

## 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. Observasi

Berdasarkan observasi yang telah dilakukan di Mts Miftahul Khair menghasilkan bahwa pengelolaan dana Bantuan operasional Sekolah dilakukan dengan cara mencatat satu persatu pemasukan dan pengeluaran dana BOS ke dalam buku besar oleh karyawan tata usaha kemudian diketik ke dalam *Microsoft Excel*. Sedangkan untuk melaporkan hasil pengelolaan dana Bantuan Operasional Sekolah dilakukan dengan cara memeriksa dan mencocokkan antara buku besar dengan data yang ada di *Microsoft Excel* terlebih dahulu oleh bendahara. Kemudian dibuatlah buku laporan keuangan dana BOS yang kemudian diserahkan kepada pihak kepala sekolah untuk di cek kembali dan di sahkan sehingga dapat diserahkan kepada pihak Kemenag setiap tiga bulan sekali.

#### 4.1.2. Wawancara

Adapun hasil wawancara yang dilakukan pada tanggal 21 maret 2021 pada kepala tata usaha menghasilkan bahwa pengelolaan dana BOS dilakukan dengan menulis data pengeluaran dan pemasukan ke dalam buku kemudian di ketik kedalam komputer. Media yang digunakan dalam pengelolaan dana BOS yaitu buku untuk data hardcopy dan computer sebagai penyimpanan data softcopy. Hal ini dapat menyebabkan laporan pertanggung jawaban (LPJ) penggunaan dana bantuan operasional sekolah sering mengalami kesalahan seperti kesalahan dalam menjumlah, kesalahan dalam menulis, dan juga sering

terjadi hilangnya dokumen. Pada proses pembuatan Laporan Pertanggung Jawaban (LPJ) dana BOS masih dilakukan dengan cara mencatat terlebih dahulu ke dalam buku besar kemudian diketik ke dalam *Microsoft Excel*. Laporan pertanggung jawaban dana bantuan operasional sekolah diserahkan setiap tiga bulan sekali ke kemenag.

## **4.2. Hasil Analisis dan Perancangan Sistem**

Setelah melakukan pengumpulan data, langkah selanjutnya adalah melakukan pengembangan sistem dengan menggunakan model *prototype*. Analisis kebutuhan dibagi menjadi dua macam yaitu analisis sistem lama dan analisis sistem baru.

### **4.2.1. Analisis Sistem Lama**

Dana masuk dan keluar ditulis dahulu ke dalam sebuah buku, pencatatan ini dilakukan oleh bendahara kemudian di ketik kedalam komputer. Hasil pencatatan tersebut kemudian direkap menjadi laporan yang ditujukan kepada kepala sekolah untuk meminta persetujuan laporan pertanggung jawaban dana BOS. Setelah laporan disetujui dan disahkan maka laporan tersebut akan dikirim kepada kemenag. Laporan kepada kemenag dikirim dalam bentuk *hardcopy*.

### **4.2.2. Analisis Sistem Baru**

Aplikasi ini mampu melakukan pencatatan dana masuk dan dana keluar sehingga tidak perlu ditulis ke dalam sebuah buku. Hal ini juga dapat memudahkan proses pembuatan laporan. Dalam pembuatan laporan, bendahara tidak perlu mencatat, mengetik, dan mencetak karena dalam aplikasi ini disediakan fitur untuk mencetak laporan hasil pengelolaan dana BOS. Selain itu aplikasi ini mampu melakukan pencatatan dengan mudah dan aman. Analisis sistem terdiri dari 2 bagian yaitu kebutuhan fungsional dan kebutuhan non fungsional.

a. Kebutuhan Fungsional

Kebutuhan fungsional merupakan segala macam kebutuhan yang berkaitan langsung dengan aplikasi yang telah dibuat. Adapun kebutuhan yang ada dalam sistem ini sebagai berikut:

1. Mempermudah Bendahara dalam pembuatan laporan Dana BOS
2. Mempermudah dan mempercepat Bendahara dalam merekapitulasi Keuangan Dana BOS
3. Mempermudah Kepala Sekolah dan bendahara dalam mengelola dan menerima informasi mengenai pemasukan dan pengeluaran Dana BOS.

b. Kebutuhan Non Fungsional

1. Kebutuhan perangkat keras

Kebutuhan perangkat keras yang digunakan dalam pembuatan sistem ini ialah:

**Tabel 4.1.** Kebutuhan perangkat keras

No	Komponen	Spesifikasi
1	PC atau laptop	Processor : Core i3 Ram : 4,00 GB

2. Kebutuhan Perangkat lunak

Kebutuhan Perangkat lunak yang digunakan dalam pembuatan sistem ini ialah:

**Tabel 4.2.** Kebutuhan perangkat lunak

No	Komponen	Kategori
1.	Python	Perangkat lunak
2.	Visual studio code	Perangkat lunak
3.	Git bash	Perangkat lunak
4.	Google chrome	Perangkat lunak

### 4.2.3. Perancangan Sistem

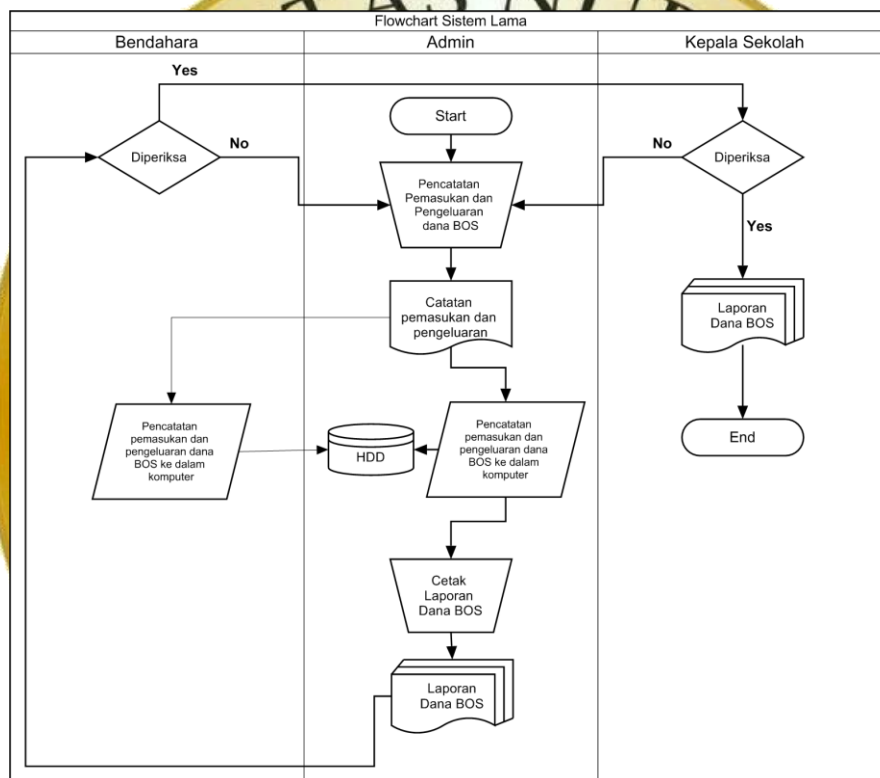
Dalam penelitian ini perancangan sistem yang digunakan terdiri dari beberapa bagian, yaitu bagan alir sistem (*Flowchart*), *Data Flow Diagram*

(DFD), Entity Relationship Diagram(ERD), serta dilengkapi dengan desain database dan desain program.

a. Bagan Alir sistem (Flowchart)

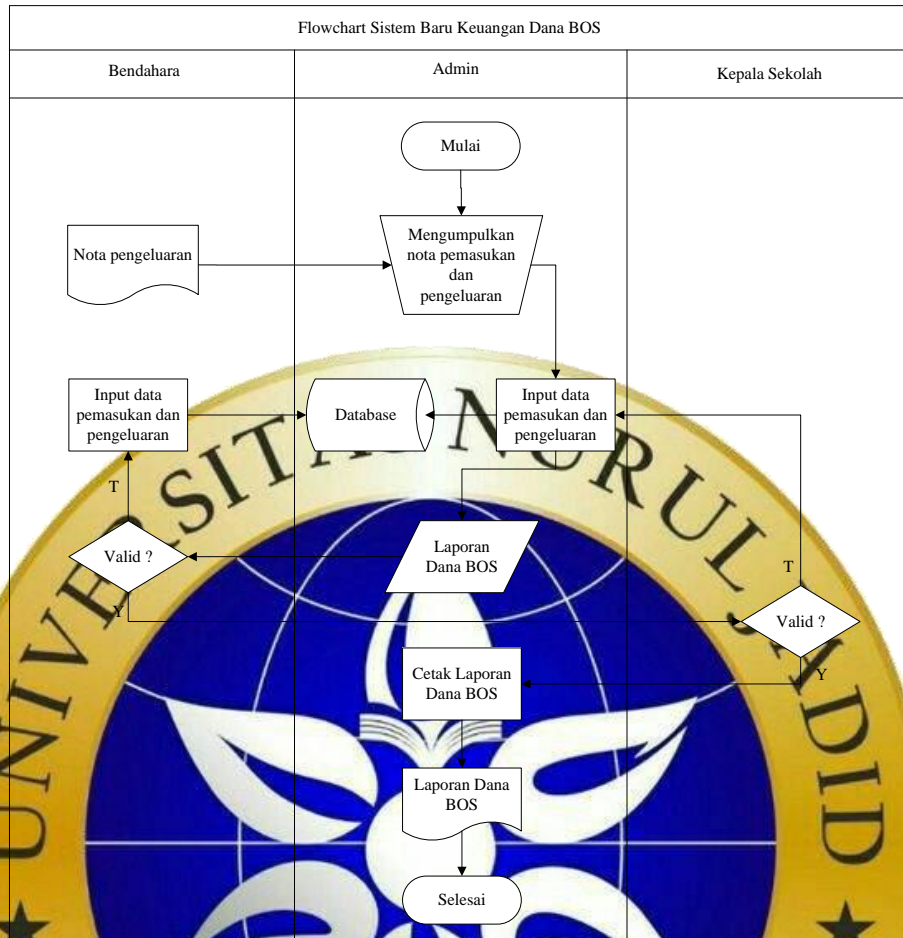
Bagan alir sistem pada sistem ini menggambarkan secara keseluruhan mengenai alur sistem yang digunakan untuk menyelesaikan permasalahan yang terjadi dan untuk mendapatkan gambaran yang lebih jelas. Berikut merupakan gambar flowchart terlihat pada gambar dibawah ini.

1. Flowchart sistem lama



Gambar 4.1 Flowchart Sistem Lama

## 2. Flowchart sistem baru



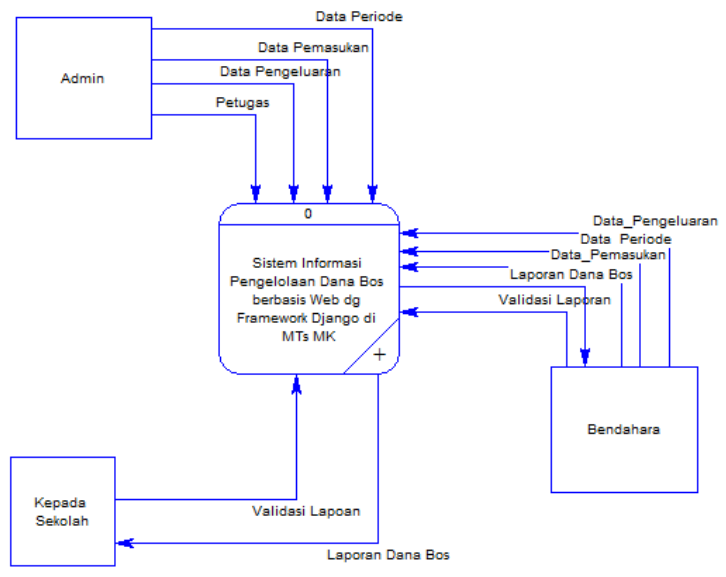
**Gambar 4.2** Flowchart Sistem Baru

### b. Data Flow Diagram (DFD)

Data Flow Diagram merupakan suatu gambaran aliran data dari sebuah sistem yang akan dirancang untuk mendapatkan informasi yang dibutuhkan. Dalam Data Flow Diagram menampilkan siapa saja yang terlibat dengan sistem tersebut mulai dari awal hingga akhir. Dalam pembuatan Data Flow Diagram (DFD) terdiri dari beberapa level diantaranya sebagai berikut:

#### 1. Diagram context

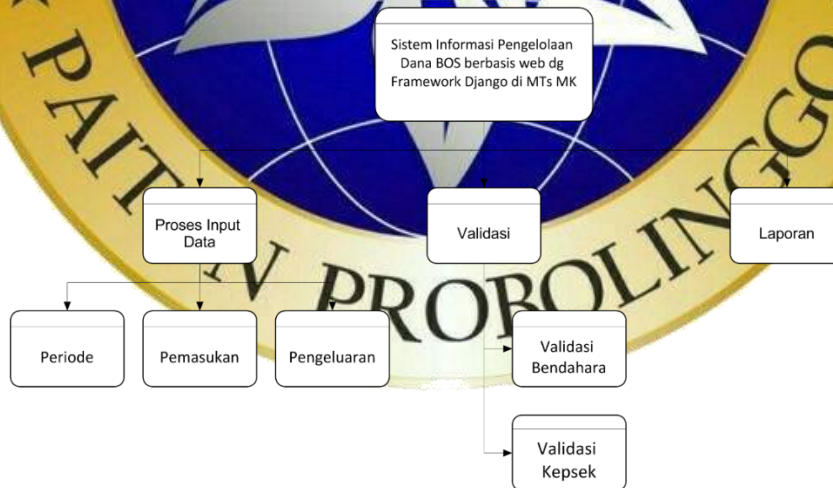
Diagram konteks atau yang disebut juga dengan level 0 dan merupakan penentu utama dalam menjelaskan suatu sistem. Diagram konteks adalah diagram sederhana yang hanya mempunyai satu proses dengan menentukan sistem secara keseluruhan. Berikut merupakan diagram konteks dari sistem pengelolaan Dana BOS.



Gambar 4.3. Diagram Context

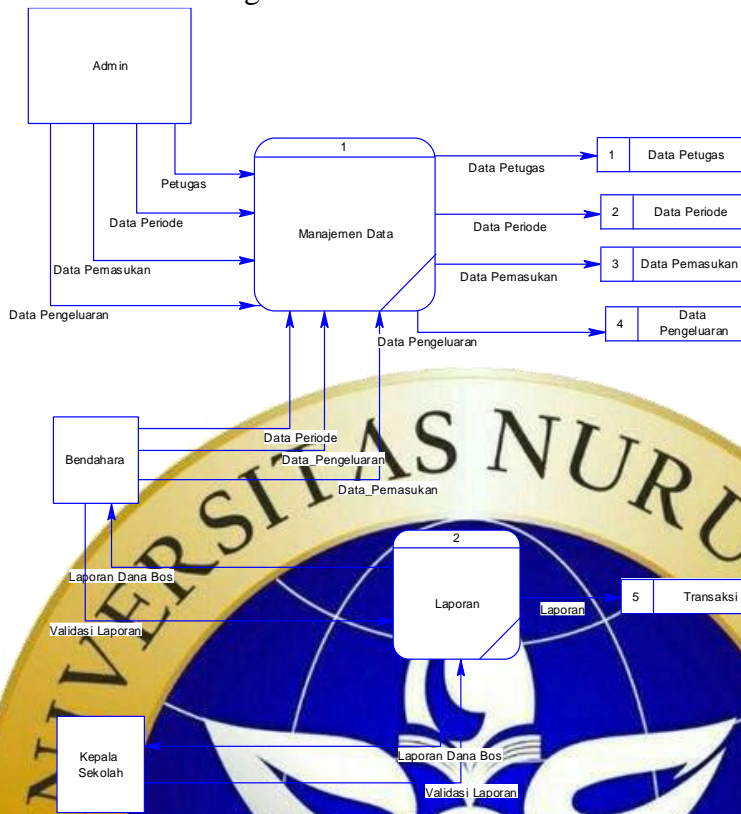
## 2. Bagan Berjenjang

Bagan berjenjang merupakan suatu bagan yang menggambarkan segala proses yang ada, digunakan untuk mempermudah dalam pembuatan sistem dengan jelas dan tersusun.



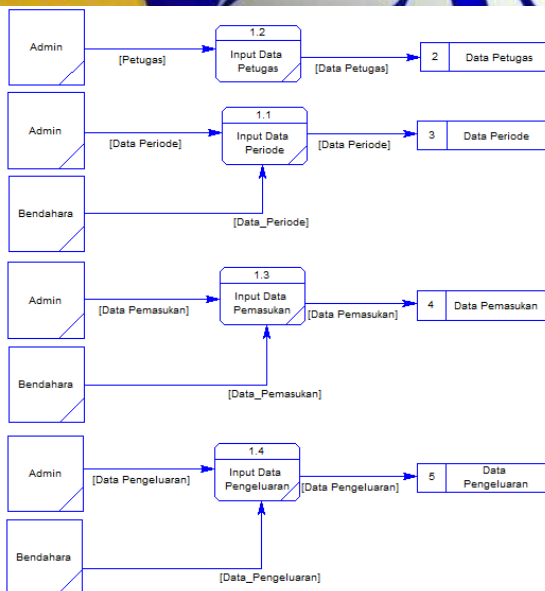
Gambar 4.4 Bagan berjenjang

### 3. Data Flow Diagram Level 1



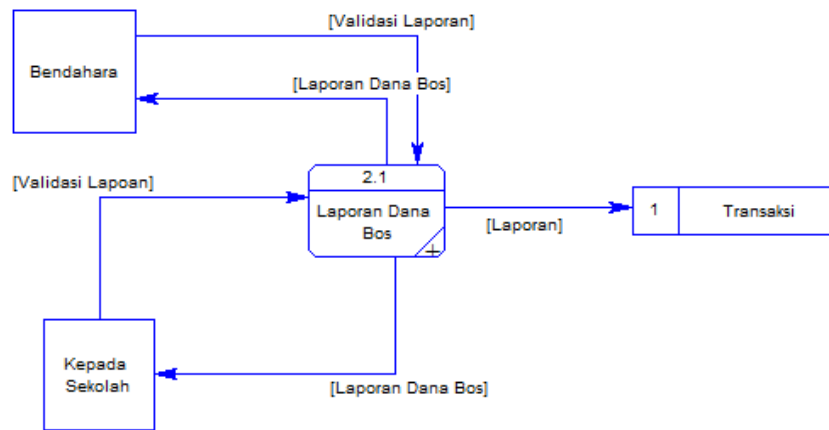
Gambar 4.5 DFD Level 1

### 4. Data Flow Diagram Level 2 Proses 1



Gambar 4.6 DFD Level 2 proses 1

5. Data Flow Diagram Level 2 Proses 2



Gambar 4.7 DFD Level 2 Proses 2

c. Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* (ERD) merupakan suatu diagram yang digunakan untuk menunjukkan relasi antar entitas ataupun objek. Berikut ini merupakan gambaran *Entity Relationship Diagram* (ERD) dalam perancangan sistem.



Gambar 4.8 ERD



Kamus Data :

Petugas	: { *id, user, nip, nama, level }
Periodik	: { *id, periode, tahun, kepek, bendahara }
Pemasukan	: { *id, tgl, periode, kode, nobukti, uraian, pemasukan }
Pengeluaran	: { *id, tgl, periode, kode, nobukti, bukti_pic, uraian, pengeluaran }
Transaksi	: { *id, periode, pemasukan, pengeluaran }

d. Desain Database

Tahapan desain database merupakan tahap mendesain database hasil perancangan dari entity relationship diagram yang kemudian ditentukan struktur dari tabel berupa kolom, tipe data, kunci utama dan lain-lain. Berikut adalah desain database:

**Tabel 4.3** Petugas

No.	Fields	Tipe Data	Keterangan
1	Id	Integer	Primary Key
2	User	-	Forgein Key
3	Nip	CharField	
4	Nama	CharField	
5	Level	CharField	

**Tabel 4.4** Periode

No.	Fields	Tipe Data	Keterangan
1	Id	Integer	Primary Key
2	Periode	CharField	Forgein Key
3	Tahun	CharField	
4	Kepsek	Boolean	
5	Bendahara	Boolean	

**Tabel 4.5 Pemasukan**

No.	Fields	Tipe Data	Keterangan
1	Id	Integer	Primary Key
2	Tgl	Date	
3	Periode		Forgein Key
4	Kode	CharField	
5	Nobukti	Integer	
6	Uraian	CharField	
7	Pemasukan	Integer	

**Tabel 4.6 Pengeluaran**

No.	Fields	Tipe Data	Keterangan
1	Id	Integer	Primary Key
2	Tgl	Date	
3	Periode		Forgein Key
4	Kode	CharField	
5	Nobukti	Integer	
6	Bukti_pic	Image	
7	Uraian	CharField	
8	Pengeluaran	Integer	

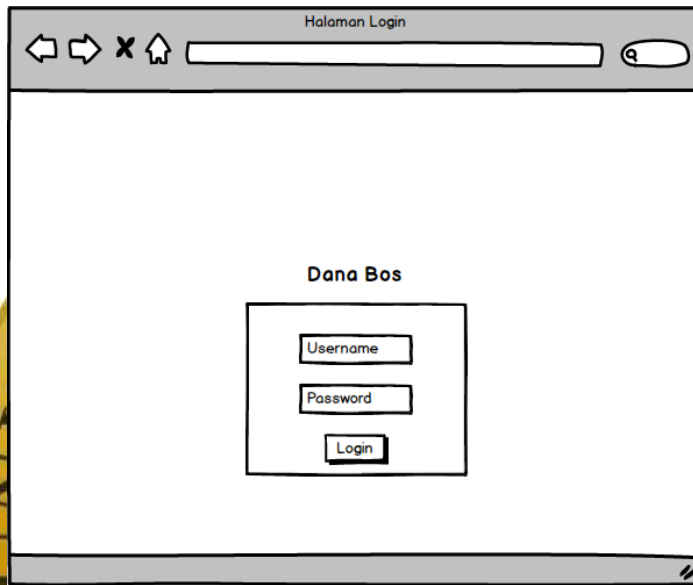
**Tabel 4.7 Transaksi**

No.	Fields	Tipe Data	Keterangan
1	Id	Integer	Primary Key
2	Periode		Forgein Key
3	Pemasukan		Forgein Key
4	Pengeluaran		Forgein Key

e. Desain Interface

Desain *interface* adalah suatu gambaran atau tampilan yang sudah dirancang pada pembuatan program aplikasi. Hal ini bertujuan agar pemrograman tidak terjadi kesalahan dan berkesinambungan dari hasil analisa yang telah ada.

1. Halaman Login



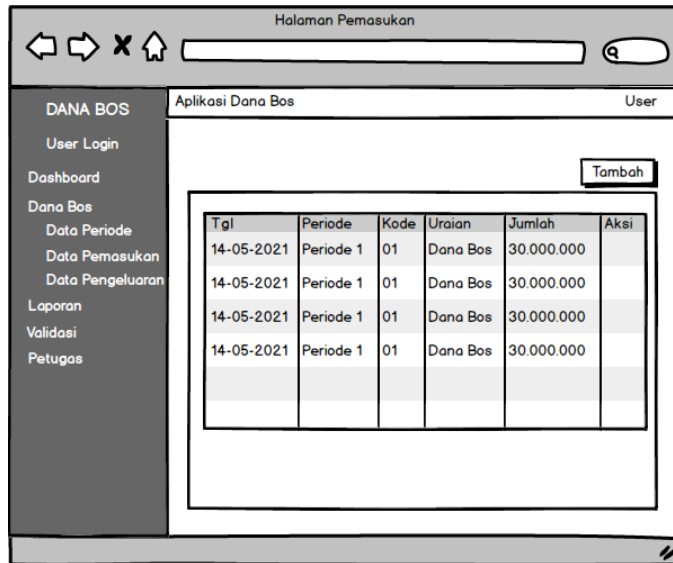
Gambar 4.9 Halaman Login

2. Halaman Dashboard



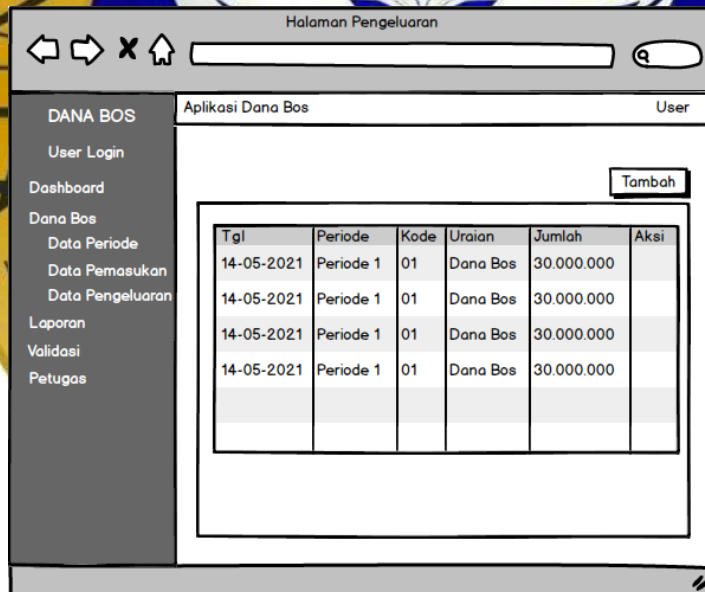
Gambar 4.10 Halaman Dashboard

3. Halaman pemasukan



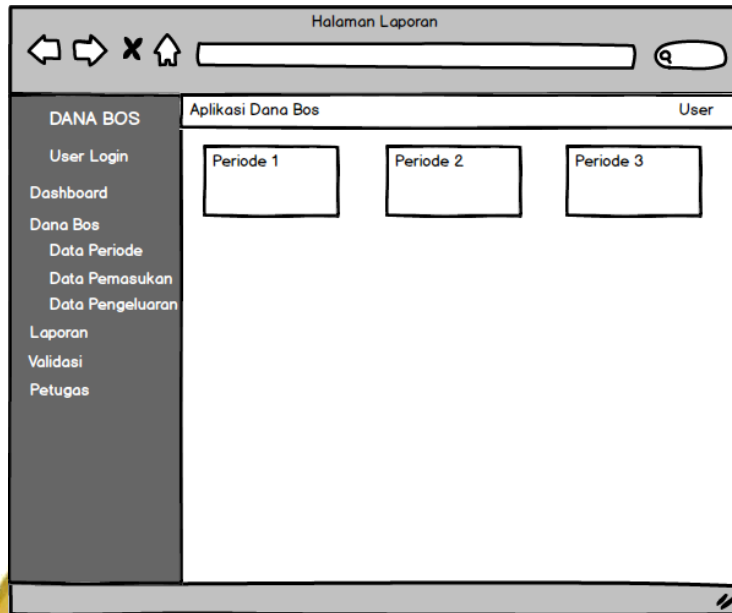
Gambar 4.11 Halaman pemasukan

4. Halaman pengeluaran



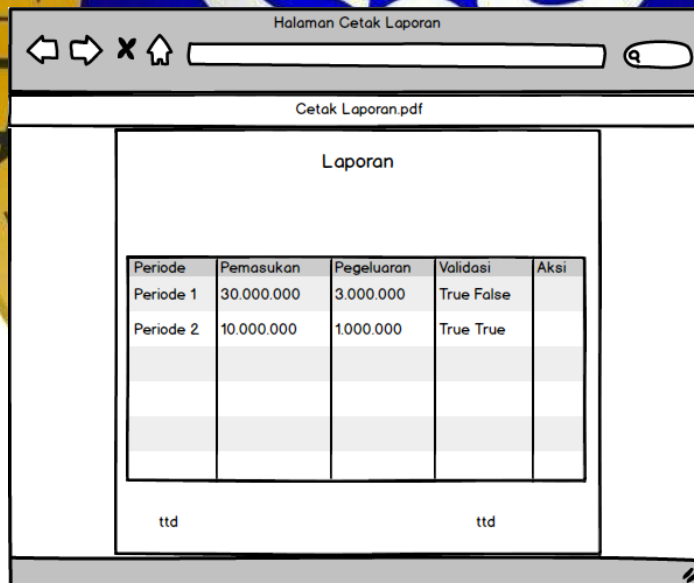
Gambar 4.12 Halaman pengeluaran

5. Halaman laporan



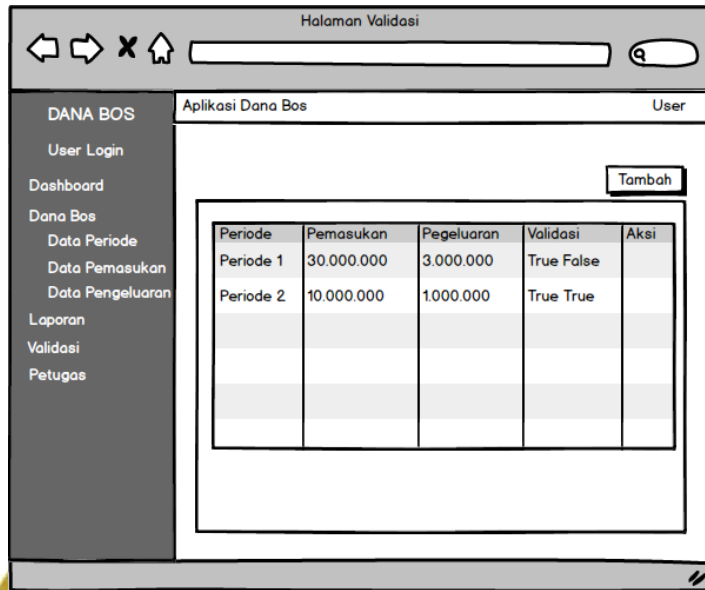
Gambar 4.13 Halaman laporan

6. Halaman cetak laporan



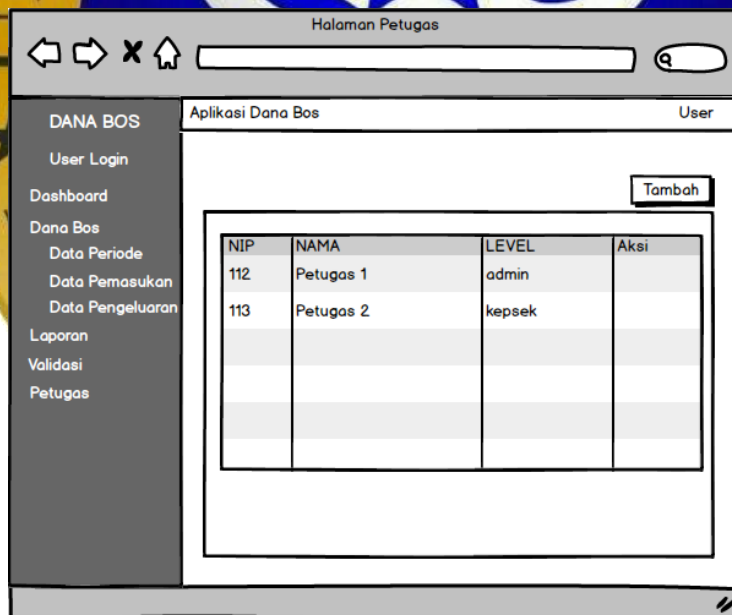
Gambar 4.14 Halaman cetak laporan

7. Halaman validasi



Gambar 4.15 Halaman validasi

8. Halaman data petugas



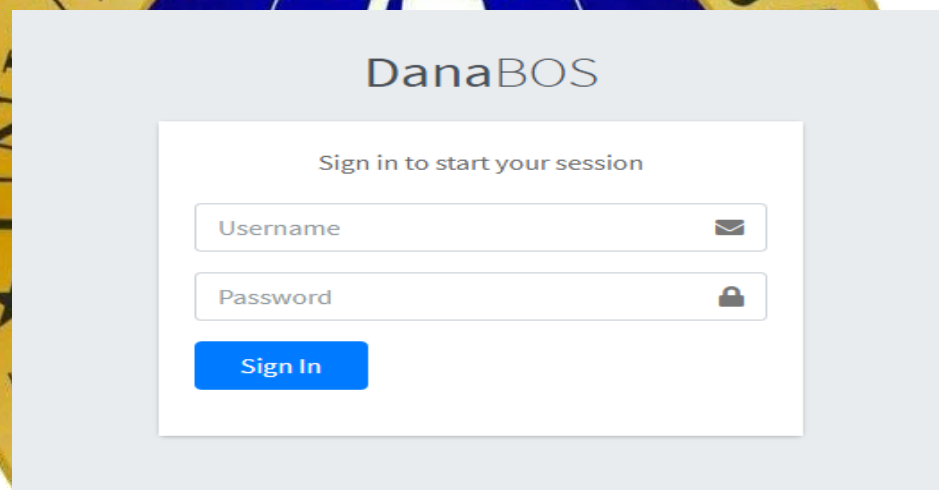
Gambar 4.16 Halaman data petugas

#### 4.2.4. Implementasi program

Setelah menganalisa desain yang telah dibuat, maka tahap selanjutnya adalah menganalisis atau mengimplementasikan sistem yang terdapat pada sistem informasi pengelolaan dana bantuan operasional sekolah (BOS). Berikut ini adalah hasil implementasi terhadap sistem yang telah dibuat:

##### 1. Halaman Login

Halaman login digunakan oleh pengguna untuk mengakses aplikasi dengan memasukkan nama pengguna serta kata sandi. Jika pengguna berhasil masuk, maka pengguna akan diarahkan ke halaman utama aplikasi, jika pengguna gagal masuk maka pengguna tidak dapat menuju halaman utama. Di bawah ini merupakan gambar halaman login.



Gambar 4.17 Halaman Login

##### Segmen Program 4.1 Views Halaman Login

```
def loginPage(request):  
    if request.method == 'POST':  
        username = request.POST.get('username')  
        password = request.POST.get('password')  
  
        cocokan = authenticate(request, username=username,  
                               password=password)  
        if cocokan is not None:  
            login(request, cocokan)  
            return redirect('home')
```

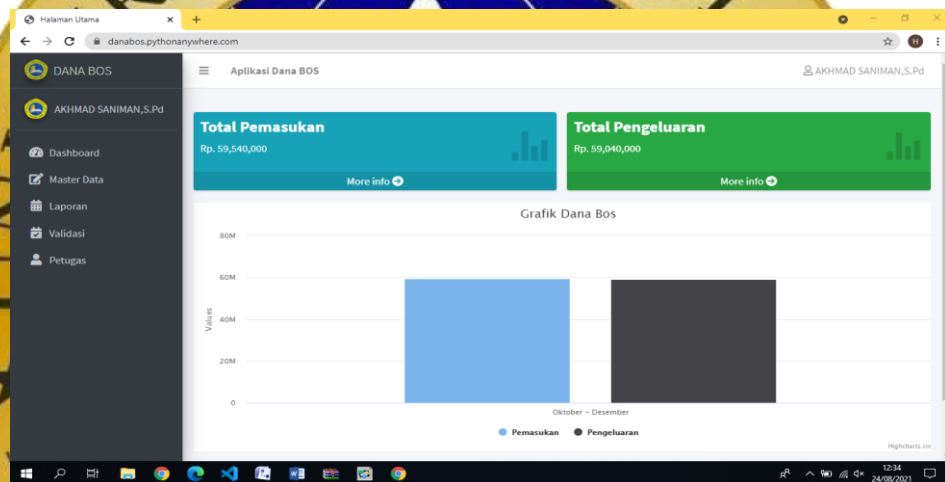
```

else:
    messages.success(request, 'Username/Password Salah')
context = {
    'judul': 'Halaman Login',
    'menu': 'login',
}
return render(request, 'data/login.html', context)

```

## 2. Halaman Dashboard Admin/Bendahara

Setelah pengguna berhasil masuk melalui login maka akan menampilkan halaman utama. Pada halaman utama akan menampilkan jumlah total pemasukan dan total pengeluaran yang dilakukan menggunakan aplikasi ini, berikut merupakan gambar halaman utama.



Gambar 4.18 Halaman Dashboard Admin

### Segmen Program 4.2 Views Halaman Dashboard Admin

```

@login_required(login_url='login')
@pilihan_login
def home(request):
    masuk =
    Transaksi.objects.all().aggregate(Sum('pemasukan__pemasukan'))
    keluar =
    Transaksi.objects.all().aggregate(Sum('pengeluaran__pengeluaran'))
    grafik =
    Transaksi.objects.values('periode__periode').annotate(total_masuk_cou
nt=Sum('pemasukan__pemasukan')).annotate(total_keluar_count=Sum(
'pengeluaran__pengeluaran'))

context = {
    'judul': 'Halaman Utama',

```



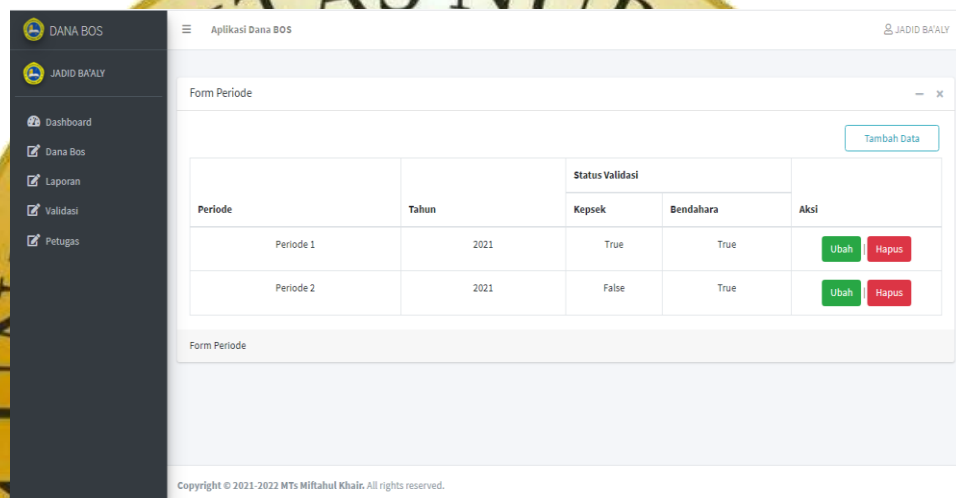
```

'masuk': masuk,
'keluar': keluar,
'grafik': grafik
}
return render(request, 'data/dashboard.html', context)

```

### 3. Halaman Data Periode

Pada halaman periode akan ditampilkan data-data periode pada laporan. Terdapat fitur berupa tambah data, ubah, dan hapus data periode. Berikut merupakan gambar tampilan halaman periode.



**Gambar 4.19** Halaman Data Periode  
**Segmen Program 4.3** Views Halaman Data Periode

```

@login_required(login_url='login')
def periode(request):
    periode = Periodik.objects.all()
    context = {
        'judul': 'Form Periode',
        'periode': periode
    }
    return render(request, 'data/periode.html', context)

```

```

@login_required(login_url='login')
def tambahperiode(request):
    formperiode = PeriodeForm()
    if request.method == 'POST':
        formsimpan = PeriodeForm(request.POST)
        if formsimpan.is_valid():
            formsimpan.save()
            return redirect('/periode/')

```

```

context = {
    'judul': 'Form Periode',
    'form': formperiode,
}
return render(request, 'data/input_form.html', context)

```

```

@login_required(login_url='login')
def ubahperiode(request, pk):
    periode = Periodik.objects.get(id=pk)
    formperiode = PeriodeForm(instance=periode)
    if request.method == 'POST':
        formedit = PeriodeForm(request.POST, instance=periode)
        if formedit.is_valid():
            formedit.save()
            return redirect('/periode/')
    context = {
        'judul': 'Form Ubah Dana Masuk',
        'form': formperiode,
    }
    return render(request, 'data/input_form.html', context)
@login_required(login_url='login')
def hapusperiode(request, pk):
    hapusperiode = Periodik.objects.get(id=pk)
    hapusperiode.delete()
    return redirect('periode')

```

#### 4. Halaman Data Pemasukan

Pada halaman ini menampilkan data berupa tabel yang menampilkan data pemasukan, pada bagian samping tabel ada tombol untuk menambah data dan terdapat aksi ubah dan hapus data pemasukan. Berikut ini merupakan gambar tampilan halaman data pemasukan.

Tgl	Periode	Kode	Uraian	Jumlah	Aksi
July 29, 2021, 4:21 a.m.	Periode 1, 2021	21	Dana Bos	Rp. 30,000,000	Ubah Hapus
July 29, 2021, 4:21 a.m.	Periode 2, 2021	431	Dana Bos	Rp. 40,000,000	Ubah Hapus

**Gambar 4.20** Halaman Data Pemasukan

#### Segmen Program 4.4 Views Halaman Data Pemasukan

```
@login_required(login_url='login')
def pemasukan(request):
    masuk = Pemasukan.objects.all()
    total = Pemasukan.objects.all().aggregate(Sum('pemasukan'))
    context = {
        'judul': 'Pemasukan Dana Bos',
        'Danamasuk': masuk,
        'totalmasuk': total
    }
    return render(request, 'data/pemasukan.html', context)
```

```
@login_required(login_url='login')
def tambahmasuk(request):
    formmasuk = MasukForm()
    if request.method == 'POST':
        formsimpan = MasukForm(request.POST)
        if formsimpan.is_valid():
            debit = formsimpan.save()
            Transaksi.objects.create(
                pemasukan=debit,
                periode=debit.periode
            )
        return redirect('/pemasukan/')
    context = {
        'judul': 'Form Dana Masuk',
        'form': formmasuk,
    }
    return render(request, 'data/input_form.html', context)
```

```
@login_required(login_url='login')
def ubahmasuk(request, pk):
    masuk = Pemasukan.objects.get(id=pk)
    formmasuk = MasukForm(instance=masuk)
    if request.method == 'POST':
        formedit = MasukForm(request.POST, instance=masuk)
        if formedit.is_valid():
            formedit.save()
            return redirect('/pemasukan/')
    context = {
        'judul': 'Form Ubah Dana Masuk',
        'form': formmasuk,
    }
    return render(request, 'data/input_form.html', context)
```

```

@login_required(login_url='login')
def hapusmasuk(request, pk):
    hapusmasuk = Pemasukan.objects.get(id=pk)
    hapusmasuk.delete()

    return redirect('pemasukan')

```

## 5. Halaman Pengeluaran

Pada halaman ini menampilkan data berupa tabel yang menampilkan data pengeluaran, pada bagian samping tabel ada tombol untuk menambah data dan terdapat aksi ubah dan hapus data pengeluaran. Berikut ini merupakan gambar tampilan halaman pengeluaran.

The screenshot displays the 'Pengeluaran Dana Bos' application. It features a sidebar with navigation options: Dashboard, Dana Bos, Laporan, Validasi, and Petugas. The main content area shows a table of expenses with the following data:

Tgl	Periode	Kode	No. Bukti	Uraian	Jumlah	Aksi
July 29, 2021, 4:20 a.m.	Periode 1, 2021	1	1	HR Guru	Rp. 3,000,000	Ubah Hapus
July 29, 2021, 4:20 a.m.	Periode 2, 2021	431	1	HR Guru	Rp. 8,000,000	Ubah Hapus
July 29, 2021, 4:22 a.m.	Periode 1, 2021	23	3	iii	Rp. 4,400,000	Ubah Hapus
July 29, 2021, 4:22 a.m.	Periode 2, 2021	21	3	hdh	Rp. 5,500,000	Ubah Hapus

Additional details from the screenshot: Total Pengeluaran : Rp. 20,900,000. A 'Tambah Data' button is visible above the table. The footer contains the copyright notice: Copyright © 2021-2022 MTs Miftahul Khair. All rights reserved.

**Gambar 4.21** Halaman Data Pengeluaran  
**Segmen Program 4.5 Views** Halaman Data Pengeluaran

```

@login_required(login_url='login')
def pengeluaran(request):
    keluar = Pengeluaran.objects.all()
    total = Pengeluaran.objects.all().aggregate(Sum('pengeluaran'))
    context = {
        'judul': 'Pengeluaran Dana Bos',
        'Danakeluar': keluar,
        'totalkeluar': total
    }
    return render(request, 'data/pengeluaran.html', context)

```

```

@login_required(login_url='login')
def tambahkeluar(request):
    formkeluar = KeluarForm()
    if request.method == 'POST':

```

```

formsimpan = KeluarForm(request.POST, request.FILES)
if formsimpan.is_valid:
    kredit = formsimpan.save()
    Transaksi.objects.create(
        pengeluaran=kredit,
        periode=kredit.periode
    )

    return redirect('/pengeluaran/')
context = {
    'judul': 'Form Dana Masuk',
    'form': formkeluar,
}
return render(request, 'data/input_form.html', context)

@login_required(login_url='login')
def ubahkeluar(request, pk):
    keluar = Pengeluaran.objects.get(id=pk)
    formkeluar = KeluarForm(instance=keluar)
    if request.method == 'POST':
        formedit = KeluarForm(request.POST, request.FILES,
            instance=keluar)
        if formedit.is_valid:
            formedit.save()
            print (formedit)
            return redirect('/pengeluaran/')
    context = {
        'judul': 'Form Ubah Dana Keluar',
        'form': formkeluar,
    }
    return render(request, 'data/input_form.html', context)

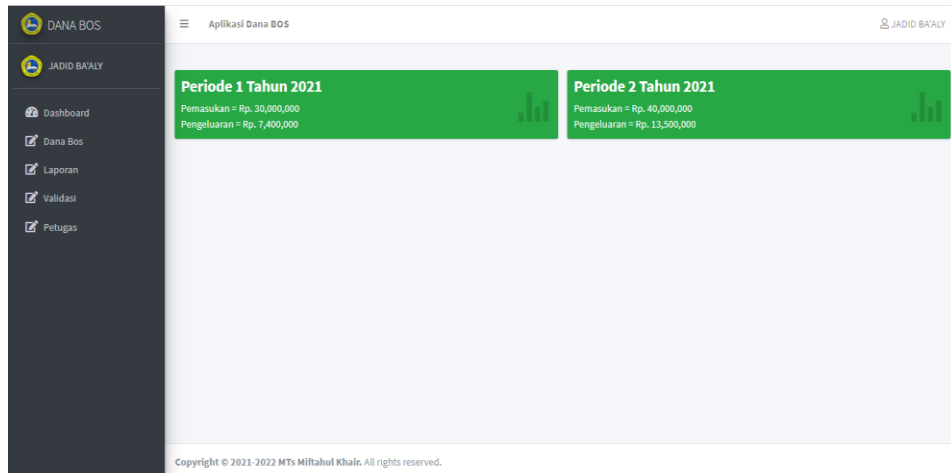
@login_required(login_url='login')
def hapuskeluar(request, pk):
    hapuskeluar = Pengeluaran.objects.get(id=pk)
    hapuskeluar.delete()

    return redirect('pengeluaran')

```

#### 6. Halaman Daftar Periode Laporan

Pada halaman daftar periode akan ditampilkan data periode pemasukan dan pengeluaran per tiga bulan. Berikut merupakan gambar tampilan halaman daftar periode laporan.



**Gambar 4.22** Halaman Daftar Periode Laporan  
**Segmen Program 4.6** Views Halaman Daftar Periode laporan

```

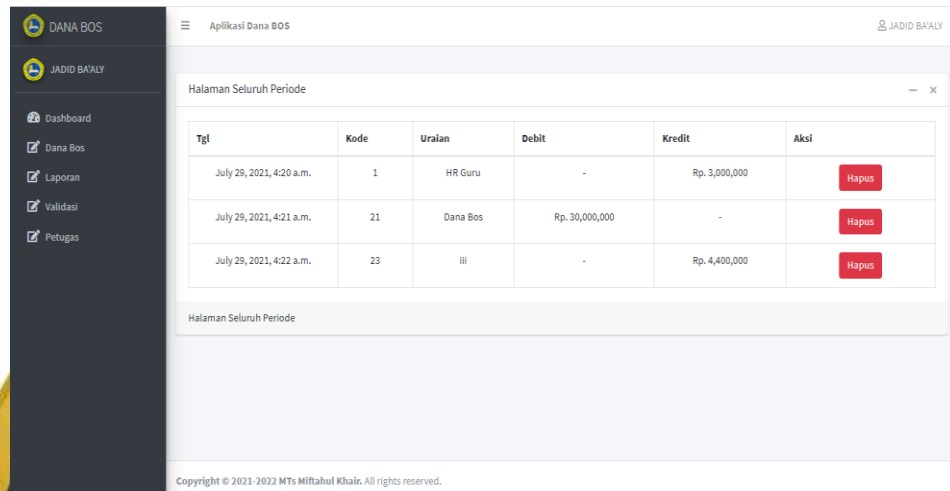
@login_required(login_url='login')
def periodebos(request):
    periode = Periodik.objects.all()
    grafik = Transaksi.objects.values('periode__periode',
    'periode__tahun', 'periode__kepsek',
    'periode__bendahara').annotate(total_masuk_count=Sum('pemasukan__pe
    masukan')).annotate(total_keluar_count=Sum('pengeluaran__pengeluaran'
    ))
    # print (grafik)
    context = {
        'judul': 'Halaman Seluruh Periode',
        'periode': grafik
    }
    return render(request, 'data/periodebos.html', context)

@login_required(login_url='login')
def lapperiodebos(request, pk):
    periode = Periodik.objects.get(id=pk)
    laporan = periode.transaksi_set.all()
    context = {
        'judul': 'Halaman Seluruh Periode',
        'laporan': laporan
    }
    return render(request, 'data/lapperiodebos.html', context)

```

## 7. Halaman Laporan perperiode

Pada halaman ini menampilkan data pemasukan dan pengeluaran per periode atau per tiga bulan. Terdapat fitur berupa hapus data. Berikut merupakan gambar tampilan halaman laporan periode.



Tgl	Kode	Uraian	Debit	Kredit	Aksi
July 29, 2021, 4:20 a.m.	1	HR Guru	-	Rp. 3,000,000	Hapus
July 29, 2021, 4:21 a.m.	21	Dana Bos	Rp. 30,000,000	-	Hapus
July 29, 2021, 4:22 a.m.	23	iii	-	Rp. 4,400,000	Hapus

**Gambar 4.23** Halaman Laporan perperiode  
**Segmen Program 4.7** Views Halaman Laporan Perperiode

```
@login_required(login_url='login')
def periode(request):
    periode = Periodik.objects.all()
    context = {
        'judul': 'Form Periode',
        'periode': periode
    }
    return render(request, 'data/periode.html', context)

@login_required(login_url='login')
def tambahperiode(request):
    formperiode = PeriodeForm()
    if request.method == 'POST':
        formsimpan = PeriodeForm(request.POST)
        if formsimpan.is_valid():
            formsimpan.save()
            return redirect('/periode/')
    context = {
        'judul': 'Form Periode',
        'form': formperiode,
    }
    return render(request, 'data/input_form.html', context)
```

```

@login_required(login_url='login')
def ubahperiode(request, pk):
    periode = Periodik.objects.get(id=pk)
    formperiode = PeriodeForm(instance=periode)
    if request.method == 'POST':
        formedit = PeriodeForm(request.POST, instance=periode)
        if formedit.is_valid():
            formedit.save()
            return redirect('/periode/')
    context = {
        'judul': 'Form Ubah Dana Masuk',
        'form': formperiode,
    }
    return render(request, 'data/input_form.html', context)

```

```

@login_required(login_url='login')
def hapusperiode(request, pk):
    hapusperiode = Periodik.objects.get(id=pk)
    hapusperiode.delete()
    return redirect('periode')

```

#### 8. Halaman Validasi Laporan

Pada halaman ini menampilkan data pengelolaan dana bos per periode yang dilengkapi aksi validasi yang bisa dijalankan oleh kepala sekolah dan bendahara. Dimana admin bisa mencetak laporan jika laporan sudah divalidasi pada halaman ini juga dilengkapi aksi lihat bukti kwintansi. Berikut merupakan gambar tampilan halaman validasi laporan.

Periode	Pemasukan	Pengeluaran	Validasi		Aksi
			Kepsek	Bendahara	
Periode 1 Tahun 2021	Rp. 30,000,000	Rp. 7,400,000	True	True	Validasi Lihat Cetak
Periode 2 Tahun 2021	Rp. 40,000,000	Rp. 13,500,000	False	True	Validasi Lihat Cetak

Gambar 4.24 Halaman Validasi Laporan



### Segmen Program 4.8 Views Halaman Validasi Laporan

```
@login_required(login_url='login')
def halval(request):
    periode = Periodik.objects.all()
    grafik = Transaksi.objects.values('periode__id','periode__periode',
'periode__tahun', 'periode__kepsek',
'periode__bendahara').annotate(total_masuk_count=Sum('pemasukan__pe
masuk')).annotate(total_keluar_count=Sum('pengeluaran__pengeluaran'
)).annotate(total_masuk_count=Sum('pemasukan__pemasukan'))
    context = {
        'judul': 'Halaman Validasi Laporan',
        'periode': grafik,
    }
    return render(request, 'data/validasi.html', context)

@login_required(login_url='login')
def halvalb(request):
    periode = Periodik.objects.all()
    grafik = Transaksi.objects.values('periode__id','periode__periode',
'periode__tahun', 'periode__kepsek',
'periode__bendahara').annotate(total_masuk_count=Sum('pemasukan__pe
masuk')).annotate(total_keluar_count=Sum('pengeluaran__pengeluaran'
))
    context = {
        'judul': 'Halaman Validasi Laporan',
        'periode': grafik,
    }
    return render(request, 'data/validasib.html', context)

@login_required(login_url='login')
def ubahvalidasi(request, pk):
    periode = Periodik.objects.get(id=pk)
    formperiode = UBPeriodeForm(instance=periode)
    if request.method == 'POST':
        formedit = UBPeriodeForm(request.POST, instance=periode)
        if formedit.is_valid():
            formedit.save()
            return redirect('/Halvalidasib/')
    context = {
        'judul': 'Form Ubah Petugas',
        'form': formperiode,
    }
    return render(request, 'data/input_form.html', context)
```

## 9. Halaman Cetak Laporan

Saat pengguna mencetak surat maka akan muncul tampilan dokumen berupa PDF. Laporan yang dihasilkan berdasarkan format laporan LPJ (Laporan Pertanggung Jawaban) Dana Bos.

The screenshot shows a PDF document titled "LAPORAN DANA BOS". At the top, it lists the organization: "Nama Sekolah: MTs MIFTAHUL KHAWAR", "Desa/Kelurahan: Alastengah Benua", "Kabupaten/Kota: Probolinggo", and "Provinsi: Jawa Timur". Below this is a table with columns: "Tgl", "No. Kode", "No. Bukti", "Uraian", "Pembelian (Debit)", "Penghasilan (Kredit)", and "Saldo". The table contains three rows of transactions and a summary row for "Jumlah Bulan Ini".

Tgl	No. Kode	No. Bukti	Uraian	Pembelian (Debit)	Penghasilan (Kredit)	Saldo
July 29, 2021: 4:20 a.m.	1	1	HR Guru	-	Rp. 3.000.000	
July 29, 2021: 4:21 a.m.	21	0	Dana Bos	Rp. 30.000.000	-	
July 29, 2021: 4:22 a.m.	23	3	ii	-	Rp. 4.400.000	
Jumlah Bulan Ini				Rp. 30.000.000	Rp. 7.400.000	Rp. 22.600.000

Below the table, there is a section for "Saldo Buku Kas Umum" and "Terdiri dari" with a list of items. At the bottom, there are fields for "Mengetahui, Kepala Sekolah" (signed by SAYID MUHAMMAD, S.Pd) and "Probolinggo, Bendahara" (signed by JACID BAHLY).

**Gambar 4.25** Halaman Cetak Laporan  
**Segmen Program 4.9** Views Halaman Cetak Laporan

```
def pdf_laporan(request, pk):
    periode = Periodik.objects.get(id=pk)
    laporan = periode.transaksi_set.all()
    masuk =
periode.transaksi_set.all().aggregate(Sum('pemasukan__pemasukan'))
    keluar =
periode.transaksi_set.all().aggregate(Sum('pengeluaran__pengeluaran'))

    grafik =
Transaksi.objects.filter(periode=periode).annotate(total_masuk_count=Sum('pemasukan__pemasukan')).annotate(total_keluar_count=Sum('pengeluaran__pengeluaran'))
    saldo = masuk['pemasukan__pemasukan__sum']-
keluar['pengeluaran__pengeluaran__sum']
    template_path = 'data/pdf_template.html'
    context = {'laporan': laporan, 'masuk': masuk, 'keluar': keluar, 'saldo':
saldo, 'periode': periode}
    html = template.render(context)

    pisa_status = pisa.CreatePDF(
        html, dest=response)
    if pisa_status.err:
        return HttpResponse('We had some errors <pre>' + html + '</pre>')
```

```
return response
```

```
def pdfbukti(request, pk):
```

```
    periode = Periodik.objects.get(id=pk)
```

```
    laporan = periode.pengeluaran_set.all()
```

```
    template_path = 'data/cetakbukti.html'
```

```
    context = {'laporan': laporan}
```

```
    response = HttpResponse(content_type='application/pdf')
```

```
    response['Content-Disposition'] = 'filename="report.pdf"'
```

```
    template = get_template(template_path)
```

```
    html = template.render(context)
```

```
    pisa_status = pisa.CreatePDF(
```

```
        html, dest=response)
```

```
    if pisa_status.err:
```

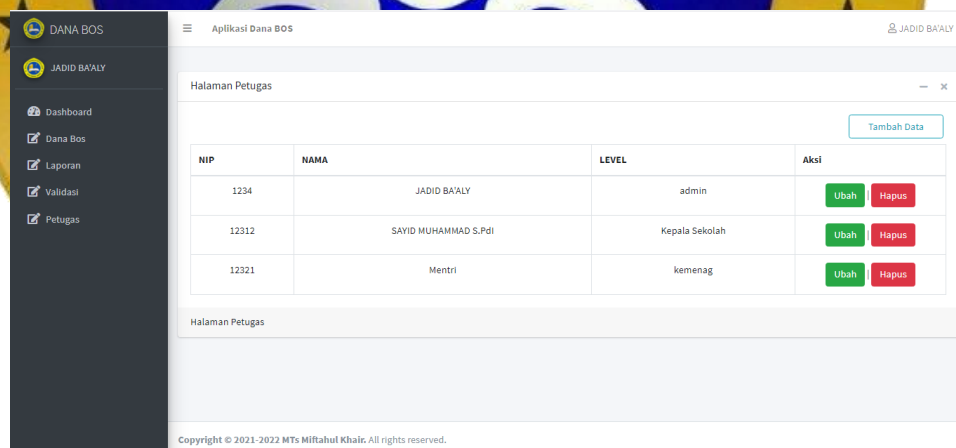
```
        return HttpResponse('We had some errors <pre>' + html + '</pre>')
```

```
    return response
```

#### 10. Halaman Data Petugas

Pada halaman ini digunakan pengguna untuk menambah data petugas sesuai dengan levelnya, terdapat fitur ubah dan hapus data petugas.

Berikut merupakan tampilan halaman data petugas.



**Gambar 4.26** Halaman Data Petugas

### Segmen Program 4.10 Views Halaman Data Petugas

```
@login_required(login_url='login')
def petugas(request):
    petugas = Petugas.objects.all()
    context = {
        'judul': 'Halaman Petugas',
        'petugas': petugas
    }
    return render(request, 'data/petugas.html', context)
```

```
@login_required(login_url='login')
def tambahpetugas(request):
    formpetugas = UserForm()
    if request.method == 'POST':
        formsimpan = UserForm(request.POST)
        if formsimpan.is_valid():
            petugas = formsimpan.save()
            grup = Group.objects.get(name='admin')
            petugas.groups.add(grup)
            Petugas.objects.create(
                user=petugas,
                nip=petugas.username,
                nama=petugas.first_name,
                level=petugas.last_name),
            return redirect('/petugas')
    context = {
        'judul': 'Form Petugas',
        'form': formpetugas,
    }
    return render(request, 'data/input_form.html', context)
```

```
@login_required(login_url='login')
def tambahkepek(request):
    formpetugas = UserForm()
    if request.method == 'POST':
        formsimpan = UserForm(request.POST)
        if formsimpan.is_valid():
            petugas = formsimpan.save()
            grup = Group.objects.get(name='kepek')
            petugas.groups.add(grup)
            Petugas.objects.create(
                user=petugas,
                nip=petugas.username,
                nama=petugas.first_name,
                level=petugas.last_name),
            return redirect('/petugas')
    context = {
```

```

        'judul': 'Form Petugas',
        'form': formpetugas,
    }
    return render(request, 'data/input_form.html', context)

```

```

@login_required(login_url='login')
def tambahbendahara(request):
    formpetugas = UserForm()
    if request.method == 'POST':
        formsimpan = UserForm(request.POST)
        if formsimpan.is_valid():
            petugas = formsimpan.save()
            grup = Group.objects.get(name='bendahara')
            petugas.groups.add(grup)
            Petugas.objects.create(
                user=petugas,
                mp=petugas.username,
                nama=petugas.first_name,
                level=petugas.last_name),
            return redirect('/petugas')
        context = {
            'judul': 'Form Petugas',
            'form': formpetugas,
        }
    return render(request, 'data/input_form.html', context)

```

```

@login_required(login_url='login')
def ubanpetugas(request, pk):
    petugas = Petugas.objects.get(id=pk)
    formpetugas = PetugasForm(instance=petugas)
    if request.method == 'POST':
        formedit = PetugasForm(request.POST, instance=petugas)
        if formedit.is_valid():
            formedit.save()
            return redirect('/petugas/')
    context = {
        'judul': 'Form Ubah Petugas',
        'form': formpetugas,
    }
    return render(request, 'data/input_form.html', context)

```

```

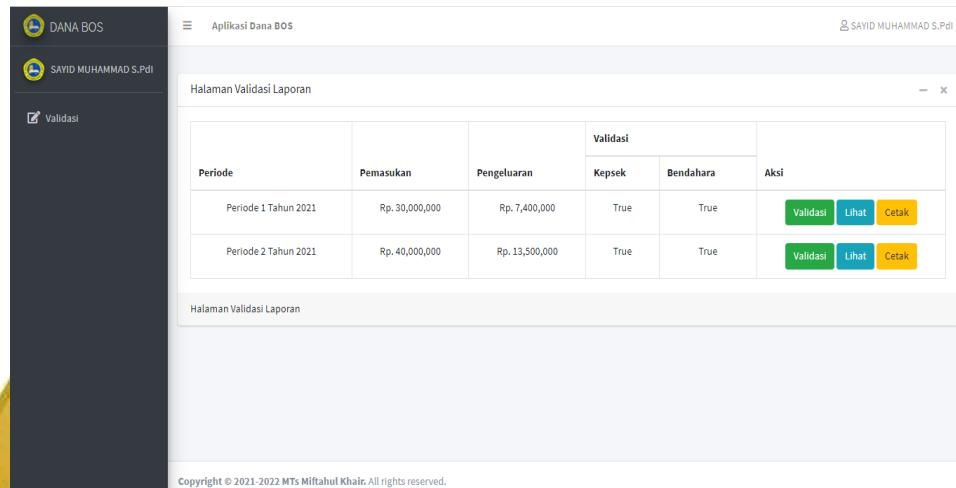
@login_required(login_url='login')
def hapuspetugas(request, pk):
    hapuspetugas = Petugas.objects.get(id=pk)
    hapuspetugas.delete()

    return redirect('petugas')

```

## 11. Halaman Dashboard Kepala Sekolah

Pada halaman ini menampilkan halaman dashboard kepala sekolah dimana kepala sekolah bisa melihat laporan periode dan bisa memvalidasi laporan yang telah dibuat oleh admin dan bendahara.



Periode	Pemasukan	Pengeluaran	Validasi		Aksi
			Kepsek	Bendahara	
Periode 1 Tahun 2021	Rp. 30,000,000	Rp. 7,400,000	True	True	Validasi Lihat Cetak
Periode 2 Tahun 2021	Rp. 40,000,000	Rp. 13,500,000	True	True	Validasi Lihat Cetak

Gambar 4.27 Halaman Dashboard Kepala Sekolah

### Segmen Program 4.11 Views Halaman Dashboard Kepala Sekolah

```
@login_required(login_url='login')
def halwalk(request):
    periode = Periodik.objects.all()
    grafik = Transaksi.objects.values('periode__id','periode__periode',
    'periode__tahun', 'periode__kepsek',
    'periode__bendahara').annotate(total_masuk_count=Sum('pemasukan__pe
    masukan')).annotate(total_keluar_count=Sum('pengeluaran__pengeluaran'
    ))
    # saldo = total_masuk_count - total_keluar_count
    # print (grafik)
    context = {
        'judul': 'Halaman Validasi Laporan',
        'periode': grafik,
        # 'saldo': saldo
    }
    return render(request, 'data/validasik.html', context)
```

## 4.3. Hasil Uji Coba

Pada proses pengujian ini, dilakukan untuk mengetahui apakah aplikasi yang telah dibuat sudah sesuai dengan kebutuhan pengguna. Adapun pengujian yang

dilakukan dalam penelitian ini meliputi pengujian internal dengan menggunakan metode *black box testing* dan pengujian eksternal dengan melakukan pengujian langsung kepada pihak instansi yang bersangkutan. Berikut ini adalah hasil pengujian yang telah dilakukan :

1. Hasil pengujian internal (*Black box testing*)

Pengujian ini akan dilakukan dengan menguji fungsional perangkat lunak dan hanya dilakukan oleh orang-orang yang ahli dibidangnya. Adapun pengujian internal pada sistem pengelolaan dana bantuan operasional sekolah adalah sebagai berikut :

**Tabel 4.8** Pengujian Internal

No.	Unit Uji	Masukan/ Kondisi	Hasil yang diharapkan	Kesesuaian		Ket.
				Ya	Tidak	
1	Login	Akses untuk menuju ke menu utama	- Jika username dan password admin, bendahara, dan kepek sesuai maka program akan melanjutkan ke menu utama.	V		
			- Jika tidak sesuai maka program tidak akan melanjutkan ke menu utama.			
2	Menu Master Data	Memilih Menu Periode	- Menampilkan data Periode.	V		
			- Dapat mengelolah data Periode.	V		
		Memilih Menu Pemasukan	- Menampilkan data Pemasukan.	V		
			- Dapat mengolah data Pemasukan.	V		

		Memilih Menu Pengeluaran	- Dapat menampilkan data Pengeluaran	V		
			- Dapat mengolah data Pegeluaran	V		
3	Menu Laporan	Memilih Periode	- Menampilkan laporan sesuai periode yang dipilih	V		
4	Menu Validasi	Validasi Laporan	- Kepala Sekolah dapat memvalidasi laporan	V		
			- Bendahara dapat memvalidasi laporan	V		
		Cetak Laporan	- Jika laporan sudah di validasi oleh bendahara dan kepala sekolah. Maka laporan dapat dicetak			
5	Menu Petugas		- Dapat Menampilkan data petugas	V		
			- Dapat Mengolah data Petugas	V		
6	Log Out		- Dapat melakukan logout	V		

## 2. Hasil pengujian eksternal

Pengujian eksternal dilakukan oleh user di MTs Miftahul Khair, dimana user akan mencoba menjalankan sistem kemudian mengisi kuesioner yang berupa pertanyaan tertulis.



**Tabel 4.9** Pengujian Eksternal

No	Pertanyaan	Nilai			
		SS	S	TS	STS
1	Sistem ini mempermudah pengelolaan dana bantuan operasional sekolah.	3	0	0	0
2	Sistem ini mempercepat pencarian data yang diperlukan.	3	0	0	0
3	Sistem ini meningkatkan efisiensi waktu dalam pembuatan laporan.	3	0	0	0
4	laporan yang dihasilkan oleh aplikasi ini telah sesuai format laporan Dana BOS.	3	0	0	0
5	Sistem ini memiliki fitur sesuai dengan kebutuhan pengguna.	2	1	0	0

Jumlah skor perolehan dari semua responden :

Jumlah Sangat Setuju (SS) =  $14 \times 4 = 56$

Jumlah Setuju (S) =  $1 \times 3 = 3$

Jumlah Tidak Setuju (TS) =  $0 \times 2 = 0$

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

Sehingga total perolehan skor dari semua responden adalah 59 .

Jumlah skor ideal diperoleh dari skor tertinggi x jumlah butir soal x jumlah responden =  $4 \times 5 \times 3 = 60$

Skor Presentase	Interpretasi
0 – 25 (%)	Tidak Baik
26 – 50 (%)	Kurang Baik
51 – 75 (%)	Baik
76 – 100 (%)	Sangat Baik

Rumus Penentuan Skor =  $(\text{Skor Perolehan} / \text{Skor Ideal}) \times 100$  Perhitungan  $(59 / 60) \times 100 = 82\%$  hasil perhitungan menggunakan metode skala likert didapatkan data sebanyak 82% yang artinya Sistem Informasi Pengelolaan

Dana Bantuan Operasional Sekolah (BOS) Berbasis Web Dengan Framework Django Di MTs Miftahul Khair ini sangat baik dan layak untuk digunakan.



