
    'lp_santri': lp_santri,
}
return render(request, 'Master/laporan/lp_santri_alumni.html', context)
def lp_santri_baru(request):
    lp_santri = ModelSantri12.objects.filter(status_santri='Santri Baru')
    context = {
        'lp_santri': lp_santri,
    }
    return render(request, 'Master/laporan/lp_santri_baru.html', context)
def lp_pembayaran(request):
    lp_pembayaran = Modelpembayaran12.objects.all()
    context = {
        'lp_pembayaran': lp_pembayaran,
    }
    return render(request, 'Master/laporan/lp_pembayaran.html', context)

```

**Segmen Program 4. 10. Tampilan Menu Laporan**

**4.4 Pengujian**

1. Hasil Pengujian Internal

Pengujian internal merupakan pengujian program berdasarkan fungsi-fungsi dari program. Tujuan dari pengujian internal untuk mengetahui kelemahan software sebelum pengujian eksternal, yang akan mengikutsertakan pihak-pihak yang menjadi target observasi yaitu pondok pesantren Mambaul Ulum Paiton. Berikut Tabel 4.6 Hasil Pengujian Internal dalam aplikasi administrasi dan pendaftaran santri baru.

**Tabel 4. 9.** Hasil Pengujian Internal

No	Item Pengujian	Skenario Pengujian	Hasil Yang Di Harapkan	Sesuai	
				Ya	Tidak
1.	Website Pesantren Mambaul Ulum Paiton	Halaman Website Pesantren Mambaul Ulum Paiton	Pada halaman ini akan menampilkan website pesantren Mambaul	✓	

			<p>UlumPaiton dengan beberapa menu, diantaranya :</p> <ol style="list-style-type: none"> <li>1. Halaman utama</li> <li>2. Tentang pesantren</li> <li>3. Pendaftaran santri</li> <li>4. Upload persyaratan</li> <li>5. Login santri</li> </ol>		
2.	Form Pendaftaran Santri Baru	Halaman Pendaftaran	<p>Pada halaman ini akan menampilkan inputan pendaftaran santri baru. Ketika sudah selesai otomatis akan ke halaman upload berkas persyaratan pendaftaran santri baru.</p>	✓	
3.	Form Upload Berkas Persyaratan	Halaman Upload Berkas Persyaratan	<p>Pada halaman ini akan menampilkan</p>	✓	



	Santri Baru	Santri Baru	inputan upload berkas pendaftaran santri baru.		
4.	Form Login Pengurus	Memilih Login	Pada sistem aplikasi ini dapat memberikan hak akses dan login use name dan password.	✓	
5.	Tampil Menu Utama Pengurus	Tampil Menu Pengurus	Pada sistem aplikasi apabila pengurus sudah memasukkan username dan password secara benar maka pengurus dapat mengelola data-data pada menu. Berikut menu yang diharapkan.	✓	
			<ol style="list-style-type: none"> <li>1. Data santri</li> <li>2. Data pendaftaran</li> <li>3. Data pembayaran</li> <li>4. Data jenis</li> </ol>		

			<p>pembayaran</p> <p>5. Data Chek Berkas Persyaratan</p> <p>6. Data kamar</p> <p>7. Data wilayah</p> <p>8. Menu laporan</p>		
6.	Menu Laporan	Memilih Menu Laporan	<p>Pada menu laporan petugas dapat mencetak laporan keseluruhan yang ada pada menu laporan yang sudah disediakan oleh sistem. Berikut menu laporan.</p> <p>4. Laporan Data santri</p> <p>5. Laporan Data pendaftaran</p> <p>6. Laporan Data pembayaran</p>		

7.	Form Login Santri	Memilih Login	Pada sistem aplikasi ini dapat memberikan hak akses dan login hanya menggunakan id santri.	✓	
8.	Tampil Menu Santri	Tampil Menu Santri	Pada halaman ini akan menampilkan tabel data pembayaran santri.	✓	

Berdasarkan hasil pengujian yang telah dilakukan di atas, dapat ditarik kesimpulan bahwa aplikasi administrasi dan pendaftaran santri baru telah memenuhi kebutuhan sistem yang diinginkan, namun kesalahan dari aplikasi masih dapat terjadi, sehingga memerlukan pemeliharaan aplikasi.

## 2. Hasil Pengujian Eksternal

Pengujian eksternal adalah pengujian yang dilakukan secara langsung kepada pengguna yakni petugs atau pengurus dengan mengisi kuesioner, dan kemudian juga akan diujikan kepada sekelompok pengguna dengan mengambil 20 sampel. Tabel Hasil Uji Eksternal ditunjukkan di bawah ini.

**Tabel 4. 10.** Pengujian Eksternal

No	Draf Pertanyaan	Jawaban				
		Sangat Baik	Baik	Cukup Baik	Tidak Baik	Sangat Tidak Baik
1	Apakah dengan	4	1	0	0	0

	Aplikasi ini petugas dan santri lebih mudah dalam proses administrasi dan pendaftaran santri baru?					
2	Apakah petugas dan santri mudah dalam mengoperasikan aplikasi ini ?	2	2	1	0	0
3	Apakah fitur-fitur sistem ini mudah dioperasikan?	4	1	0	0	0
4	Apakah aplikasi ini sudah layak digunakan ?	3	2	0	0	0
5	Apakah ada kendala dalam mengoperasikan aplikasi ini ?	1	2	2	0	0
Total		14	8	3	0	0

Berdasarkan pengumpulan hasil responden diatas, maka dapat ditentukan skor sebagai berikut:

- a. Bobot setiap kala
  - Sangat Baik = 5
  - Baik = 4

Cukup Baik = 3

Tidak Baik = 2

Sangat Tidak Baik = 1

b. Nilai Interfal

$I = 100 / \text{jumlah angket}$

$I = 100 / 5$

$I = 20$

Jumlah skor perolehan dari semua responden :

Jumlah Sangat Baik =  $14 \times 5 = 70$

Jumlah Baik =  $8 \times 4 = 32$

Jumlah Cukup Baik =  $3 \times 3 = 9$

Jumlah Tidak Baik =  $0 \times 2 = 0$

Jumlah Sangat Tidak Baik (STS) =  $0 \times 1 = 0$

Total Skor =  $70 + 32 + 9 + 0 + 0 = 111$

Jumlah nilai ideal diperoleh dari nilai tertinggi x jumlah butir soal x jumlah responden =  $5 \times 5 \times 5 = 125$

Rumus Penentuan Skor =  $(\text{Skor Perolehan} / \text{Skor Ideal}) \times 100$ . Perhitungan  $(111 / 125) \times 100 = 88$  hasil perhitungan menggunakan metode skala likert didapatkan data sebanyak 88 % yang artinya Aplikasi Administrasi dan Pendaftaran Pesantren Online Berbasis *Web* Menggunakan *Framework Django* Terintegrasi dengan di Pondok Pesantren Mambaul Ulum Paifon ini “setuju” atau layak untuk digunakan.

#### 4.5 Perawatan

Dalam hal ini, akan dijelaskan langkah langkah perawatan perangkat lunak aplikasi administrasi dan pendaftaran pesantren, antara lain:

##### 1. Perawatan Berjangka

Proses administrasi dan pendaftaran pesantren pasti akan berjalan setiap hari, oleh karena itu pemeliharaan sistem harus dilakukan setahun sekali atau sebulan sekali untuk memastikan sistem beroperasi dengan baik.

## 2. *Update* Sitem

Aspek paling penting dari pemeliharaan sistem adalah *update* sistem. Tujuan *update* sistem untuk memperbarui data yang ada serta data yang masih ingin dimasukkan, seperti nama pengguna dan kata sandi. Hal ini bertujuan agar mencegah hacker dunia maya mendapatkan akses ke sistem.

