

## BAB IV

### Hasil dan Pembahasan

#### 4.1 Hasil Pengumpulan Data

Sebelum melakukan uji coba, langkah pertama yang dilakukan ialah melakukan pengumpulan data, pengumpulan data yang dibutuhkan untuk perancangan dan pengembangan sistem, dengan cara melakukan kegiatan observasi dan wawancara kepada pihak terkait agar ditemukannya beberapa kebutuhan yang diinginkan.

##### 4.1.1 Hasil observasi

Berdasarkan observasi yang dilakukan di Desa Sumberrejo pada tanggal 17 maret 2021 menghasilkan bahwa, dalam proses pelayanan surat oleh aparat desa sudah terkomputerisasi menggunakan *Microsoft Word* namun dirasa belum cukup maksimal, dikarenakan petugas harus memasukkan data berupa Kartu Tanda Penduduk (KTP) dan Kartu Keluarga (KK) dalam bentuk fisik oleh masyarakat kedalam *Microsoft Word*. Selain itu, masyarakat diwajibkan datang untuk menyelesaikan surat-surat tersebut. Jika aparat desa tidak ada di kantor desa maka masyarakat tidak akan mendapatkan proses pelayanan. Dalam proses pembuatan surat, aparat desa hanya membuat dan mencetak tanpa adanya laporan akhir dalam bentuk data.

##### 4.1.2 Hasil wawancara

Adapun hasil wawancara yang dilakukan pada tanggal 17 maret 2021 dengan salah satu aparat desa Sumberrejo, menghasilkan bahwa desa Sumberrejo memiliki 4.626 penduduk. Ada beberapa pelayanan kepada masyarakat Desa Sumberrejo meliputi Surat Keterangan Usaha, Surat Keterangan Domisili, Surat Keterangan Tidak Mampu, Surat Keterangan Kehilangan, Surat Keterangan Kematian, Surat Keterangan Terdampak Covid, Surat Keterangan Kelakuan Baik, dan Surat Keterangan Penghasilan. Dalam pengajuan pembuatan surat, masyarakat diwajibkan datang dan membawa *fotocopy* Kartu Tanda penduduk (KTP) dan Kartu Keluarga (KK). Kemudian petugas menerima berkas yang dibawa oleh masyarakat lalu diproses untuk pembuatan surat sesuai dengan surat yang dibutuhkan oleh masyarakat. Dalam proses pelayanan surat oleh aparat desa di Desa Sumberrejo sudah

terkomputerisasi menggunakan *Microsoft Word*, namun kurang efektif dan efisien, dikarenakan dalam proses penginputan data masyarakat sering kali terjadi kesalahan seperti, terjadi kesalahan dalam penulisan NIK, nama tanggal lahir, dan lain lain. Selain itu, dalam proses pembuatan surat, aparat desa hanya membuat dan mencetak tanpa adanya laporan akhir dalam bentuk data.

## **4.2 Hasil Analisis dan Desain**

### **4.2.1 Analisis Sistem Lama**

Dalam pengajuan pembuatan surat, masyarakat diwajibkan datang dan membawa *fotocopy* Kartu Tanda penduduk (KTP) dan Kartu Keluarga (KK). Kemudian aparat desa menerima berkas yang dibawa oleh masyarakat lalu diproses untuk pembuatan surat sesuai dengan surat yang dibutuhkan oleh masyarakat. Dalam proses pembuatan surat, aparat desa perlu menginputkan data masyarakat kedalam *Microsoft Word*, dalam proses penginputan sering kali terjadi kesalahan dalam penginputan data masyarakat seperti, terjadi kesalahan dalam penulisan NIK, nama dan lain lain. Kemudian dalam proses pembuatan surat, aparat desa hanya mencetak surat yang diperlukan oleh masyarakat tanpa membuat laporan akhir.

### **4.2.2 Analisis Sistem Baru**

Aplikasi ini mampu melakukan proses pelayanan surat secara efektif dan efisien. Aparat tidak perlu menginputkan satu persatu data masyarakat, melainkan aparat hanya perlu menginputkan Nomor Induk Masyarakat (NIK) kemudian data masyarakat akan secara otomatis tampil pada kolom inputan sesuai dengan kebutuhan pembuatan surat. Aplikasi ini juga dilengkapi dengan laporan pembuatan surat.

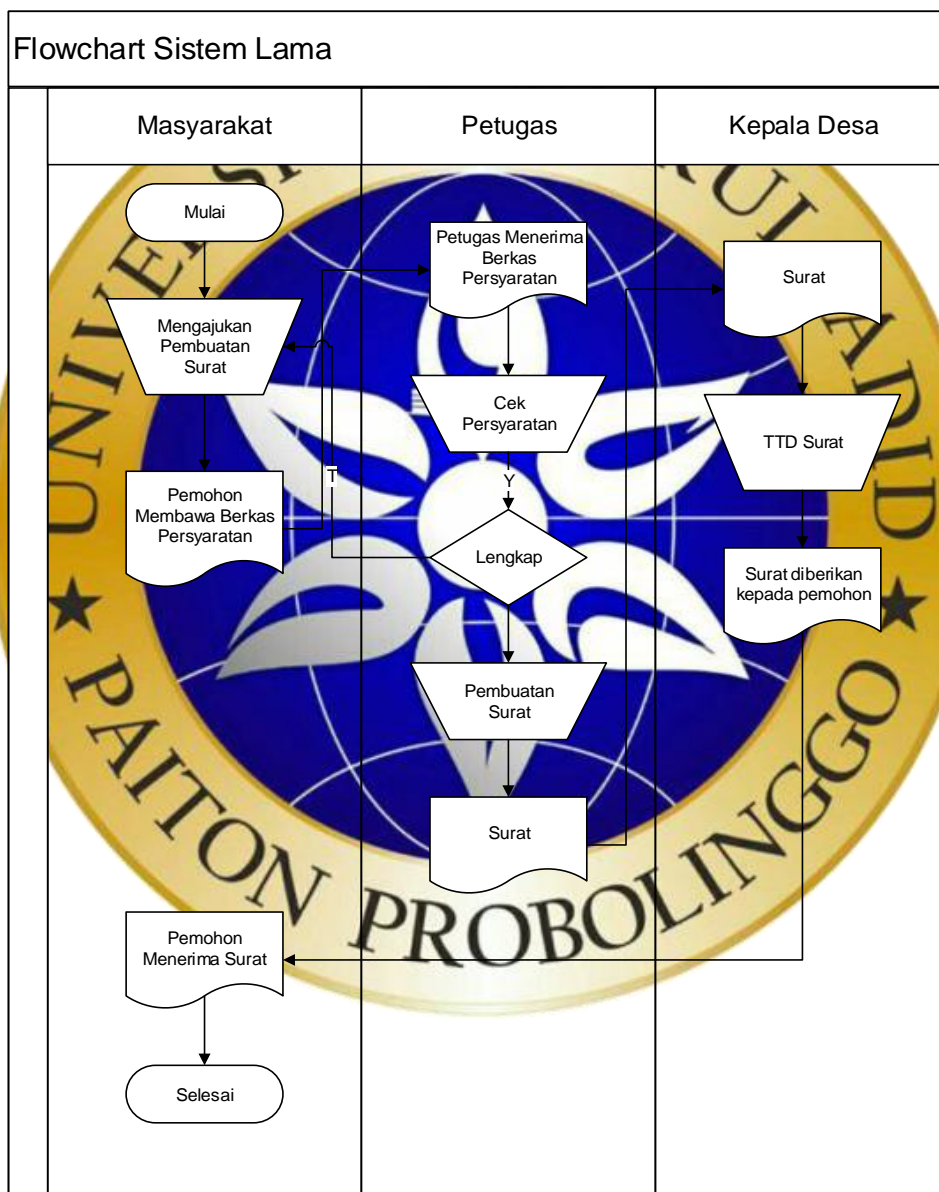
### **4.2.3 Desain Sistem**

Pada penelitian ini desain sistem dilakukan dengan perancangan bagan alur (*flowchart*), *Data Flow Diagram* (DFD), Entity Relationship Diagram (ERD), serta dilengkapi dengan desain database dan desain program agar lebih mudah di pahami oleh pengguna.

1. Bagan Alur Sistem (*flowchart*)

*Flowchart* adalah bagan alir yang menggambarkan alur sistem untuk menuntaskan permasalahan yang terjadi dan untuk memperoleh gambaran yang lebih jelas mengenai sistem ini. Berikut merupakan gambar *flowchart* seperti gambar dibawah ini.

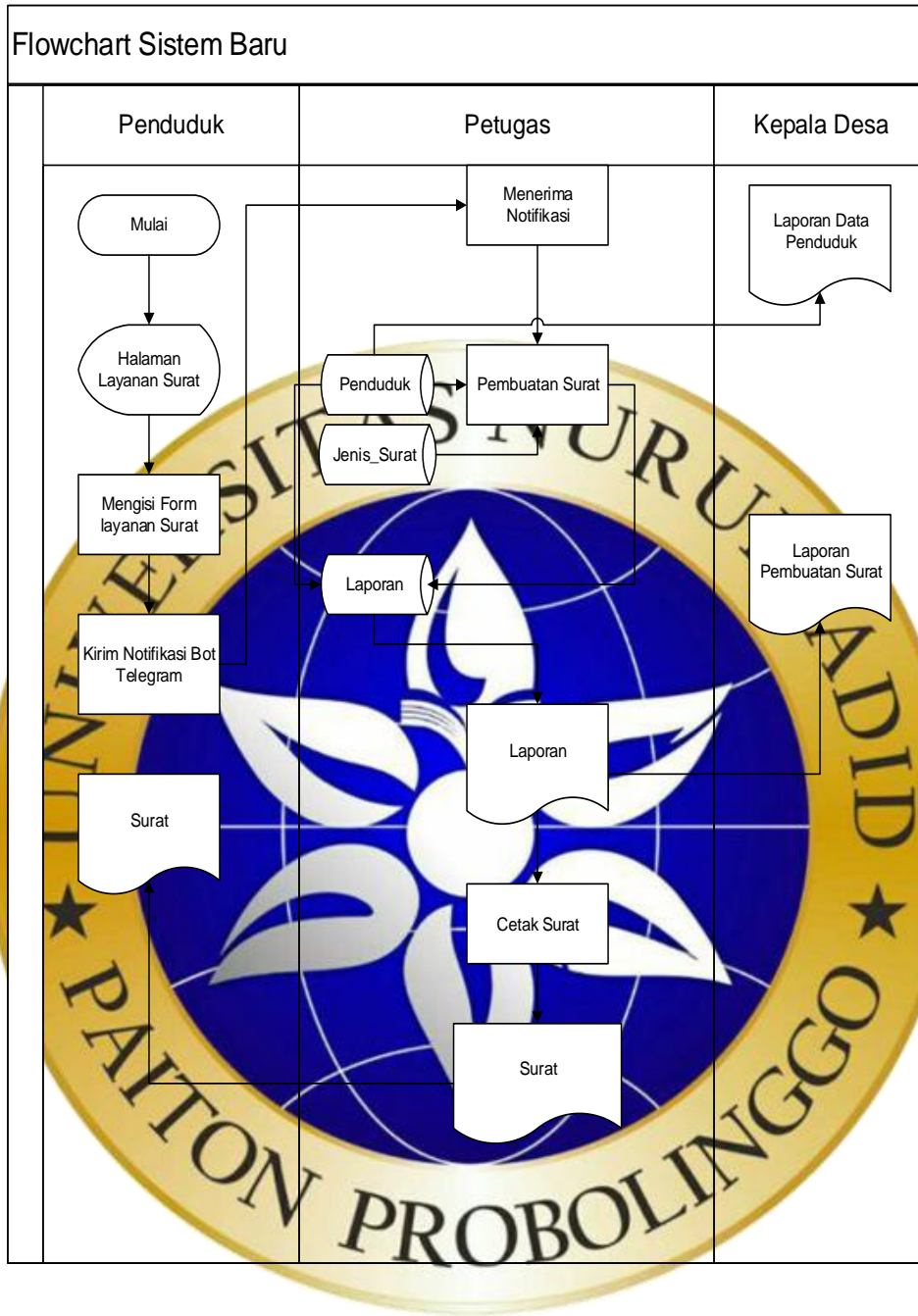
a. *Flowchart* sistem lama



**Gambar 4. 1.** *flowchart* sistem lama



b. *Flowchart* sistem baru

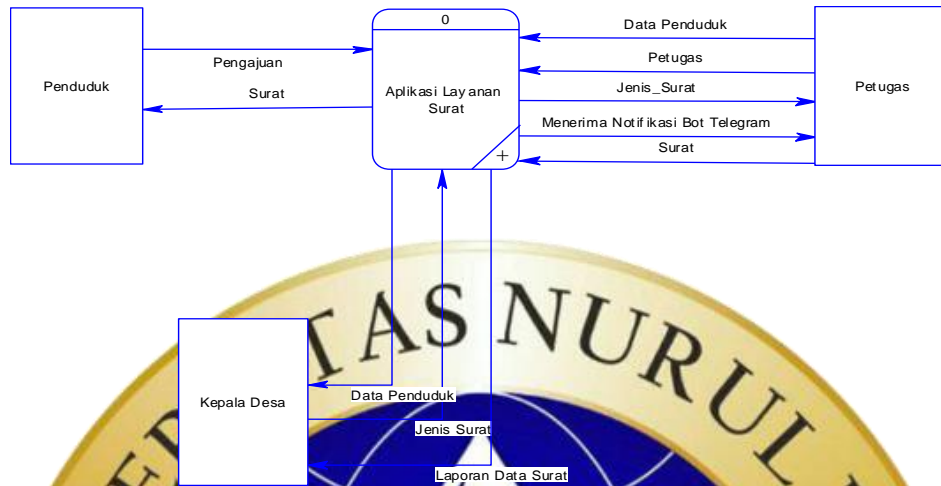


**Gambar 4. 2.** *flowchart* sistem baru

2. *Context* Diagram

*Context* diagram merupakan diagram tingkat atas dari DFD yang menjelaskan hubungan antara entitas luar, masukan dan keluaran suatu sistem. Dalam *context* diagram hanya mempunyai satu proses saja yang menunjukkan tentang

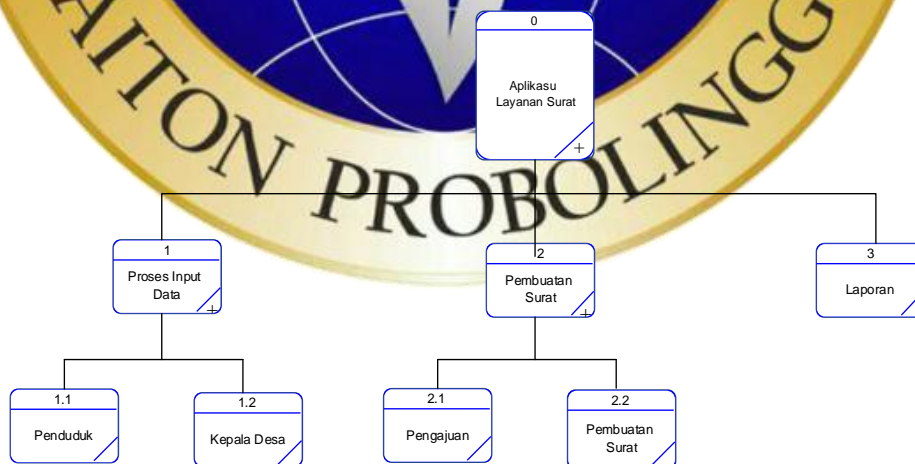
keseluruhan sistem. Context diagram untuk Aplikasi Layanan Surat Di Desa Sumberrejo adalah sebagai berikut :



**Gambar 4. 3. Context Diagram**

### 3. Bagan Berjenjang

Bagan berjenjang merupakan suatu bagan yang menunjukkan seluruh proses yang berlangsung dalam sistem dan digunakan untuk mempermudah dalam pembuatan suatu sistem. Bagan berjenjang memudahkan gambaran DFD ke level-level yang lebih bawah. Gambar bagan berjenjang dapat dilihat pada Gambar 4.4 sebagai berikut :



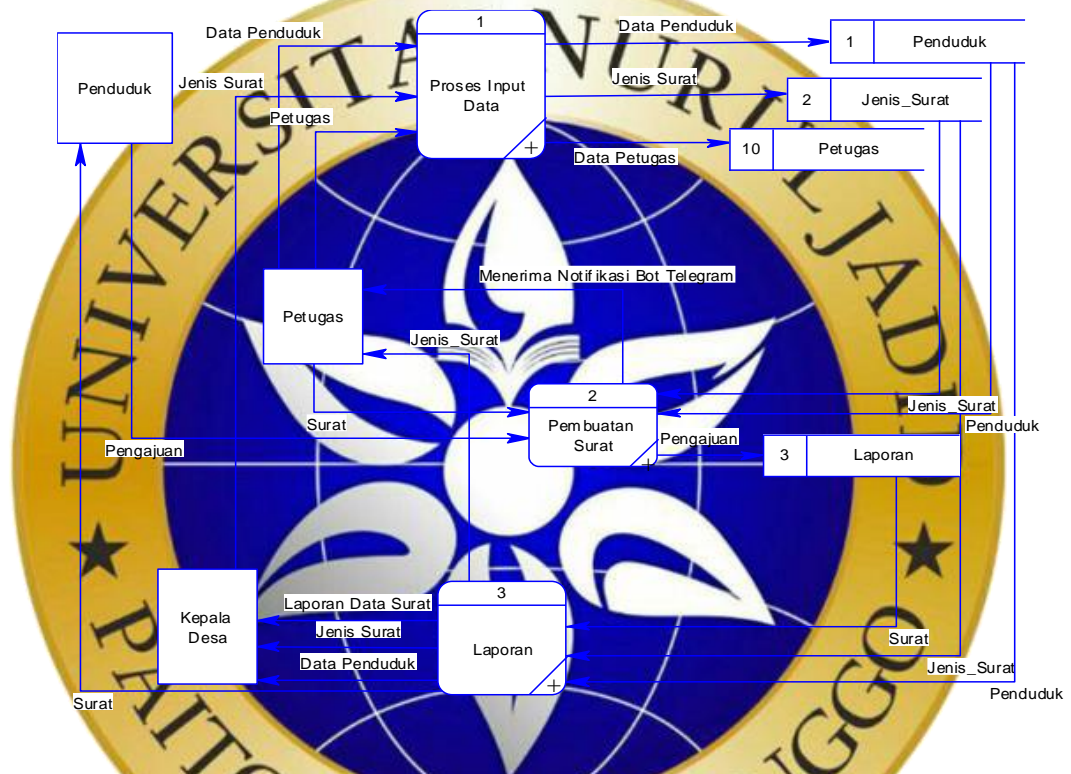
**Gambar 4. 4. Diagram Berjenjang**

### 4. Data Flow Diagram (DFD)

*Data Flow Diagram* merupakan suatu diagram alur data yang memberikan gambaran tentang aliran data dari sistem yang akan dirancang untuk mendapatkan informasi yang dibutuhkan. Dalam *Data Flow Diagram* berhubungan dengan arus data dan informasi suatu sistem yang menggambarkan siapa saja yang terlibat dengan sistem tersebut dari awal sampai akhir.

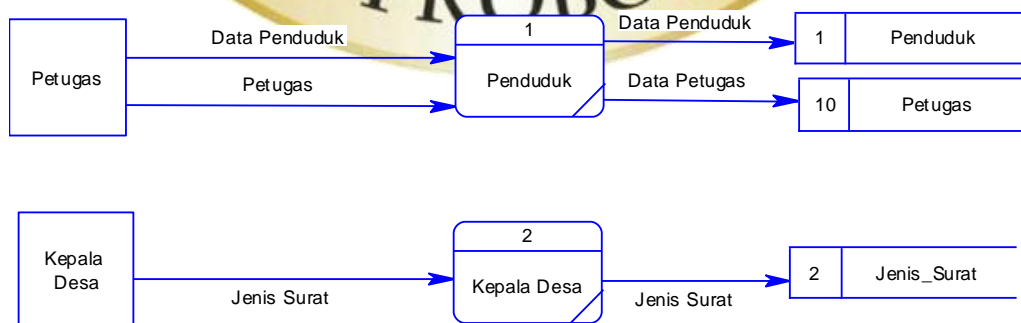
1) DFD level 1

DFD level 1 akan menjelaskan secara lebih rinci context diagram aplikasi layanan surat ke dalam beberapa proses sesuai dengan kebutuhan sistem.



Gambar 4.5. DFD level 1

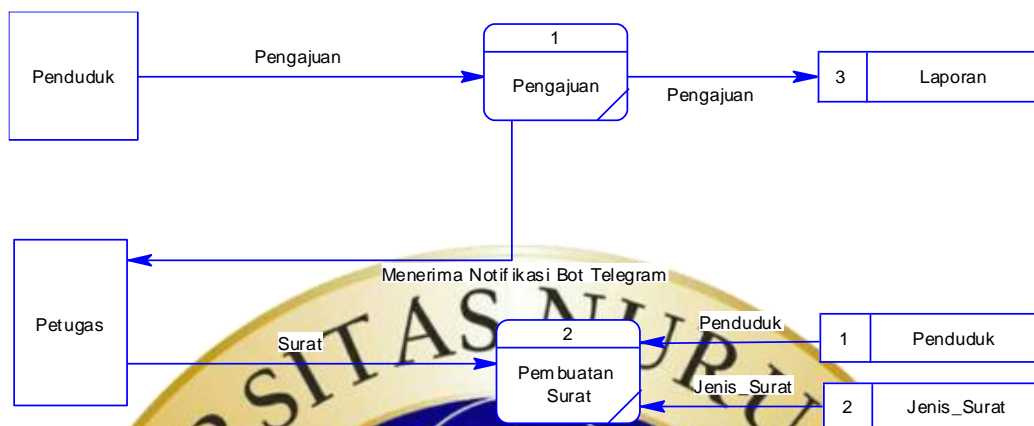
2) DFD level 2 proses input data



Gambar 4.6. DFD level 2 Proses Input Data



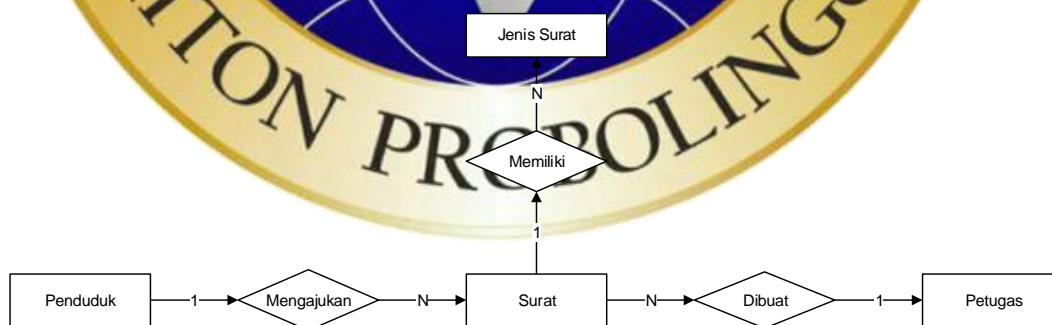
3) DFD level 2 proses 2 Pembuatan Surat



Gambar 4. 7.DFD level 2 Pembuatan Surat

5. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan suatu diagram yang digunakan untuk menunjukkan relasi antar entitas ataupun objek. Menentukan hubungan antar entitas yang membentuk sistem ini diperlukan gambaran yang sistematis untuk memudahkan dalam bentuk tabel data. Berikut hubungan antar entitas dari aplikasi layanan surat di Desa Sumberrejo.



Gambar 4. 8. Entity Relationship Diagram (ERD)

6. Kamus Data

- a. Penduduk : no\_kk, nik, nama\_lengkap, stat\_hbkel, jenis\_kelamin,

tmpt\_lahir, tgl\_lhr, status\_kawin, nama\_ibu, nama\_ayah, agama, jenis\_pekerjaan, alamat, no\_rt, no\_rw, nm\_kel, nm\_kec, keterangan

b. Surat : tanggal, jenis\_surat, no\_surat, no\_kk, nik, nama\_lengkap, stat\_hbkel, jenis\_kelamin, tmpt\_lahir, tgl\_lhr, status\_kawin, agama, bulan\_jenis\_pekerjaan, alamat, no\_rt, no\_rw, nm\_kel, nm\_kec, keterangan, keperluan, bulan.

c. Jenis\_Surat : jenis\_surat

d. Tahun : kode\_tahun, tahun\_awal, tahun\_akhir, nama\_kades, nip, status

e. Petugas : user, nama\_petugas, nip\_petugas, tempat\_lahir, pendidikan\_petugas, email, alamat, no\_hp.

#### 4.2.4 Desain Database

Desain database sebagai tempat untuk menampung semua data yang diinputkan dari sebuah sistem. Database yang digunakan adalah MySQL yang terdiri dari beberapa tabel. Berikut merupakan desain database dari aplikasi layanan surat di desa Sumberrejo.

##### a. Data Penduduk

Tabel data penduduk menyimpan semua data penduduk yang dilakukan oleh petugas, menginputkan data mulai NIK, nama, tempat lahir dan lain lain.

**Tabel 4. 1.** Penduduk

No	Nama	Jenis	Ukuran
1	no_kk	CharField	16
2	Nik	CharField	16
3	nama_lengkap	CharField	100
4	Stat_hbkel	Enum	30
5	jenis_kelamin	Enum	20
6	Tempat_lahir	Enum	25
7	Tanggal_lahir	CharField	25
8	Status_kawin	Enum	15
9	Nama_ibu	CharField	100



10	Nama_Ayah	CharField	100
11	Agama	CharField	10

**Tabel 4. 2.** Penduduk (Lanjutan)

No	Nama	Jenis	Ukuran
12	Jenis_pekerjaan	Enum	100
13	Alamat	CharField	20
14	No_RT	CharField	3
15	No_RW	CharField	3
16	Nm_Kel	CharField	25
17	Nm_kec	CharField	25
18	Keterangan	CharField	50

b. Surat

Tabel Surat menampilkan semua data proses pembuatan surat sesuai dengan kebutuhan surat yang diperlukan.

**Tabel 4. 3.** Surat

No	Nama	Jenis	Ukuran
1	No_surat	Charfield	10
2	Tanggal	CharField	20
3	Jenis_surat	Enum	50
4	Nik	CharField	16
5	nama_lengkap	CharField	100
6	Stat_hbkel	Enum	30
7	Jenis_kelamin	Enum	20
8	Tempat_lahir	Enum	25
9	Tanggal_lahir	CharField	25
10	Status_kawin	Enum	15
11	Agama	CharField	10
12	Jenis_pekerjaan	CharField	100
13	Alamat	CharField	50
14	No_rt	Enum	3

15	No_rw	CharField	3
16	Nm_kel	CharField	20

**Tabel 4. 4.** Surat (Lanjutan)

No	Nama	Jenis	Ukuran
17	Nm_kec	CharField	20
18	Keterangan	CharField	100
19	Catatan	CharField	250
20	Bulan	CharField	20

c. Jenis Surat

Tabel jenis surat menyimpan semua data jenis surat yang ada di Desa Sumberrejo.

**Tabel 4. 5.**Jenis Surat

No	Nama	Jenis	Ukuran
1	jenis_surat	CharField	50

d. Petugas

Tabel saya menyimpan semua data saya atau data petugas yang ada di Desa Sumberrejo.

**Tabel 4. 6.** Petugas

No	Nama	Jenis	Ukuran
1	Nama_Petugas	CharField	50
2	Nip_petugas	CharField	18
3	Tempat_lahir	CharField	50
4	Pendidikan_Petugas	CharField	20
5	Email	CharField	30
6	Alamat	CharField	100
7	No_hp	CharField	12

e. Tahun

Tabel Tahun menyimpan semua data kepala desa yang ada di Desa Sumberrejo.

**Tabel 4. 7.** Tahun

No	Nama	Jenis	Ukuran
1	Kode_tahun	CharField	5
2	Tahun_awal	CharField	4
3	Tahun_akhir	CharField	4
4	Nama_kades	CharField	100
5	Nip	CharField	30
6	Status	CharField	20

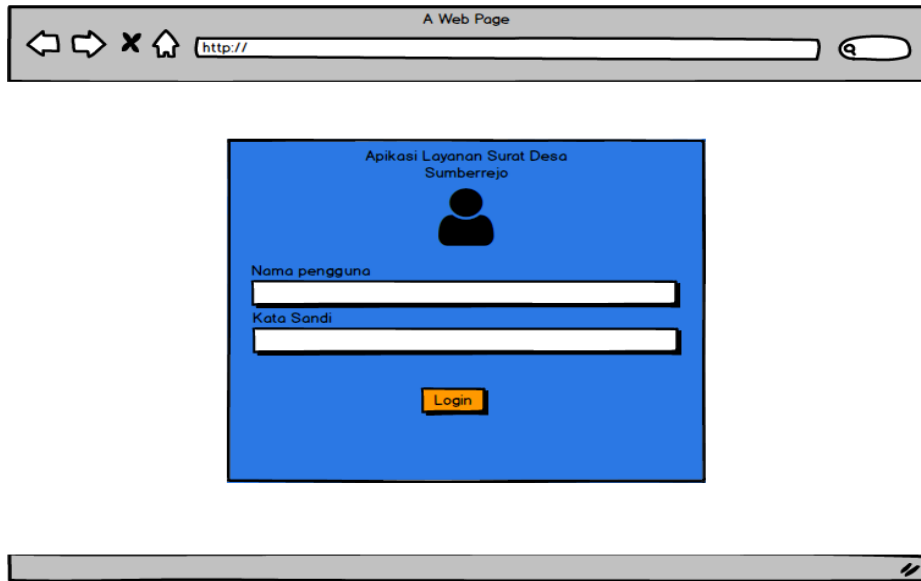
#### 4.2.5 Desain Input dan Output

Pada tahap 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 layanan surat ini ialah halaman input login, halaman input penduduk, dan halaman input layanan surat. Desain output adalah hasil dari proses yang diinputkan oleh pengguna yang berisikan informasi hasil pembuatan surat. Dalam aplikasi ini terdapat beberapa desain input dan output meliputi :

a. Desain Halaman Login

Menu login merupakan menu halaman pertama saat aplikasi layanan surat dijalankan dan di halaman ini terdapat form yang harus diisi oleh pengguna untuk dapat mengakses halaman utama menurut hak akses masing-masing.

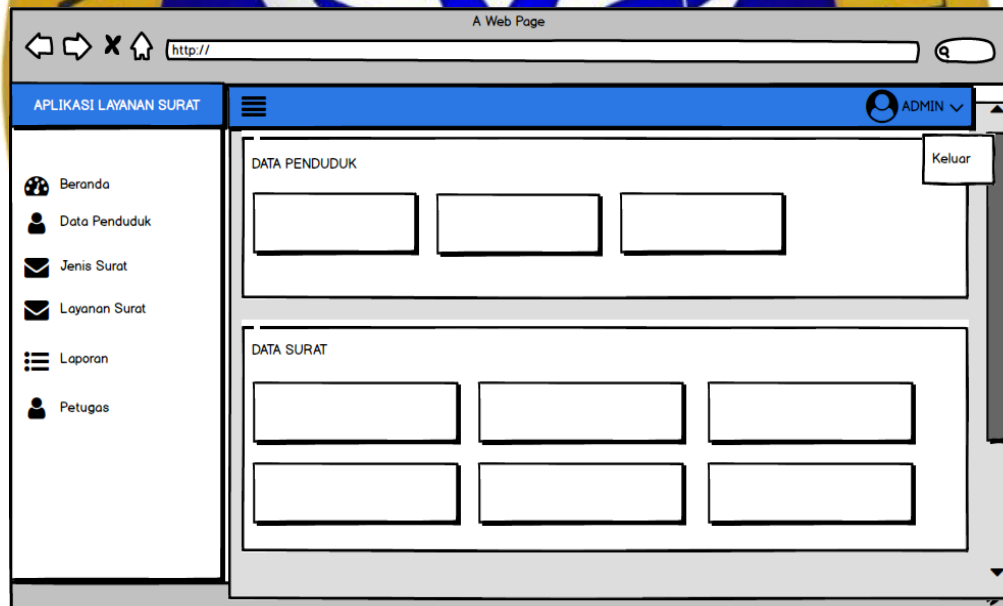




**Gambar 4. 9.** Desain Halaman Login

b. Desain Halaman Utama

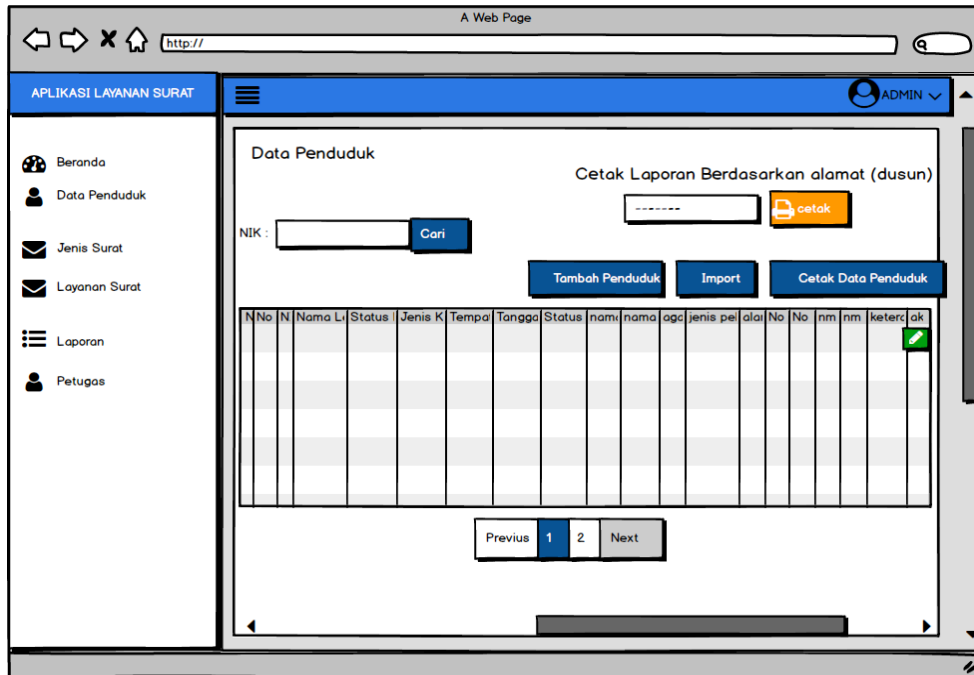
Saat pengguna berhasil melakukan login, maka akan diarahkan ke halaman beranda



**Gambar 4. 10.** Desain Halaman Utama

c. Desain Data Penduduk

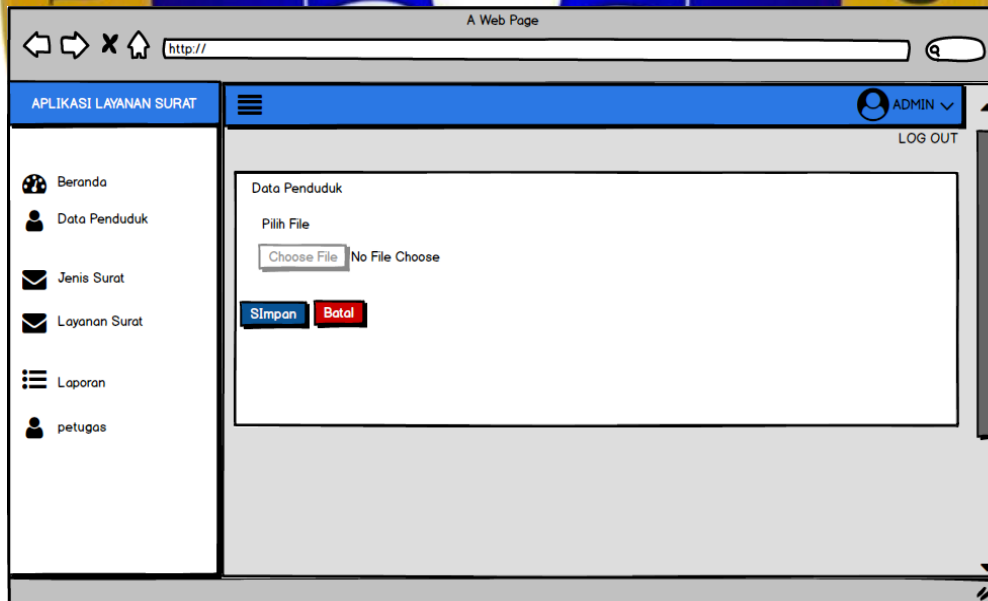
Menu data penduduk merupakan menu yang berisikan data penduduk yang telah diinputkan oleh petugas.



**Gambar 4. 11.**Desain Halaman Penduduk

d. Desain Import Data Penduduk

Menu ini digunakan oleh petugas untuk import data penduduk dalam jumlah banyak



**Gambar 4. 12.** Desain Import Data Penduduk

e. Desain Tambah Data Penduduk

Menu tambah data penduduk merupakan menu yang digunakan oleh petugas untuk menginputkan semua data data penduduk desa Sumberrejo.

The image shows a web browser window displaying a form titled "Tambah Data Penduduk" (Add Resident Data). The browser's address bar shows "http://". The page header includes "APLIKASI LAYANAN SURAT" on the left and "ADMIN" on the right. A sidebar on the left contains navigation links: Beranda, Data Penduduk, Jenis Surat, Layanan Surat, Laporan, and Petugas. The main content area contains the following form fields:

- no KK
- NIK
- Nama Lengkap
- Stat\_hbkel
- Jenis Kelamin
- Tempat Lahir
- Tanggal Lahir
- Status\_kawin
- Nama ibu
- Nama Ayah
- agama
- Jenis Pekerjaan
- alamat
- no rt
- no rw
- nm kel Keterangan
- nm kec
- keterangan

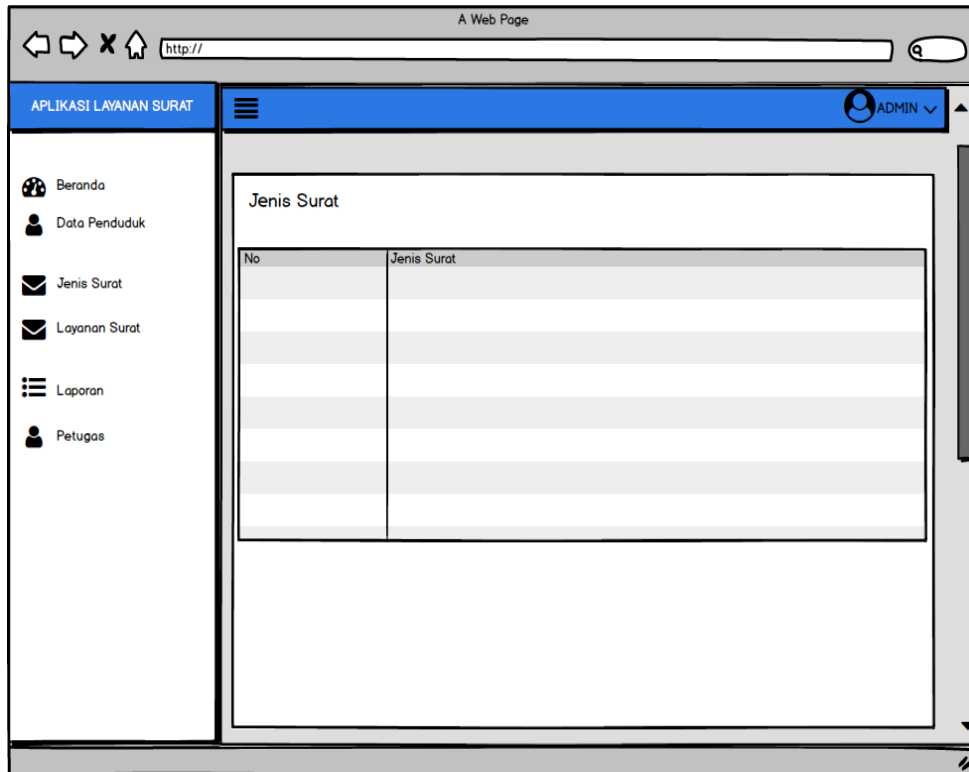
At the bottom of the form are two buttons: "Simpan" (Save) and "Batal" (Cancel).

**Gambar 4. 13.** Desain Tambah Data Penduduk

f. Desain jenis Surat

Menu data penduduk merupakan menu yang berisikan data penduduk yang telah diinputkan oleh petugas.





**Gambar 4.14.** Desain Jenis Surat

g. Desain Layanan Surat

Menu input layanan surat merupakan menu yang digunakan oleh petugas untuk menginputkan proses pembuatan surat



The image shows a web browser window displaying a web application titled "APLIKASI LAYANAN SURAT". The browser's address bar shows "http://". The application has a sidebar menu on the left with the following items: Beranda, Data Penduduk, Jenis Surat, Layanan Surat, Laporan, and Petugas. The main content area is titled "Input Layanan Surat" and contains a form with the following fields:

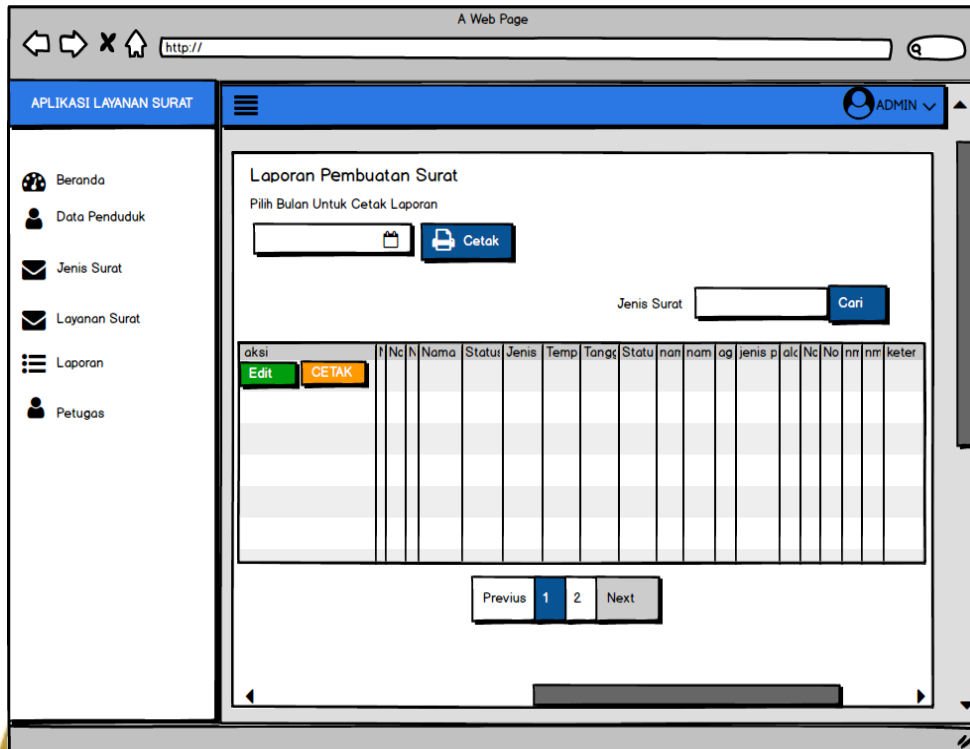
- Tanggal
- Jenis Surat
- NIK
- Nama Lengkap
- Stat\_hbkel (dropdown menu)
- Jenis Kelamin (dropdown menu)
- Tempat Lahir
- Tanggal Lahir
- Status\_kawin (dropdown menu)
- Nama ibu
- Nama Ayah
- agama
- Jenis Pekerjaan (dropdown menu)
- alamat
- no rt
- no rw
- nm kel
- nm kec
- Keterangan
- catatan
- pilih bulan

At the bottom of the form, there are two buttons: "Simpan" (Save) and "Batal" (Cancel).

Gambar 4. 15. Desain Layanan Surat

h. Desain laporan

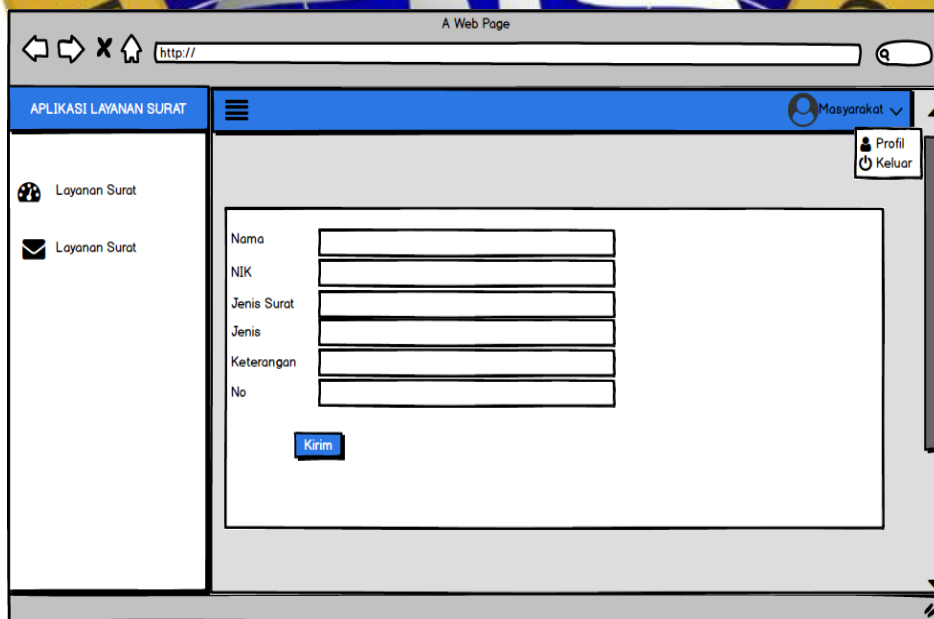
Menu data penduduk merupakan menu yang berisikan data penduduk yang telah diinputkan oleh petugas.



**Gambar 4. 16. Desain Laporan**

i. Desain Layanan Masyarakat

Menu input layanan surat merupakan menu yang digunakan oleh masyarakat untuk menginputkan proses pengajuan surat



**Gambar 4. 17. Desain Layanan Masyarakat**

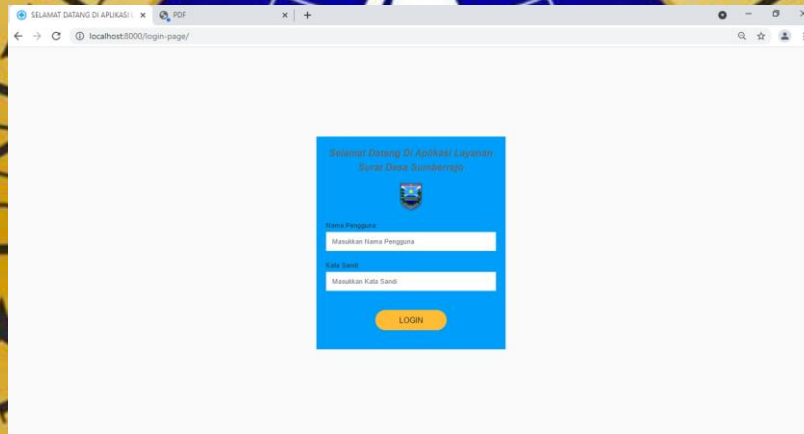


### 4.3 Pemograman (Coding)

Setelah desain dari aplikasi layanan surat surat berhasil dibuat , langkah selanjutnya adalah melakukan implementasi desain tersebut menjadi kode program. Dibawah ini adalah implementasi dari Aplikasi Layanan Surat Berbasis Web Menggunakan *Framework Django* Terintegrasi dengan Notifikasi Bot Telegram di Desa Sumberrejp, meliputi.

#### 1. Halaman login

Halaman login ini digunakan oleh pengguna yaitu, petugas, masyarakat dan kepala desa dengan memasukkan nama pengguna dan kata sandi. Halaman login ini digunakan untuk masuk kehalaman utama sistem. Berikut halaman login pada aplikasi layanan surat terlihat pada gambar 4.18 Halaman Login.



**Gambar 4.18.** Halaman Login

Kode :

```
def loginPage(request):
    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')

        cocok = authenticate(request, username=username,
            password=password)
        if cocok is not None:
            login(request, cocok)
            return redirect('beranda')
        else:
            messages.error(request, "Username / Password Tidak
            Cocok")
            return redirect('login')
```

```

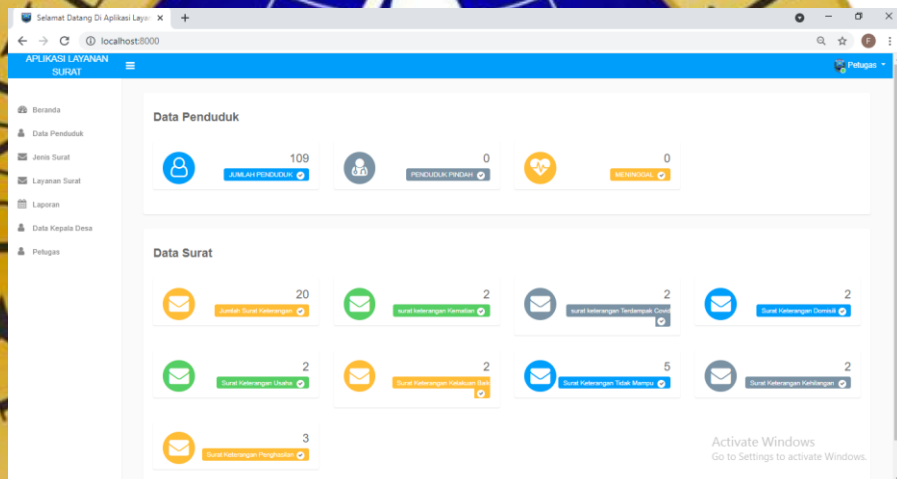
context = {
    "menu" : 'Halaman Login',
    "page" : 'login',
}
return render(request, 'surat/login.html', context)

```

### Segmen Program 4. 1. Halaman Login

#### 2. Halaman Beranda

Setelah pengguna melakukan proses login maka pengguna akan menuju ke halaman beranda berdasarkan hak akses pengguna. Pada halaman beranda ini mengelola menu-menu meliputi, data penduduk mengelola data penduduk, jenis surat mengelola jenis surat, layanan surat mengelola proses pembuatan surat, laporan mengelola laporan pembuatan surat yang sudah dibuat. Berikut Gambar 4.19 Halaman Beranda.



Gambar 4. 19. Halaman Beranda

Kode :

```

def beranda(request):
    datapenduduk = Penduduk.objects.all()
    list_surat = Surat.objects.all()
    total_penduduk = Penduduk.objects.count()
    total_kematian = datapenduduk.filter(KETERANGAN =
'MENINGGAL'). count()
    total_pindah= datapenduduk.filter(KETERANGAN =
'PINDAH'). count()
    total_surat = Surat.objects.count()
    total_mati= list_surat.filter(JENIS_SURAT = 'Surat Keterangan
Kematian'). count()
    total_domisili= list_surat.filter(JENIS_SURAT = 'Surat

```

```

Keterangan Domisili'). count()
total_covid = list_surat.filter(JENIS_SURAT = 'Surat
Keterangan Terdampak Covid'). count()
total_usaha= list_surat.filter(JENIS_SURAT = 'Surat
Keterangan Usaha'). count()
total_skkb = list_surat.filter(JENIS_SURAT = 'Surat Keterangan
Kelakuan Baik'). count()
total_sktm = list_surat.filter(JENIS_SURAT = 'Surat
Keterangan Tidak Mampu'). count()
total_hilang = list_surat.filter(JENIS_SURAT = 'Surat
Keterangan Kehilangan'). count()
total_hasil = list_surat.filter(JENIS_SURAT = 'Surat Keterangan
Penghasilan'). count()
context = {
    "menu" : 'Beranda',
    "page" : 'Selamat Datang di Aplikasi Layanan Surat',
    'dt_penduduk' : datapenduduk,
    'data_penduduk' : total_penduduk,
    'data_kematian' : total_kematian,
    'data_pendudukpindah' : total_pindah,
    'jumlah_surat' : total_surat,
    'surat_kematian' : total_mati,
    'surat_covid' : total_covid,
    'surat_domisili' : total_domisili,
    'surat_usaha' : total_usaha,
    'surat_skkb' : total_skkb,
    'surat_sktm' : total_sktm,
    'surat_hilang' : total_hilang,
    'surat_hasil' : total_hasil,
}
return render(request, 'surat/beranda.html', context)

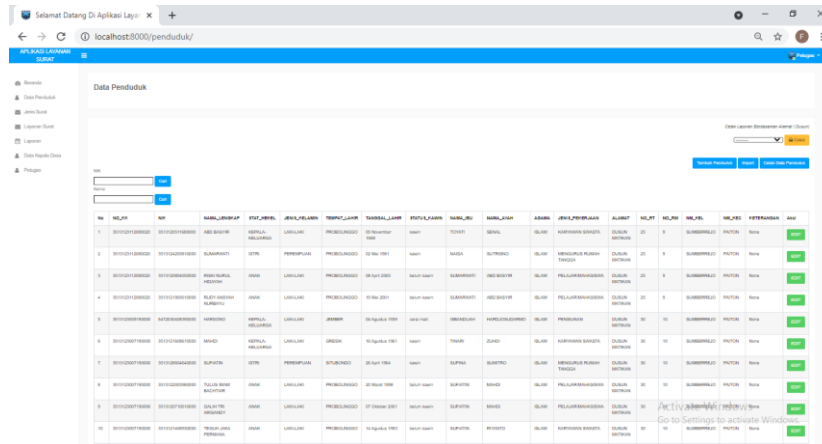
```

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

### 3. Halaman Data Penduduk

Data penduduk ini dikelola oleh petugas Desa Sunberrejo. Pada halaman data Penduduk terdapat beberapa fitur meliputi, tambah data penduduk, import data penduduk, ekspor data penduduk dan edit data penduduk. Berikut gambar 4.20 Halaman data Penduduk.





Gambar 4.20. Halaman Data Penduduk

Kode :

```

@login_required(login_url='login')
@ijinkan_pengguna(yang_diizinkan=['petugas'])
def penduduk(request):
    list_penduduk = Penduduk.objects.order_by('-NO_KK')
    filter_penduduk = PendudukFilter(request.GET,
    queryset=list_penduduk)
    list_penduduk=filter_penduduk.qs
    halaman_tampil = Paginator(list_penduduk, 100)
    halaman_url = request.GET.get('halaman',1)
    halaman_penduduk = halaman_tampil.get_page(halaman_url)
    if halaman_penduduk.has_previous():
        url_previous =
    f'?halaman={halaman_penduduk.previous_page_number()}'
    else:
        url_previous = ""
    if halaman_penduduk.has_next():
        url_next =
    f'?halaman={halaman_penduduk.next_page_number()}'
    else:
        url_next = ""
    context = {
        "menu" : 'Data ',
        "page" : 'Data Penduduk',
        "tampil" : list_penduduk,
        'halaman_list_penduduk':halaman_penduduk,
        'filter_list_penduduk': filter_penduduk,
        'previous' : url_previous,
        'next' : url_next
    }
    return render(request, 'surat/penduduk.html', context)
@login_required(login_url='login')
@ijinkan_pengguna(yang_diizinkan=['petugas'])
def tambahpenduduk(request):

```

```

tambahpenduduk = DataForm()
if request.method == 'POST' :
    #print('cetak POST:', request.POST)
    formsimpan = DataForm(request.POST)
    if formsimpan.is_valid:
        formsimpan.save()
        return redirect('penduduk')
context = {
    "menu" : 'tambah penduduk',
    "page" : 'tambah-Penduduk',
    "tambahpenduduk": tambahpenduduk,
}
return render(request, 'surat/tambahpenduduk.html', context)
@login_required(login_url='login')
@ijinkan_pengguna(yang_diizinkan=['petugas'])
def editpenduduk (request, pk):
    up_penduduk = Penduduk.objects.get(id=pk)
    editpenduduk = DataForm(instance=up_penduduk)
    if request.method == 'POST':
        # print('Cetak POST:', request.POST)
        formedit = DataForm(request.POST, instance=up_penduduk)
        if formedit.is_valid:
            formedit.save()
            return redirect('penduduk')
    context = {
        "menu" : 'editpenduduk',
        "page" : 'Edit Penduduk',
        "editpenduduk" : editpenduduk,
    }
    return render(request, 'surat/editpenduduk.html', context)

def upload(request) :
    if request.method == 'POST' :
        penduduk_resource = PendudukResource()
        dataset = Dataset()
        new_penduduk = request.FILES['myfile']

        if not new_penduduk.name.endswith('xlsx'):
            messages.info(request, 'wrong format')
            return render(request, 'upload.html')
        imported_data = dataset.load(new_penduduk.read(),
format='xlsx')
        for data in imported_data:
            value = Penduduk(
                data[0],
                data[1],
                data[2],
                data[3],

```

```

data[4],
data[5],
data[6],
data[7],
data[8],
data[9],
data[10],
data[11],
data[12],
data[13],
data[14],
data[15],
data[16],
data[17],
data[18]
)
value.save()
return render(request, 'upload.html')
def export (request):
penduduk_resource = PendudukResource()
dataset = penduduk_resource.export()
response = HttpResponse(dataset.xls, content_type =
'application/vnd.ms-excel')
response['Content-Disposition'] = 'attachment; filename="Data
Penduduk.xls"'
return response

```

### Segmen Program 4. 3. Halaman Data Penduduk

#### 4. Halaman jenis Surat

Pada halaman jenis surat terdapat tabel yang beri nomer, jenis surat. Berikut gambar 4.21 Halaman Jenis Surat

NO	JENIS SURAT
1	surat keterangan usaha
2	Surat Keterangan Tidak Mampu
3	Surat Keterangan Terdampak Covid
4	Surat Keterangan Penghasilan
5	Surat Keterangan Kamatan
6	Surat Keterangan Kelakuan Baik
7	Surat Keterangan Kehilangan
8	Surat Keterangan Domisili

**Gambar 4. 21.** Halaman Jenis Surat



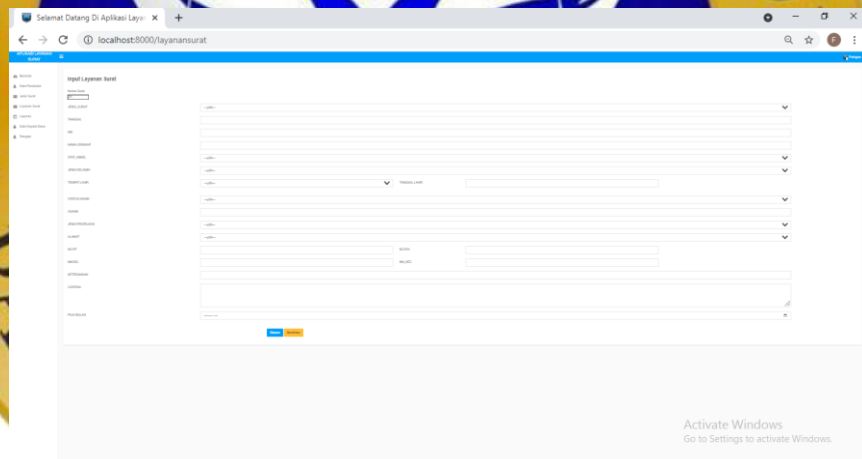
Kode :

```
@login_required(login_url='login')
@ijinkan_pengguna(yang_diizinkan=['petugas'])
def jenissurat(request):
    list_jenissurat = Jenis_Surat.objects.order_by('-JENIS_SURAT')
    context = {
        "menu" : 'Jenis',
        "page" : 'Jenis Surat',
        "tampil" : list_jenissurat,
    }
    return render(request, 'surat/jenissurat.html', context)
```

#### Segmen Program 4. 4. Halaman Jenis Surat

##### 5. Halaman layanan surat

Pada halaman layanan surat akan muncul halaman inputan layanan surat. Input layanan surat dilakukan oleh petugas untuk menginputkan data yang diperlukan untuk proses pembuatan surat, sesuai dengan kebutuhan surat pemohon. Berikut Gambar 4.22 Halaman Layanan Surat.



Gambar 4. 22. Halaman Layanan Surat

Kode :

```
@login_required(login_url='login')
@ijinkan_pengguna(yang_diizinkan=['petugas'])
def layanansurat (request):
    no = Surat.objects.all().count()
    nosurat = 1 + int(no)
    context = {
        "menu" : 'halaman input surat',
        "page" : 'tambah surat',
        'nosurat' : nosurat
    }
    }
```



```

return render(request, 'surat/inputsurat.html', context)

@login_required(login_url='login')
@ijinkan_pengguna(yang_diizinkan=['petugas'])
def inputsurat(request):
    if request.method == 'POST':
        NO_SURAT = request.POST.get('NO_SURAT')
        JENIS_SURAT = request.POST.get('JENIS_SURAT')
        TANGGAL= request.POST.get('TANGGAL')
        NIK = request.POST.get('NIK')
        NAMA LENGKAP =
request.POST.get('NAMA LENGKAP')
        STAT_HBKEL = request.POST.get('STAT_HBKEL')
        JENIS_KELAMIN = request.POST.get('JENIS_KELAMIN')
        TMPT_LAHIR = request.POST.get('TMPT_LAHIR')
        TGL_LHR = request.POST.get('TGL_LHR')
        STATUS_KAWIN = request.POST.get('STATUS_KAWIN')
        AGAMA = request.POST.get('AGAMA')
        JENIS_PKERJAAN =
request.POST.get('JENIS_PKERJAAN')
        ALAMAT = request.POST.get('ALAMAT')
        NO_RT = request.POST.get('NO_RT')
        NO_RW = request.POST.get('NO_RW')
        NM_KEL = request.POST.get('NM_KEL')
        NM_KEC = request.POST.get('NM_KEC')
        KETERANGAN = request.POST.get('KETERANGAN')
        CATATAN = request.POST.get('CATATAN')
        BULAN = request.POST.get('BULAN')

        simpan =
Surat.objects.create(NO_SURAT=NO_SURAT,JENIS_SURAT=J
ENIS_SURAT,TANGGAL=TANGGAL, NIK=NIK,
NAMA LENGKAP=NAMA LENGKAP,
STAT_HBKEL=STAT_HBKEL,
JENIS_KELAMIN=JENIS_KELAMIN,
TMPT_LAHIR=TMPT_LAHIR, TGL_LHR=TGL_LHR,
STATUS_KAWIN=STATUS_KAWIN, AGAMA=AGAMA,
JENIS_PKERJAAN=JENIS_PKERJAAN,
ALAMAT=ALAMAT, NO_RT=NO_RT, NO_RW=NO_RW,
NM_KEL=NM_KEL, NM_KEC=NM_KEC,
KETERANGAN=KETERANGAN, CATATAN=CATATAN,
BULAN=BULAN )
        simpan.save()
    return JsonResponse({'status' : 0})

```

#### Segmen Program 4. 5. Halaman Layanan Surat



```

        'halaman_list_surat': halaman_surat,
        'filter_list_surat': filter_surat,
        'previous' : url_previous,
        'next' : url_next,
    }
    return render(request, 'surat/surat.html', context)
def editsurat (request, pk):
    ed_surat = Surat.objects.get(id=pk)
    editsurat = SuratForm(instance=ed_surat)
    if request.method == 'POST':
        # print('Cetak POST:', request.POST)
        formedit = SuratForm(request.POST, instance=ed_surat)
        if formedit.is_valid():
            formedit.save()
            return redirect('surat')
    context = {
        "menu" : 'edit surat',
        "page" : 'Edit Surat',
        "editsurat" : editsurat,
    }
    return render(request, 'surat/editsurat.html', context)

#laporan perbulan
def LaporanBView(request):
    kepala_desa = Tahun.objects.filter(status='Aktif')
    if 'BULAN' in request.GET:
        cek=request.GET['BULAN']
        tampil_data = Surat.objects.filter(BULAN=cek)
    else:
        tampil_data = Surat.objects.filter(BULAN=None)

    context = {'tampil_data': tampil_data, 'kepala_desa':
kepala_desa,}
    return render(request, 'surat/lp_perbulan.html', context)

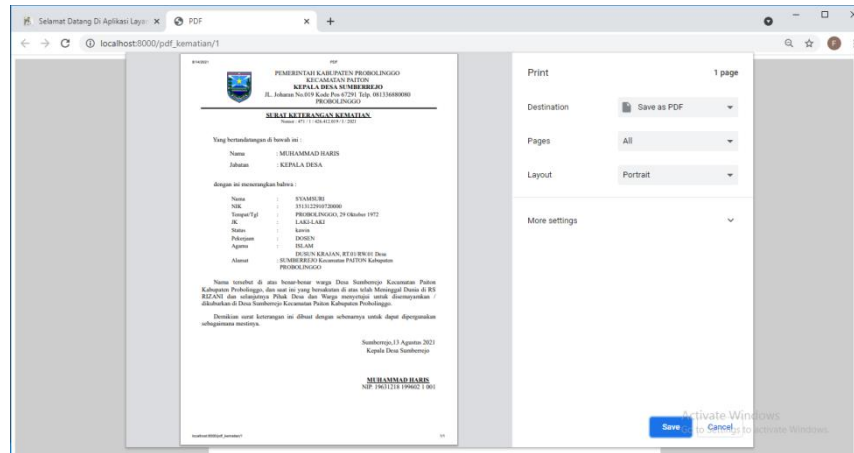
```

#### Segmen Program 4. 6. Halaman Laporan

##### 7. Cetak Surat pdf

Pada halaman ini menampilkan cetak surat dalam bentuk pdf, kemudian petugas dapat menyimpan hasil cetak pdf dan dapat mencetak langsung hasil pdf. Berikut Gambar 4.24 Cetak Surat Pdf





Gambar 4. 24. Cetak Surat Pdf

Kode :

```
#pdf surat
def pdf_usaha(request, pk):
    buat_surat = Surat.objects.get(id=pk)
    kepala_desa = Tahun.objects.filter(status='Aktif')
    context = {'buat_surat': buat_surat, 'kepala_desa': kepala_desa,}
    return render(request, 'surat/pdf_usaha.html', context)
def pdf_covid(request, pk):
    buat_surat = Surat.objects.get(id=pk)
    kepala_desa = Tahun.objects.filter(status='Aktif')
    context = {'buat_surat': buat_surat, 'kepala_desa': kepala_desa,}
    return render(request, 'surat/pdf_covid.html', context)
def pdf_tidakmampu(request, pk):
    buat_surat = Surat.objects.get(id=pk)
    kepala_desa = Tahun.objects.filter(status='Aktif')
    context = {'buat_surat': buat_surat, 'kepala_desa': kepala_desa,}
    return render(request, 'surat/pdf_tidakmampu.html', context)
def pdf_kelakuanbaik(request, pk):
    buat_surat = Surat.objects.get(id=pk)
    kepala_desa = Tahun.objects.filter(status='Aktif')
    context = {'buat_surat': buat_surat, 'kepala_desa': kepala_desa,}
    return render(request, 'surat/pdf_kelakuanbaik.html', context)
def pdf_kehilangan(request, pk):
    buat_surat = Surat.objects.get(id=pk)
    kepala_desa = Tahun.objects.filter(status='Aktif')
    context = {'buat_surat': buat_surat, 'kepala_desa': kepala_desa,}
    return render(request, 'surat/pdf_kehilangan.html', context)
def pdf_domisili(request, pk):
    buat_surat = Surat.objects.get(id=pk)
    kepala_desa = Tahun.objects.filter(status='Aktif')
    context = {'buat_surat': buat_surat, 'kepala_desa': kepala_desa,}
    return render(request, 'surat/pdf_domisili.html', context)
```



```

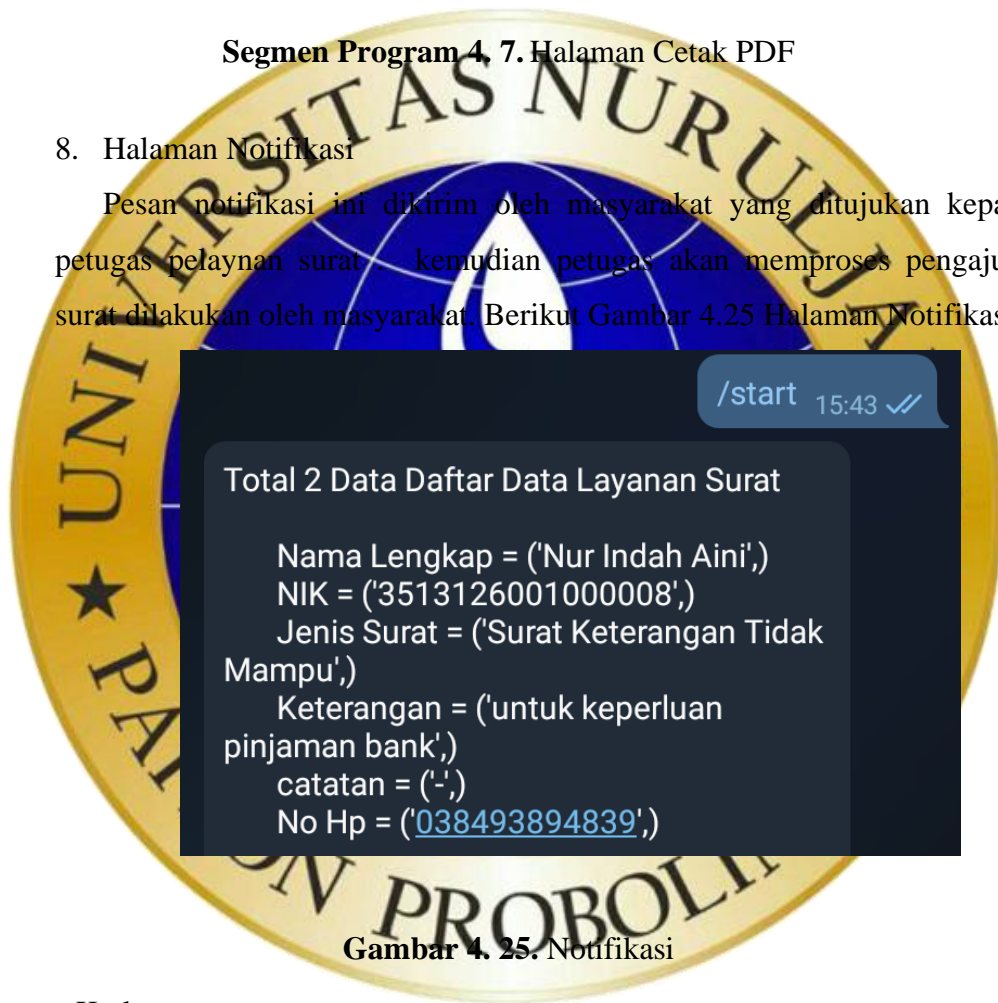
def pdf_kematian(request, pk):
    buat_surat = Surat.objects.get(id=pk)
    kepala_desa = Tahun.objects.filter(status='Aktif')
    context = {'buat_surat': buat_surat, 'kepala_desa': kepala_desa,}
    return render(request, 'surat/pdf_kematian.html', context)
def pdf_penghasilan(request, pk):
    buat_surat = Surat.objects.get(id=pk)
    kepala_desa = Tahun.objects.filter(status='Aktif')
    context = {'buat_surat': buat_surat, 'kepala_desa': kepala_desa,}
    return render(request, 'surat/pdf_penghasilan.html', context)

```

#### Segmen Program 4. 7. Halaman Cetak PDF

#### 8. Halaman Notifikasi

Pesan notifikasi ini dikirim oleh masyarakat yang ditujukan kepada petugas pelayanan surat. kemudian petugas akan memproses pengajuan surat dilakukan oleh masyarakat. Berikut Gambar 4.25 Halaman Notifikasi.



Gambar 4. 25. Notifikasi

Kode :

```

import telebot
import django
import os
os.environ['DJANGO_SETTINGS_MODULE'] = 'aplikasi.settings'
django.setup()
#from .models import *
from telegram.ext import *
from surat.models import*
api =
'1817619498:AAESj3i69z5Q6mcW27GPaYUE117GYHsR1_k'

```

```

bot = telebot.TeleBot(api)

@bot.message_handler(commands=['start'])
def tampildata(update, context):
    text = str(update.message.text).lower()
    data_surat = Masyarakat.objects.all()
    message = f"" Total {data_surat.count()} Data ""
    for data in data_surat: message += f""Daftar Data Layanan
Surat
Nama Lengkap = {data.nama_lengkap}
NIK = {data.nik}
Jenis Surat = {data.jenis_surat}
Keterangan = {data.keterangan}
catatan = {data.catatan}
No Hp = {data.no_telp} \n \n
""
    update.message.reply_text(f"{message}")
    return
    #print(update)
    #update.message.reply_text(f"Hi,
{update['message']['chat']['first_name']} Informasi Layanan (Surat
Enter NIK")
    print('bot Success Running')
    print(tampildata)
    #bot.polling()
if __name__ == '__main__':
    updater = Updater(api, use_context = True)
    dp = updater.dispatcher
    dp.add_handler(MessageHandler(Filters.text, tampildata))

    updater.start_polling(1.0)
    updater.idle()

```

Segmen Program 4. 8. Halaman Notifikasi

#### 4.4 Pengujian

##### 1. Hasil Pengujian Internal

Pengujian internal ini merupakan hasil pengujian program berdasarkan fungsi dari program. Tujuan dari pengujian internal ini adalah untuk menemukan kesalahan fungsi pada program sebelum dilakkan pengujian eksternal yang melibatkan pihak dari objek pengamatan, yaitu Desa Sumberrejo. Berikut Tabel 4.7 hasil Pengujian Internal dalam aplikasi layanan surat.

**Tabel 4. 8.** Pengujian Internal

No	Pengujian	Target yang dicapai	Hasil yang diharapkan	Keterangan	
				Ya	Tidak
1	Masuk	Nama Pengguna dan sandi benar	Menuju beranda		
		Nama Pengguna dan sandi salah	Tidak dapat menuju beranda		
2	Beranda	Halaman Beranda	Menampilkan Halaman Beranda		
3	Data Penduduk	Data Penduduk	Menampilkan data penduduk		
		Cari penduduk	Menampilkan data penduduk sesuai pencarian		
		Tambah penduduk	Menampilkan suatu inputan		
		Export data	Dapat mendownload data penduduk		
		Import data	Dapat meng-Upload data		
4	Layanan Surat	Input layanan surat	Menampilkan inputan layanan surat		
5	Laporan	Laporan pembuatan surat	Menampilkan laporan pembuatan surat		
		Cetak surat	Dapat mencetak surat permohonan masysrakat		



**Tabel 4. 9.** Pengujian Internal (Lanjutan)

No	Pengujian	Target yang dicapai	Hasil yang diharapkan	Keterangan	
				Ya	Tidak
6	Laporan	Cetak laporan per bulan	Dapat mencetak data pembuatan surat		
7	Telegram	Notifikasi Bot Telegram	Dapat memberikan notifikasi tentang permohonan surat kepada aparat desa dengan notifikasi Bot Telegram.		
8	Keluar	keluar	Kembali ke Login		

Berdasarkan hasil pengujian diatas, dapat disimpulkan bahwa aplikasi layanan surat sudah memenuhi kebutuhan sistem yang diharapkan, akan tetapi kesalahan dari aplikasi masih memungkinkan untuk terjadi, sehingga perlu adanya perawatan dari aplikasi yang dibuat.

## 2. Hasil Pengujian Eksternal

Pengujian eksternal merupakan pengujian yang dilakukan secara langsung kepada pengguna yang bersangkutan dengan mengisi kuisioner yang telah dibuat, yang kemudian dibagikan kepada beberapa pengguna yang akan melakukan pengujian aplikasi yang telah dibuat, dengan mengambil 20 sampel. Berikut Tabel 4.10 hasil Pengujian Eksternal.

**Tabel 4. 10.** Pengujian Eksternal

No	Draf Pertanyaan	Jawaban			
		Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
1	Apakah dengan adanya Aplikasi ini memudahkan serta mempercepat dalam proses pelayanan surat	1	4	0	0



**Tabel 4. 11.** Penujian Eksternal(Lanjutan)

No	Draf Pertanyaan	Jawaban			
		Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
2	Apakah aparat desa mudah dalam mengoprasikan aplikasi ini ?	0	3	2	0
3	Apakah dengan adanya aplikasi ini, dapat membantu dalam proses pelaporan?	0	2	2	0
4	Apakah fitur-fitur system ini mudah di operasikan oleh pengguna?	0	5	0	0
5	Apakah aplikasi ini sudah layak digunakan ?	0	5	0	0
6	Apakah ada kendala dalam mengoperasikan aplikasi ini ?	0	4	1	0
Total		1	23	6	

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

a. Bobot setiap skala

Sangat Setuju = 4

Setuju = 3

Tidak Setuju = 2

Sangat Tidak Setuju = 1

b. Nilai Interval

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

$I = 100 / 4$

$I = 25$

Jadi persentase penilaian adalah sebagai berikut

**Tabel 4. 12.** Tabel Hasil Presentase

Jawaban	Keterangan
0% - 24.99%	Sangat Tidak Setuju
25% - 49.99%	Tidak Setuju
50% - 74.99%	Setuju
75% - 100%	Sangat Setuju

Jumlah skor perolehan dari semua responden :

Jumlah Sangat Setuju (SS) =  $1 \times 4 = 4$

Jumlah Setuju (S) =  $23 \times 3 = 69$

Jumlah Tidak Setuju (TS) =  $6 \times 2 = 12$

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

Total Skor =  $4 + 69 + 12 + 0 = 85$

Jumlah nilai ideal diperoleh dari nilai tertinggi x jumlah butir soal x jumlah responden =  $4 \times 6 \times 5 = 120$

Rumus Penentuan Skor =  $(\text{Skor Perolehan} / \text{Skor Ideal}) \times 100$ . Perhitungan  $(85 / 120) \times 100 = 70$  hasil perhitungan menggunakan metode skala likert didapatkan data sebanyak 70 % yang artinya Aplikasi Layanan Surat Berbasis Web Menggunakan *Framework Django* Terintegrasi dengan Notifikasi Bot Telegram Di Desa Sumberrejp ini “setuju” atau layak untuk digunakan.

#### 4.5 Perawatan

Perawatan suatu software diperlukan karena software yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada error

kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada software tersebut. Perawatan meliputi perbaikan eror dan perbaikan pemograman sistem sesuai dengan sistem yang dibutuhkan.

*Update* sistem merupakan hal yang paing penting dalam merawat sebuah sistem. *Update* sistem bertujuan untuk memperbarui data yang sudah ada maupun data yang masih ingin diinputkan, seperti *update* nama pengguna dan kata sandi. Hal ini dilakukan untuk mencegah terjadinya *hacking* pada sistem yang dilakukan oleh para *hacker* dunia maya.

