

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Pengumpulan Data

Hasil pengumpulan data pada penelitian ini diperoleh dari instrumen pengumpulan data, yaitu melakukan observasi dan wawancara kepada pihak yang terkait di *Corry Wedding* Kraksaan. Berikut pemaparan atas hasil dari pengumpulan data yang dilakukan.

4.1.1 Hasil Observasi

Berdasarkan observasi penelitian pada tanggal 20 maret 2021 yang dilakukan dengan melakukan pengamatan secara langsung Pada *Corry Wedding* yaitu mengamati beberapa kegiatan sistem berjalan yang berhubungan dengan penelitian, seperti proses pemesanan, pencatatan transaksi, dan promosi *wedding*, pada observasi ini menghasilkan bahwa pada pengolahan data, sistem pemesanan dan promosi jasa WO pada *Corry Wedding Organizer* masih bisa dikatakan belum begitu efektif dan efisien. Pemesanan jasa WO dilakukan menggunakan aplikasi *WhatsApp* dan juga Klien harus ke tempat WO untuk melakukan pemesanan. Dalam pencatatan transaksi masih dilakukan secara manual yaitu menggunakan buku. Sementara itu promosi *Wedding Organizer* masih dilakukan melalui media sosial. sedangkan permasalahan yang dialami oleh konsumen yaitu kurangnya informasi seperti harga, alamat tempat penyewaan, rute, jarak, serta belum tersedianya proses penyewaan perlengkapan *Wedding Organizer* di Kabupaten Probolinggo secara online dan mereka masih menghubungi WO satu-persatu dengan mencari terlebih dahulu informasi kesesuaian harga dan ketersediaan tanggal.

4.1.2 Hasil Wawancara

Adapun hasil wawancara yang dilaksanakan pada tanggal 20 maret 2021 di *Corry Wedding Organizer* kepada asisten *Corry Wedding* adalah sebagai berikut :

- a. *Corry Wedding Organizer* didirikan pada tahun 2018.

- b. Alur penyewaan pada *Corry Wedding* ialah pelanggan datang ke tempat vendor dan admin akan memberikan brosur paket pernikahan dan menjelaskan paket tersebut, selanjutnya admin memberikan formulir pemesanan untuk diisi oleh pelanggan, setelah mengisi formulir pelanggan menyerahkan formulir tersebut kepada admin, jika sudah lengkap admin akan mencatat data pemesanan di buku, dan memberikan kwitansi kepada pelanggan.
- c. Pelayanan yang ditawarkan ialah paket pernikahan, make up, pakaian/gaun, dekorasi, dan catering.
- d. Metode pembayaran bisa menggunakan cash atau transfer, pembayaran dilakukan pada awal pemesanan yaitu membayar DP terlebih dahulu selanjutnya membayar lunas setelah selesai acara.
- e. Pencatatan transaksi masih menggunakan buku.
- f. Kendala yang sering terjadi ialah data pelanggan yang belum terkomputerisasi, media promosi hanya dilakukan melalui media sosial.

4.2 Hasil Analisis dan Desain

Data yang diperoleh dari hasil observasi pada *Corry Wedding Organizer* dapat digunakan sebagai referensi dalam penyusunan sistem monitoring agar lebih akurat. Data-data tersebut tentunya sangat membantu dalam pembuatan analisis sistem dan desain sistem pada penelitian ini. Berikut pemaparan hasil perbandingan antara sistem lama dan sistem baru serta desain sistem yang akan berjalan.

4.2.1 Analisis Sistem Lama

Analisis sistem lama merupakan hasil analisis setelah penelitian terhadap sistem dan kinerja yang selama ini berjalan. Adapun hasil analisis sistem lama yang berjalan dalam proses penyewaan *Wedding Organizer* ialah pelanggan datang ketempat Vendor, kemudian Admin memberikan formulir pemesanan kepada pelanggan, pelanggan kemudian mengisi formulir tersebut seperti nama, alamat, tanggal

pernikahan dan paket apa saja yang akan diambil serta membayar DP sesuai dengan perjanjian, selanjutnya pelanggan menyerahkan persyaratan dan formulir tersebut kepada admin, setelah itu admin menerima dan mengecek kelengkapan jika sudah lengkap admin mencatat data pelanggan di buku yang sering kali terjadi kesalahan penulisan, kemudian admin memberikan kwitansi kepada pelanggan.

4.2.2 Analisis Sistem Baru

Analisis sistem baru adalah hasil analisis yang berasal dari penelitian pada sistem lama sebagai pengembangan sistem baru yang akan menjadi solusi dan bermanfaat dalam penggunaannya. Adapun hasil analisis sistem baru pada penelitian ini adalah membangun sistem monitoring dan evaluasi *Wedding Organizer* yang dapat membantu dalam pemantauan penyewaan *Wedding Organizer*, sistem ini diimplementasikan dengan bahasa pemrograman WEB.

Aplikasi ini mampu memverifikasi akun vendor, menghapus atau memblokir akun vendor, mengetahui data vendor, data user, produk vendor, memonitoring pemesanan, menambah dan menghapus kategori produk, mengetahui grafik transaksi penjualan terlaris menurut penilaian pelanggan, serta statistik produk terlaris dari semua vendor.

4.2.3 Desain Sistem

Tahap ini memaparkan formulasi dan pemilihan alternatif solusi. Berdasarkan permasalahan sistem pada *Corry Wedding*, maka dibutuhkan suatu perencanaan dalam pengembangan sistem sehingga mampu mempermudah kinerja dalam penyewaan *Wedding Organizer* sesuai dengan analisa yang berasal dari data yang didapat dan membutuhkan suatu Sistem Informasi Monitoring dan Evaluasi *Wedding Organizer* dari proses pemantauan dan pencatatan manual menjadi terkomputerisasi.

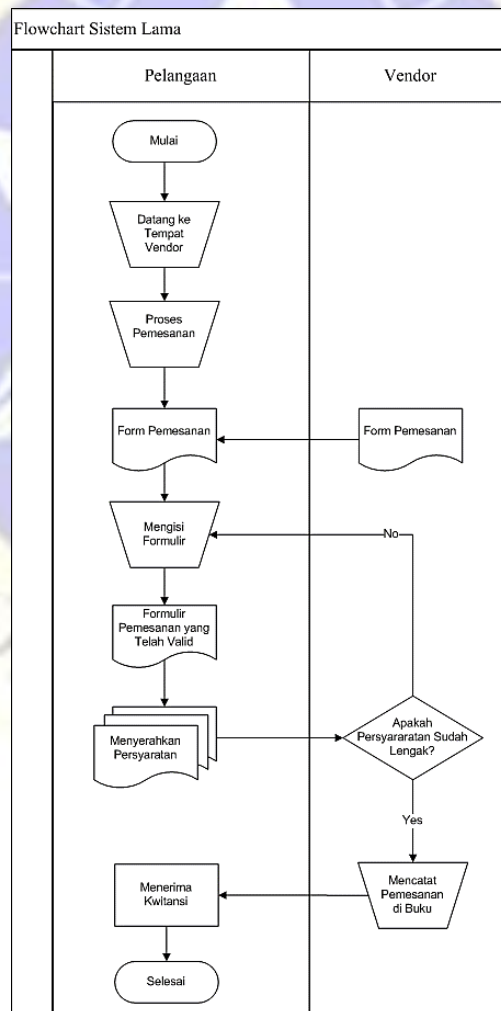
Adapun sistem yang terkomputerisasi memerlukan suatu desain perancangan sistem sebagai acuan dalam pembuatan aplikasi, agar proses pembuatan sistem menjadi lebih tertata dan sesuai dengan prosedur yang

ada. Selain itu, kegunaan desain sistem yaitu untuk mengetahui dan memberikan gambaran sistem secara menyeluruh, baik sistem yang sedang digunakan ataupun yang akan diimplementasikan.

Sedangkan desain atau perancangan sistem yang digunakan dalam proses pembuatan aplikasi Sistem Monitoring dan Evaluasi *Wedding Organizer* terdiri dari *Flowchart*, *Data Flow Diagram (DFD)*, *Eternity Relationship Diagram (ERD)*, desain *Database* dan desain *Interfacenya*. Berikut ini penjabaran mengenai desain-desain tersebut :

1. *Flowchart* sistem lama

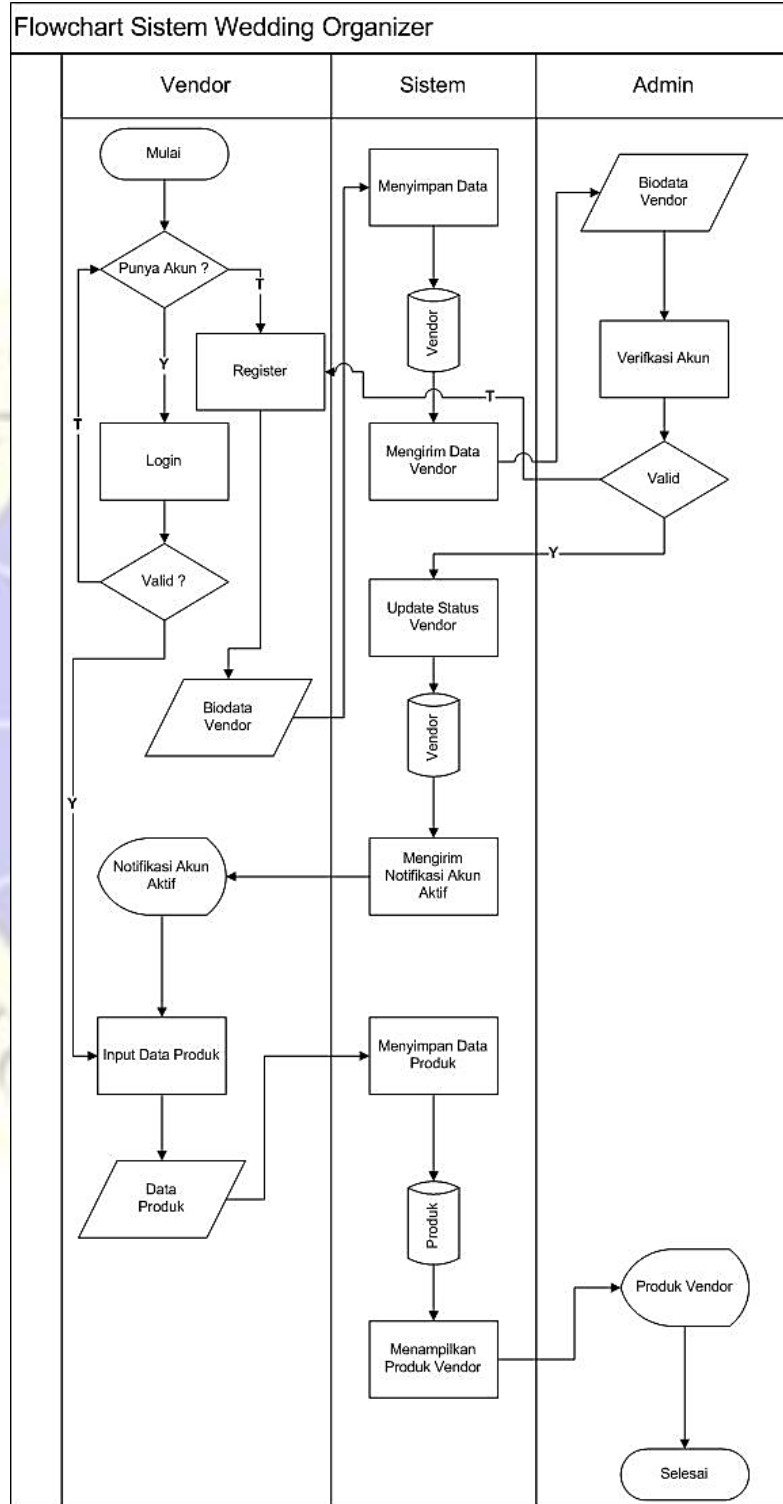
Flowchart sistem lama yang berjalan selama ini pada proses penyewaan *Wedding Organizer* sebagaimana gambar berikut :



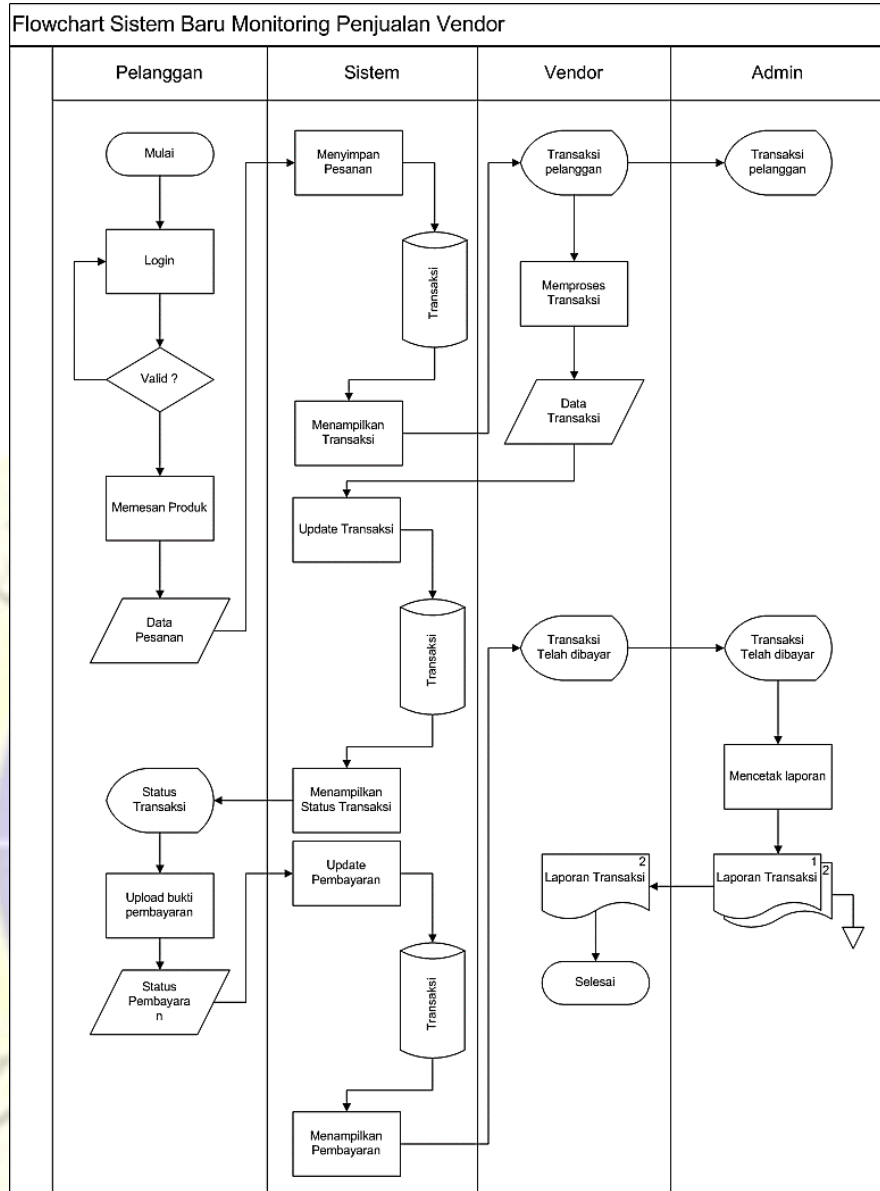
Gambar 4.1 *Flowchart* sistem lama

2. Flowchart Sistem Baru

Flowchart sistem baru pada proses monitoring dan evaluasi *Wedding Organizer* dapat dilihat pada gambar berikut:



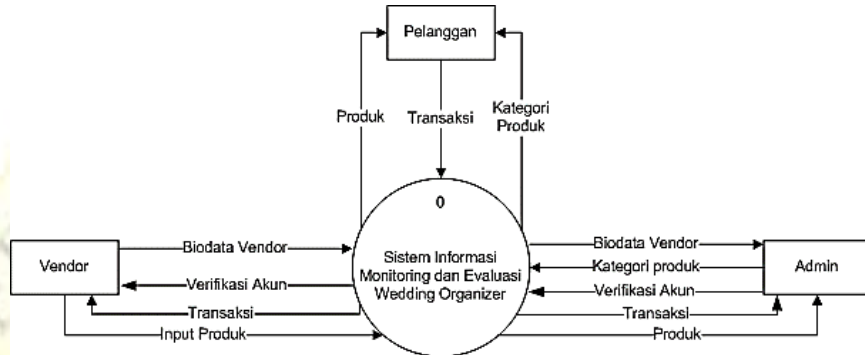
Gambar 4.2 Flowchart sistem baru



Gambar 4.3 Flowchart sistem baru monitoring penjualan Vendor

3. Diagram Context

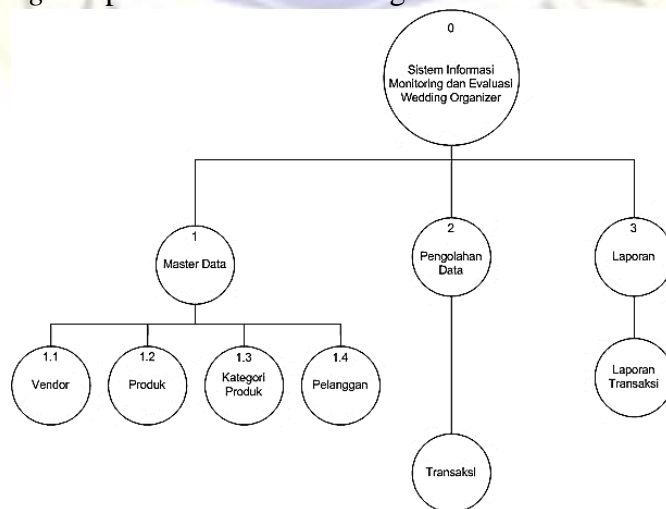
Data Flow Diagram (DFD) pada *Diagram Context* yang juga dapat disebut DFD level 0 menggambarkan keterkaitan antara Sistem informasi dengan entitas-entitas yang saling berhubungan secara global. Adapun diagram context sistem monitoring dan evaluasi *Wedding Organizer* adalah sebagai berikut :



Gambar 4.4 *Diagram Context*

4. Decomposition Diagram

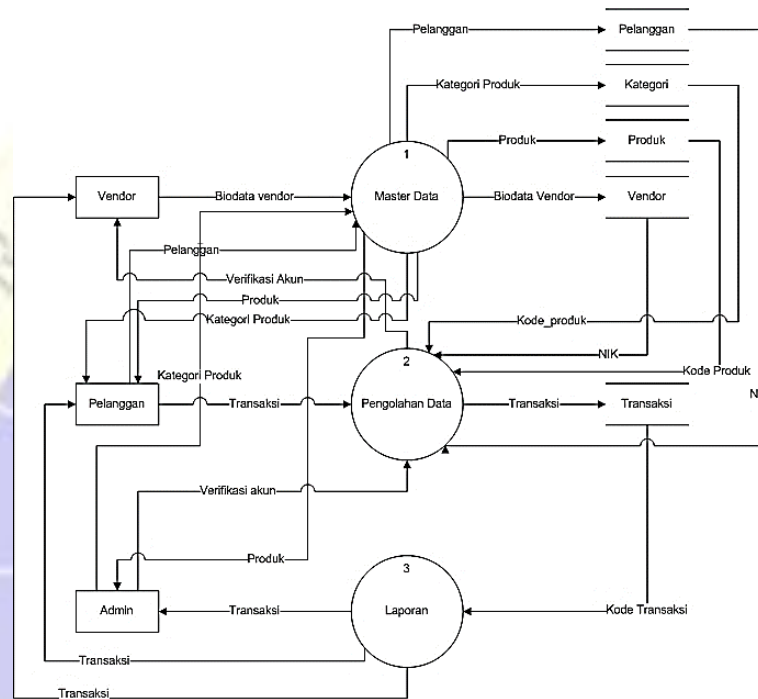
Decomposition Diagram adalah alat yang digunakan untuk menggambarkan bagan berjenjang dan alat perencanaan untuk model proses yang lebih detail. Bagan ini disebut juga bagan hierarki karena menunjukkan dekomposisi fungsional *top-down* dan struktur sistem. Dekomposisi diagram dapat di gambarkan dengan menggunakan notasi proses yang digunakan dalam *Data Flow Diagram (DFD)*. Gambar di bawah ini merupakan *Decomposition Diagram* pada sistem monitoring dan evaluasi *Wedding Organizer*.



Gambar 4.5 *Decomposition Diagram*

5. Data Flow Diagram (DFD) Level 1

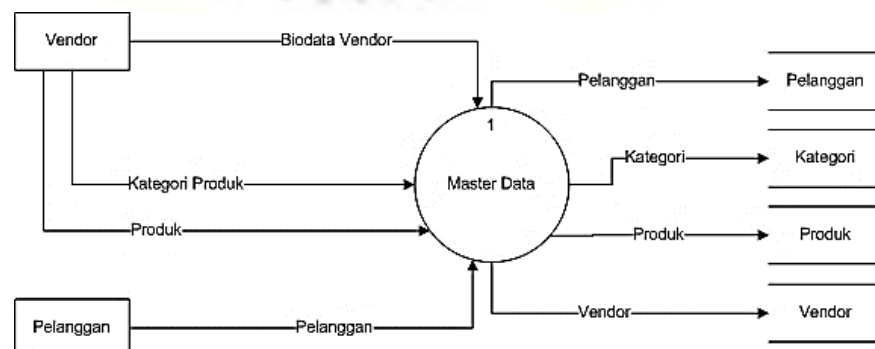
Pada Data Flow Diagram Level 1 alur sistem lebih terperinci. Diagram ini menambah proses hubungan yang terjadi pada sistem entitas dan proses akan membentuk sistem secara utuh. DFD level 1 sistem monitoring dan evaluasi *Wedding Organizer* dapat dilihat pada gambar 4.5 di bawah ini :



Gambar 4.6 Data Flow Diagram (DFD) Level 1

6. Data Flow Diagram (DFD) Level 2 Proses 1 Master Data

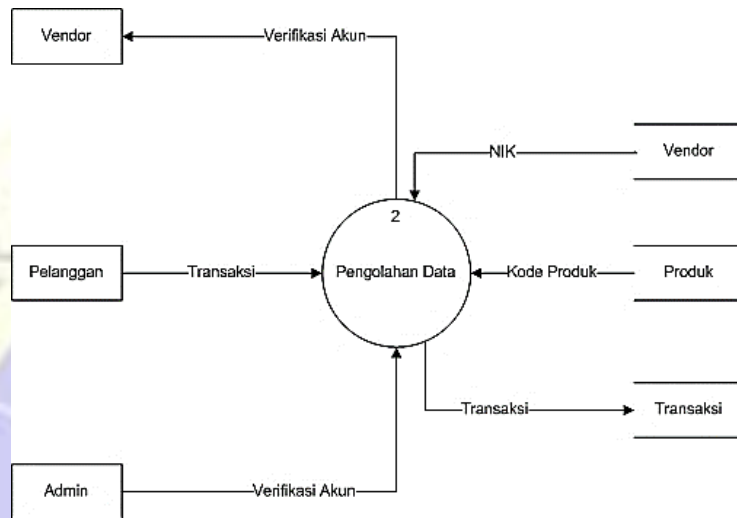
Pada level ini DFD hanya fokus terhadap pembahasan kinerja proses Master Data pada sistem. Sedangkan Master data sendiri sebagai variabel database Vendor. Adapun aliran data pada Master data dapat dilihat pada gambar di bawah ini :



Gambar 4.7 Data Flow Diagram (DFD) Level 2 Proses 1 Master Data

7. Data Flow Diagram (DFD) Level 2 Proses 2 Pengolahan Data

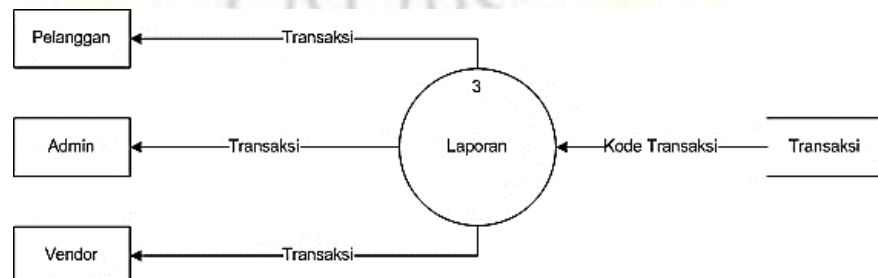
Data Flow Diagram (DFD) Level 2 yang berupa proses pengolahan data mencakup proses data Vendor, produk, transaksi . Dalam hal ini pihak-pihak yang terlibat dalam pengolahan data di antaranya adalah Vendor dan Admin. Gambar di bawah ini memaparkan alir data pada proses pengolahan data:



Gambar 4.8 Data Flow Diagram (DFD) Level 2 Proses 2 pengolahan Data

8. Data Flow Diagram (DFD) Level 2 Proses 3 Laporan

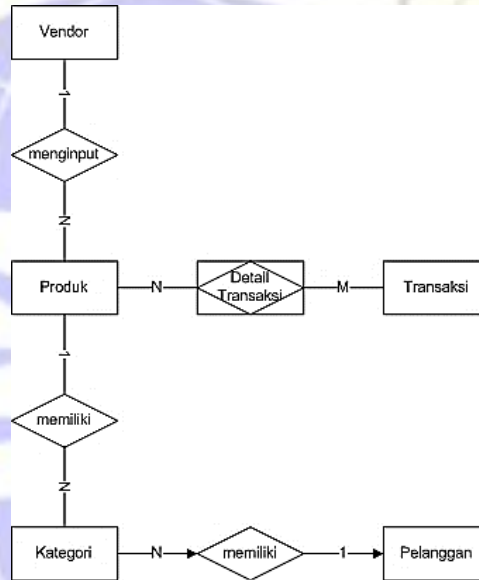
Laporan merupakan proses terakhir dari sistem monitoring dan evaluasi *Wedding Organizer*. Laporan yang dimaksud berupa laporan transaksi penjualan yang diserahkan kepada Admin dan Vendor. Di bawah ini merupakan gambar proses Laporan sistem monitoring dan evaluasi *Wedding Organizer*.



Gambar 4.9 Data Flow Diagram (DFD) Level 2 Proses 3 Laporan

9. Entity Relationship Diagram (ERD)

Setelah merancang DFD yang diterapkan terhadap sistem yang akan dibangun, tahap yang akan dilewati selanjutnya adalah penentuan pola hubungan antar entitas. Oleh karena itu dibutuhkan Entity Relationship Diagram (ERD) dalam penyempurnaan desain sistem pada suatu sistem. Sebab dengan adanya ERD akan dapat dipahami istilah hubungan Many to Many, One to Many dan One to One. Dengan begitu beberapa entitas yang saling berkaitan akan dipahami secara mendetail. ERD tersebut dapat dilihat pada gambar 4.9 di bawah ini.



Gambar 4.10 Entity Relationship Diagram (ERD)

Kamus data :

1. Vendor : nik, nama_user, tgl_lahir, jenis_kelamin, provinsi, kabupaten, kecamatan, desa, alamat, no_telepon, username, password, foto, status, verifikasi, blokir
2. Produk : nama_produk, kode_produk, harga, nik_vendor, kategori
3. Transaksi: id, user, vendor, kode_transaksi, tgl_acara, tgl_persetujuan, dp, bukti dp, lunas, status, notif
4. Kategori : id, nama

5. Pelanggan: id, nik, nama_user, tgl_lahir, jenis_kelamin, provinsi, kabupaten, kecamatan, desa, alamat, no_telphone, username, password, foto, verifikasi, komentar, blokir

4.2.4 Desain Database

Database merupakan variabel penyimpanan data pada suatu sistem serta pengolahannya. Oleh sebab itu sangat diperlukan pola struktur database yang baik untuk menghindari redundansi data. Database pada sistem ini menggunakan aplikasi *MySQL* yang terdiri dari beberapa tabel yang dipaparkan di bawah ini :

1. Tabel Vendor

Tabel 4.1 Tabel Vendor

Field Name	Data Type	Size	Primary Key
Id	bigint	20	*
Nik	varchar	16	
nama_user	varchar	100	
tgl_lahir	Date		
jenis_kelamin	varchar	1	
provinsi	varchar	100	
kabupaten	varchar	100	
kecamatan	varchar	100	
desa	varchar	100	
alamat	varchar	255	
no_telphone	varchar	100	
username	varchar	100	
password	varchar	100	
foto	varchar	100	
status	enum('u', 'v')		
verifikasi	enum('0', '1', '2')		
komentar	varchar	255	
blokir	Int	1	

2. Tabel Produk

Tabel 4.2 Tabel Produk

Field Name	Data Type	Size	Primary Key
Id	bigint	20	*
nama_produk	varchar	100	
kode_produk	varchar	25	
harga	Int	11	
deskripsi	Text		
nik_vendor	varchar	16	
kategori	varchar	1	
status	enum('t', 'f')		

3. Tabel Transaksi

Tabel 4.3 Tabel Transaksi

Field Name	Data Type	Size	Primary Key
Id	Bigint	20	*
user	Bigint	16	
vendor	Bigint	16	
kode_transaksi	Varchar	25	
tgl_acara	Datetime		
tgl_persetujuan	Datetime		
dp	Int	11	
bukti_dp	Varchar	100	
lunas	Int	1	
status	enum('pengajuan', 'diproses', 'dikirim', 'selesai')		
notif	enum('0', '1')		

4. Tabel Kategori

Tabel 4.4 Tabel Kategori

Field Name	Data Type	Size	Primary Key
Id	bigint	20	*
Nama	varchar	100	

5. Tabel Pelangan

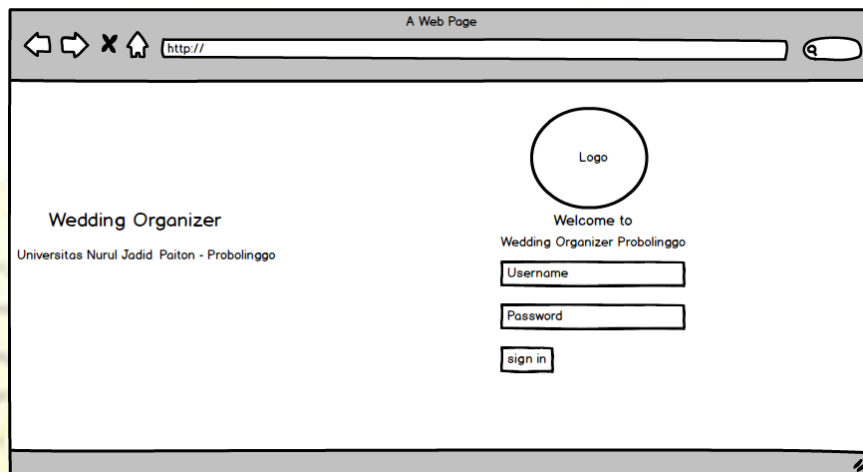
Tabel 4.5 Tabel Pelanggan

Field Name	Data Type	Size	Primary Key
Id	bigint	20	*
nik	varchar	100	
nama_user	varchar	16	
tgl_lahir	date	1	
jenis_kelamin	varchar	100	
Provinsi	varchar	100	
kabupaten	varchar	100	
kecamatan	varchar	100	
Desa	varchar	100	
Alamat	varchar	255	
no_telphone	varchar	100	
username	varchar	100	
password	varchar	100	
Foto	varchar	100	
verifikasi	enum('0', '1', '2')	100	
komentar	varchar	255	
Blokir	varchar	100	

4.2.5 Desain *Interface*

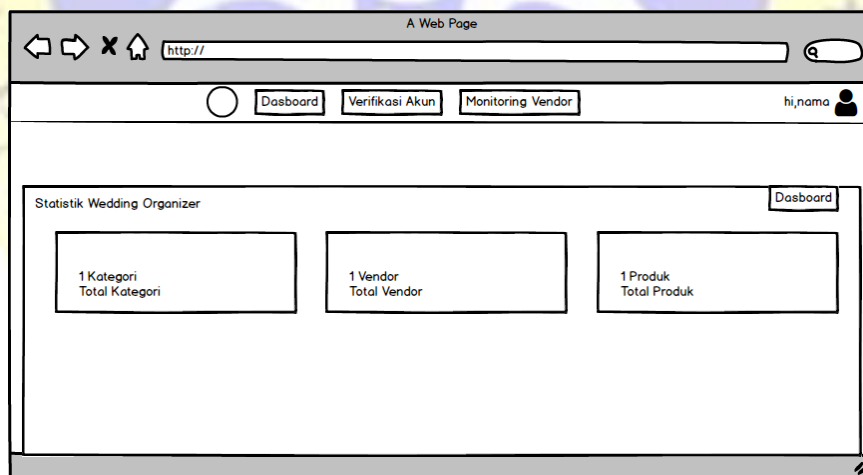
Pada desain ini akan ditampilkan gambaran *interface* aplikasi yang akan diimplementasikan ke dalam sebuah perangkat lunak. Gambaran aplikasi yang dibuat berupa *interface* untuk melakukan proses yang akan berjalan. Berikut di bawah ini akan menunjukkan desain *interface* pada Sistem Informasi Monitoring dan Evaluasi *Wedding Organizer*

1. Desain *Interface Login*



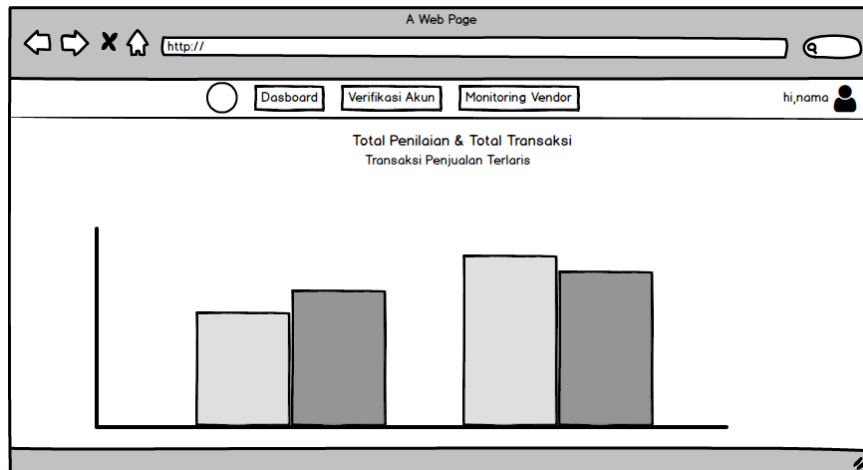
Gambar 4.11 Desain *Interface Login*

2. Desain *Interface Dashboard*



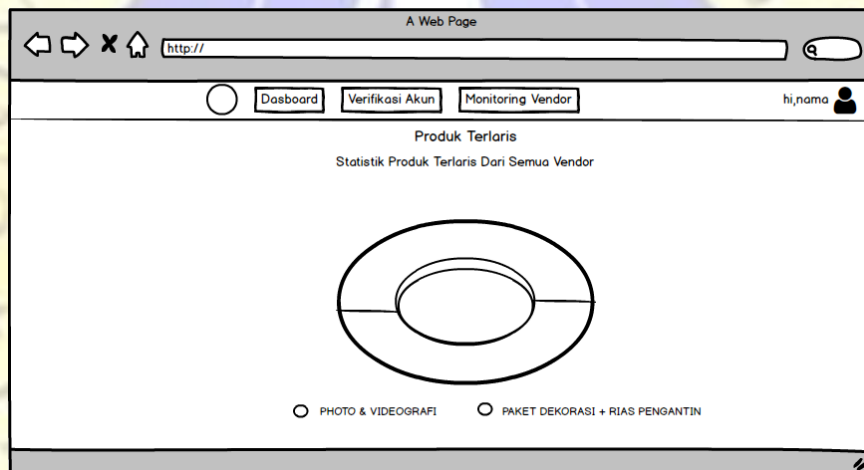
Gambar 4.12 Desain *Interface Dashboard*

3. Desain *Interface Dashboard* Grafik Transaksi Penjualan Terlaris



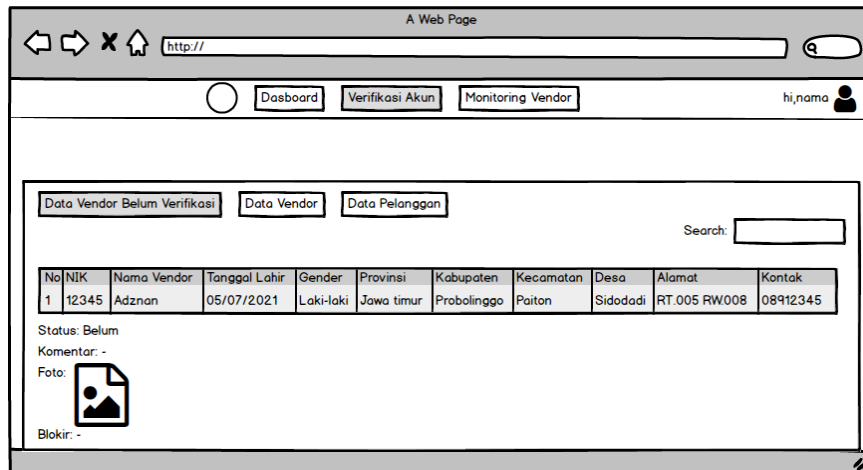
Gambar 4.13 Desain *Interface Dashboard* Grafik Transaksi Penjualan Terlaris

4. Desain *Interface Dashboard* Statistik Produk Terlaris



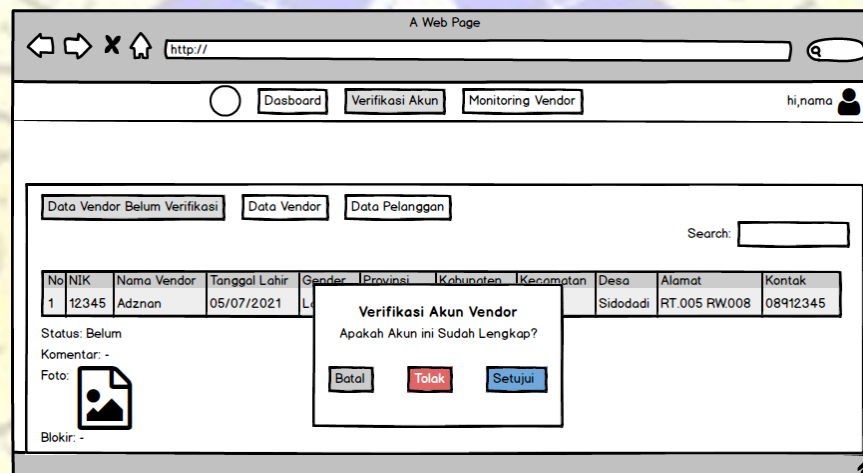
Gambar 4.14 Desain *Interface Dashboard* Statistik Produk Terlaris

5. Desain *Interface* Data Vendor Belum Verifikasi



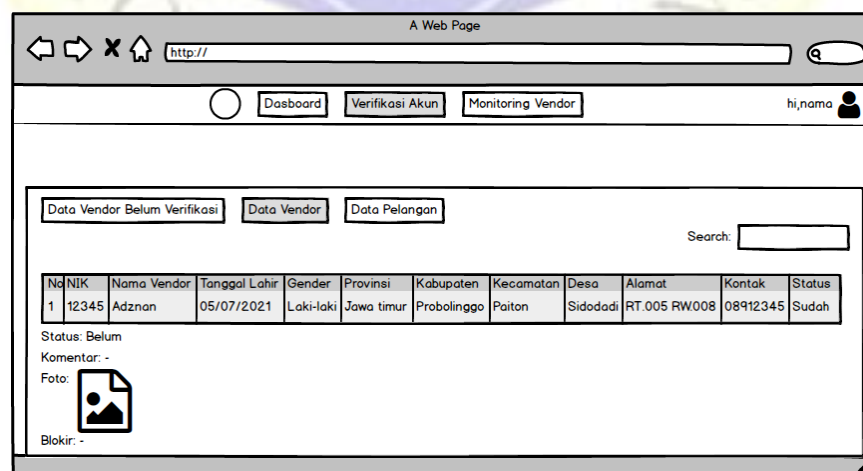
Gambar 4.15 Desain *Interface* Data Vendor Belum Verifikasi

6. Desain *Interface* Verifikasi Akun Vendor



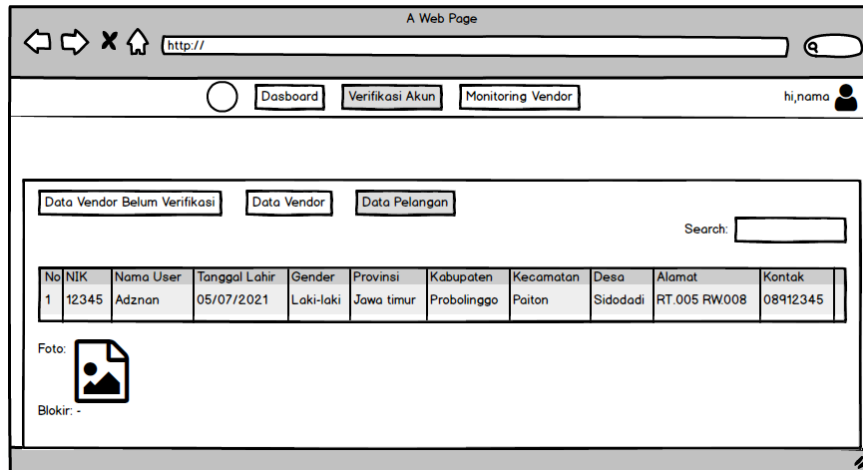
Gambar 4.16 Desain *Interface* Verifikasi Akun Vendor

7. Desain *Interface* Data Vendor



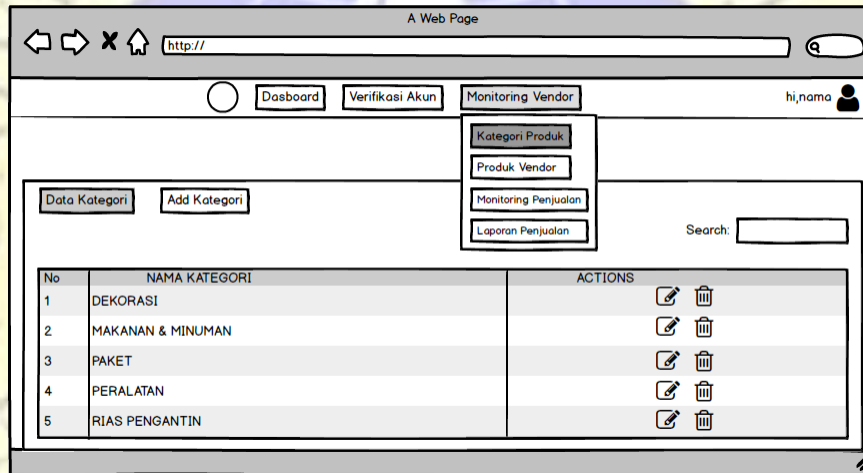
Gambar 4.17 Desain *Interface* Data Vendor

8. Desain *Interface* Data Pelanggan



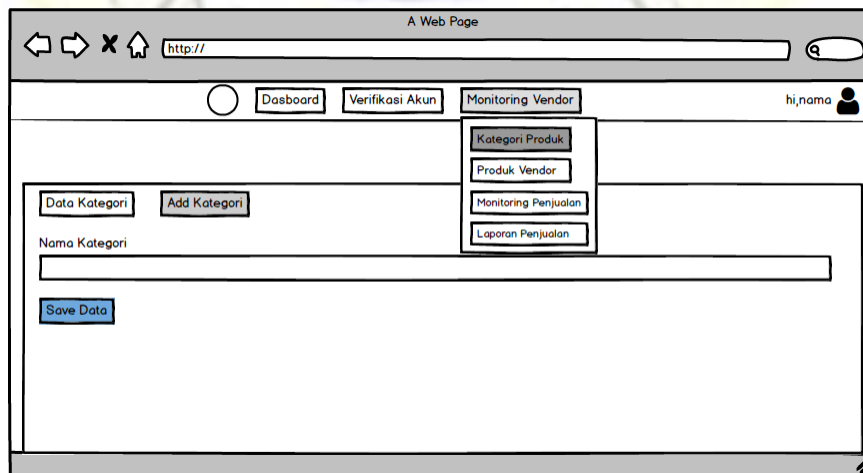
Gambar 4.18 Desain *Interface* Data Pelanggan

9. Desain *Interface* Kategori Produk



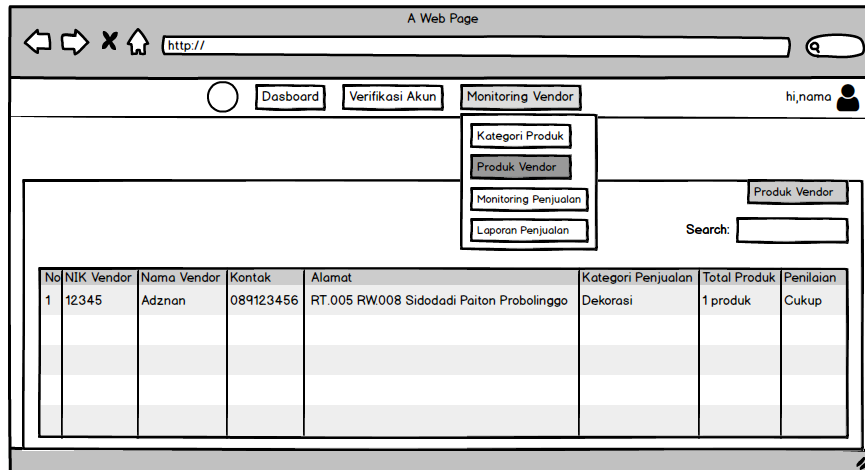
Gambar 4.19 Desain *Interface* Kategori Produk

10. Desain *Interface* add Kategori Produk



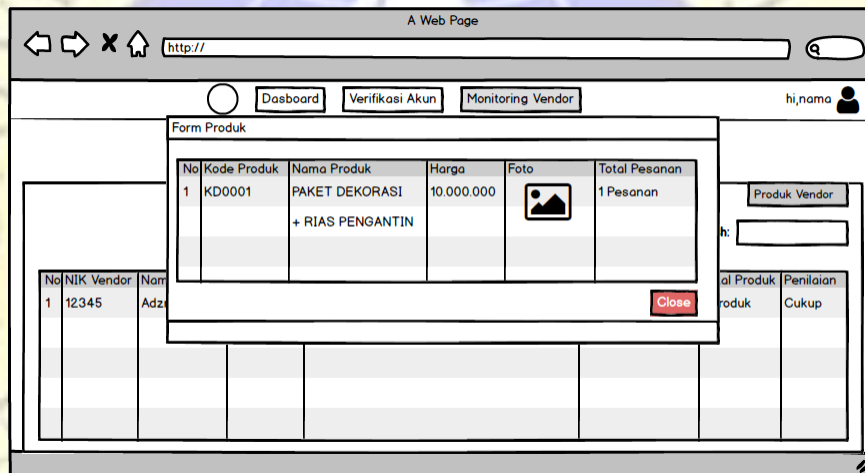
Gambar 4.20 Desain *Interface* add Kategori Produk

11. Desain *Interface* Produk Vendor



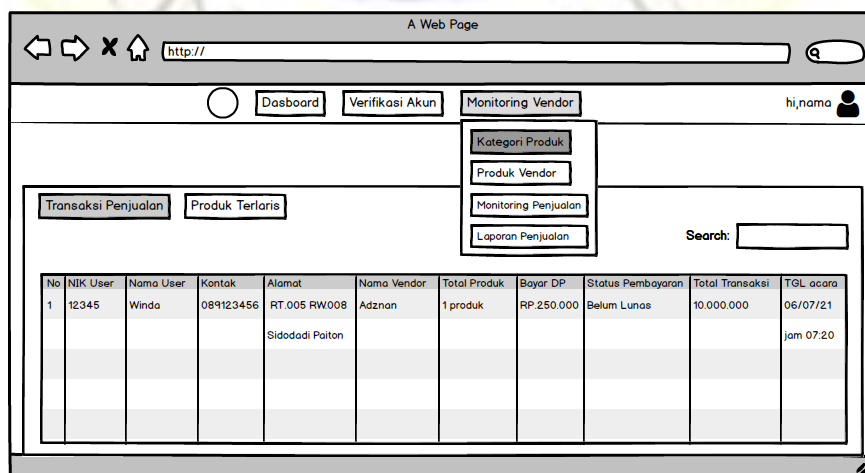
Gambar 4.21 Desain *Interface* Produk Vendor

12. Desain *Interface* Form Produk



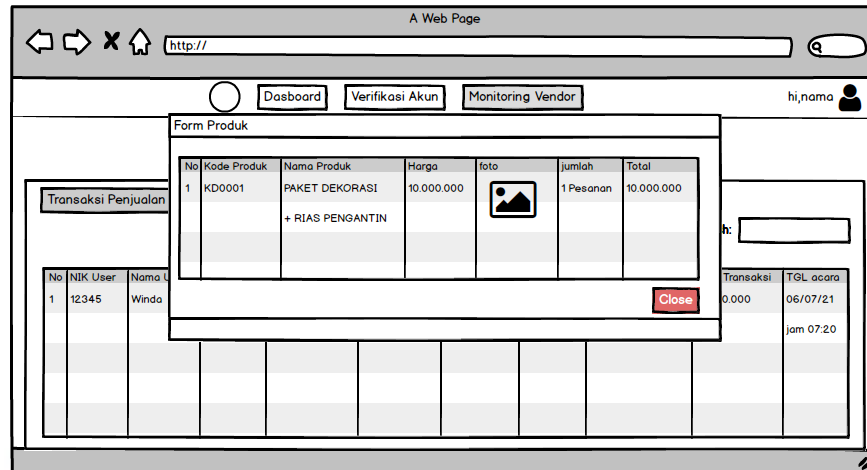
Gambar 4.22 Desain *Interface* Form Produk

13. Desain *Interface* Transaksi Penjualan



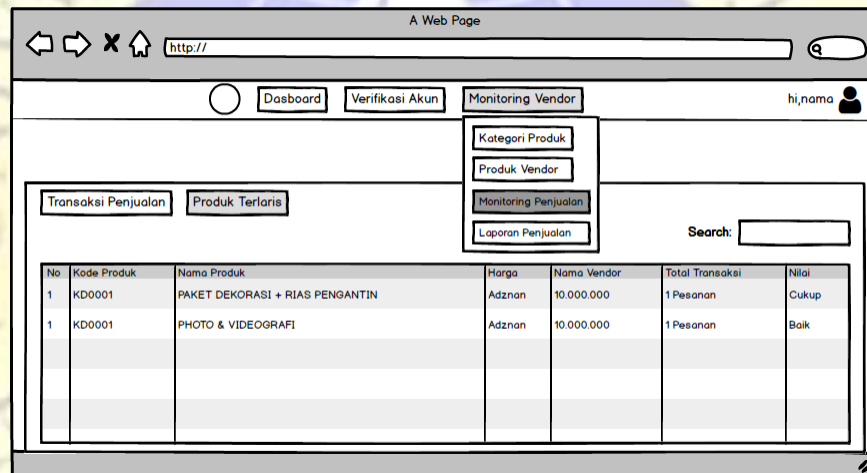
Gambar 4.23 Desain *Interface* Transaksi Penjualan

14. Desain *Interface* Form Produk Transaksi Penjualan



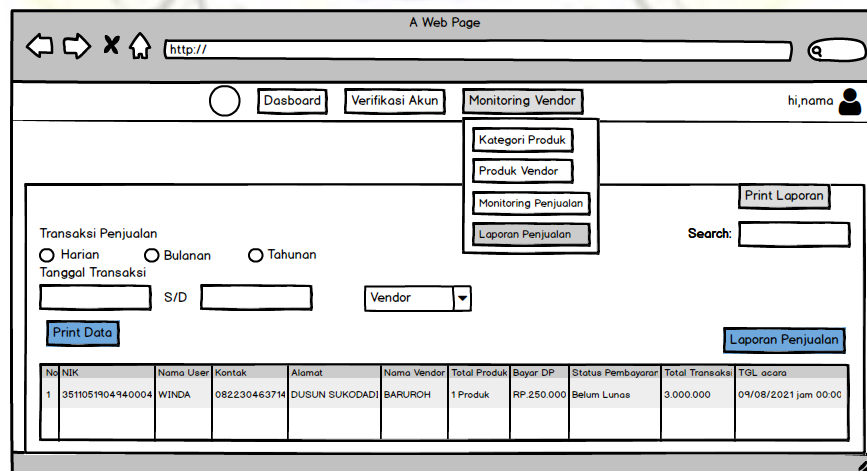
Gambar 4.24 Desain *Interface* Form Produk Transaksi Penjualan

15. Desain *Interface* Produk Terlaris



Gambar 4.25 Desain *Interface* Produk Terlaris

16. Desain *Interface* Laporan Penjualan



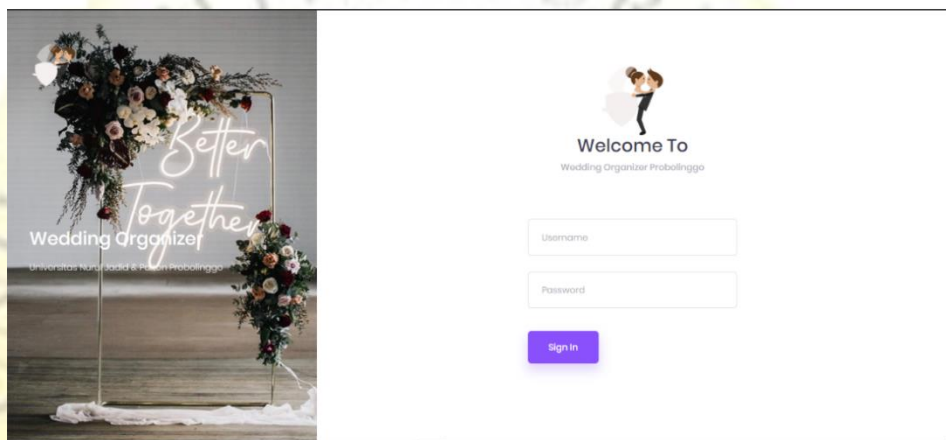
Gambar 4.26 Desain *Interface* Laporan Penjualan

4.3 Implementasi

Dalam implementasi Sistem Informasi Monitoring dan Evaluasi *Wedding Organizer* ini menghasilkan beberapa tampilan (*Interface*) yang digunakan, di antaranya adalah sebagai berikut :

4.3.1 Halaman Login

Untuk dapat mengakses Sistem Informasi Monitoring dan Evaluasi *Wedding Organizer* diperlukan *Login User*. Hal ini juga bertujuan untuk membatasi hak akses masing-masing pengguna, sebagaimana tampilan berikut ini :



Gambar 4.27 Halaman Login

Segmen program 4.1 Halaman login

```
@csrf_exempt

def auth(request):

    arr = 0

    if request.method == "POST":

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

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

        auth = (

            Admin.objects.filter(username=username).filter(

                password=password).exists()

        )

        # set session
```


Segmen program 4.1 Halaman login

```
# request.session['fav_color'] = 'blue'

# get session

# request.session['fav_color']

if auth:

    data = (

        Admin.objects.filter(username=username).filter(

            password=password).all()

        )

    for val in data.iterator():

        request.session["nik"] = val.nik

        request.session["nama"] = val.nama_admin

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

        request.session["gender"] = val.jenis_kelamin

        request.session["pro"] = val.provinsi

        request.session["kab"] = val.kabupaten

        request.session["kec"] = val.kecamatan

        request.session["desa"] = val.desa

        request.session["alamat"] = val.alamat

        request.session["hp"] = val.no_telphone

        arr = 1

    else:

        arr = 2

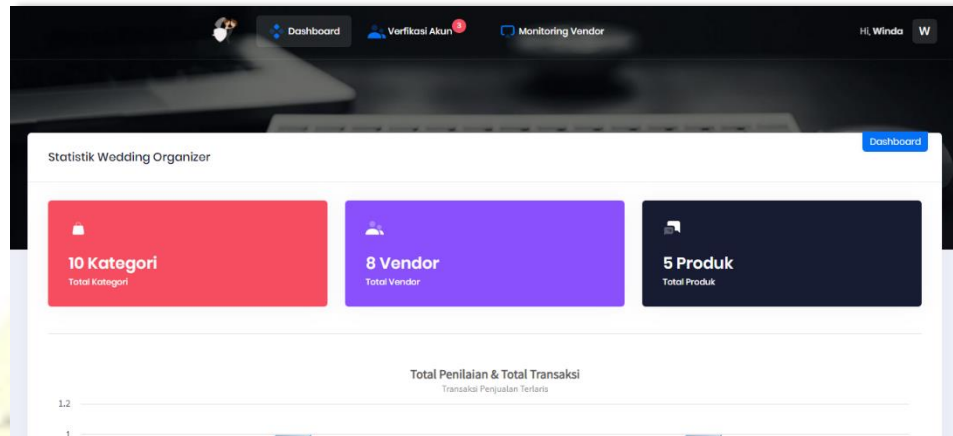
else:

    arr = 3

return HttpResponse(arr)
```

4.3.2 Halaman *Dashboard*

Setelah validasi *login* berhasil dilakukan, maka *user* akan masuk pada tampilan menu utama (*Dashboard*). Halaman ini menampilkan segala hak akses yang didapat oleh Admin. Gambar di bawah ini merupakan tampilan *Dashboard* :



Gambar 4.28 Halaman *Dashboard*

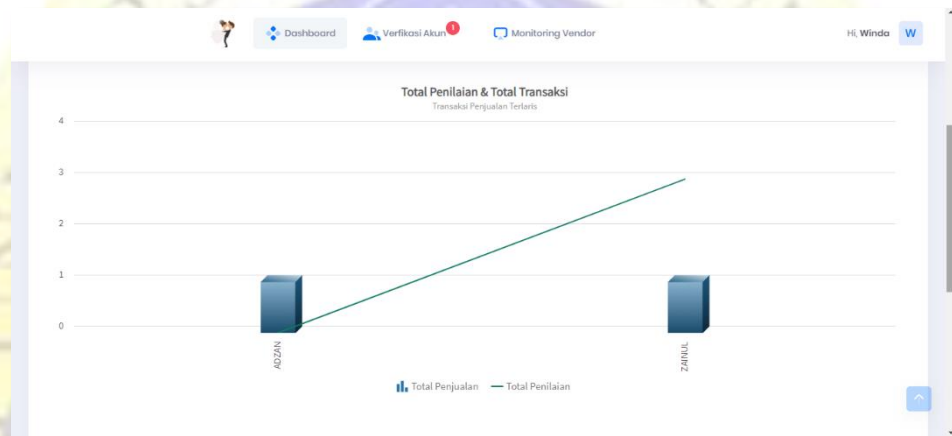
Segmen program 4.2 Halaman *Dashboard*

```
def home(request):  
    if request.session["nik"] == None:  
        return render(request, "pages/index.html")  
    else:  
        kategori = Kategori.objects.all().count  
        subkategori = SubKategori.objects.all().count  
        produk = Produk.objects.all().count  
        vendor = User.objects.filter(status='V').all().count  
        outpus = {  
            "url": "home",  
            "kategori": kategori,  
            "subkategori": subkategori,  
            "produk": produk,  
            "vendor": vendor,
```

Segmen program 4.2 Halaman *Dashboard*(lanjutan)

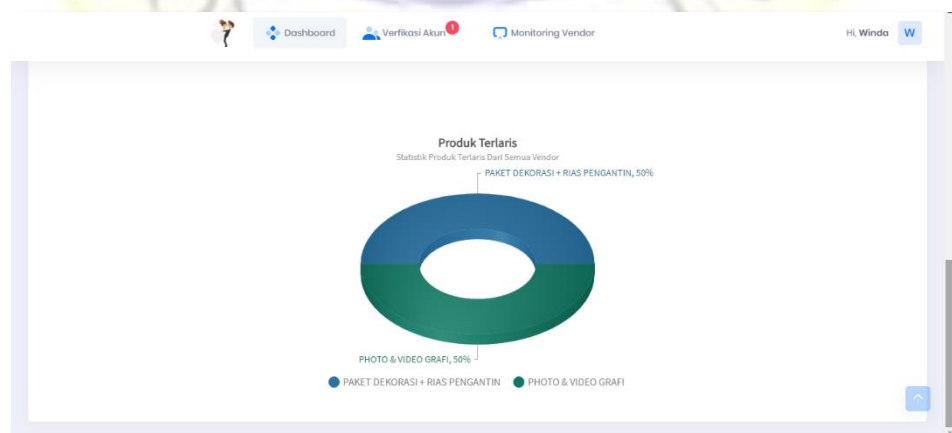
```
}  
  
return render(request, "pages/home.html", outpus)
```

Selanjutnya tampilan Grafik Transaksi Penjualan Terlaris yang terdapat di Dashboard, grafik ini ditentukan oleh total penilaian dan total transaksi pada setiap vendor, Berikut ini adalah tampilan Grafik Transaksi Penjualan Terlaris:



Gambar 4.29 Halaman *Dashboard* Grafik Transaksi Penjualan Terlaris

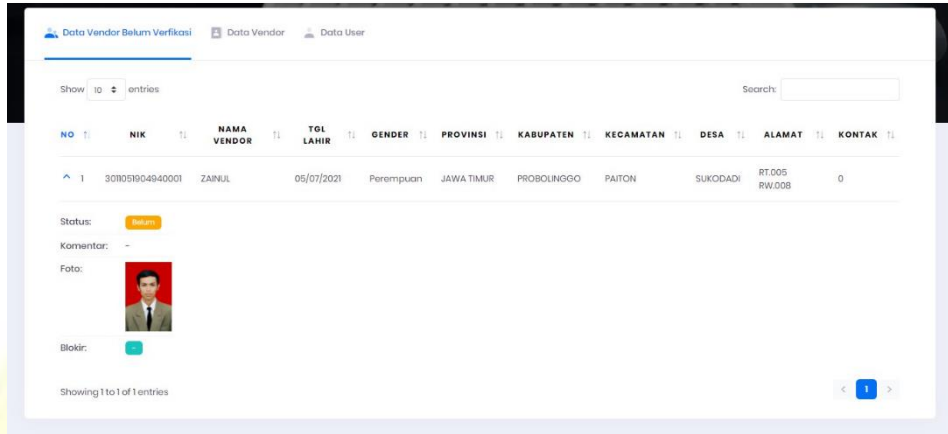
Kemudian tampilan Statistik Produk Terlaris yang terdapat di Dashboard, Statistik ini merupakan statistik produk terlaris dari semua Vendor, Berikut ini adalah tampilan Statistik Produk Terlaris:



Gambar 4.30 Halaman *Dashboard* Statistik Produk Terlaris

4.3.3 Halaman Data Vendor Belum Verifikasi

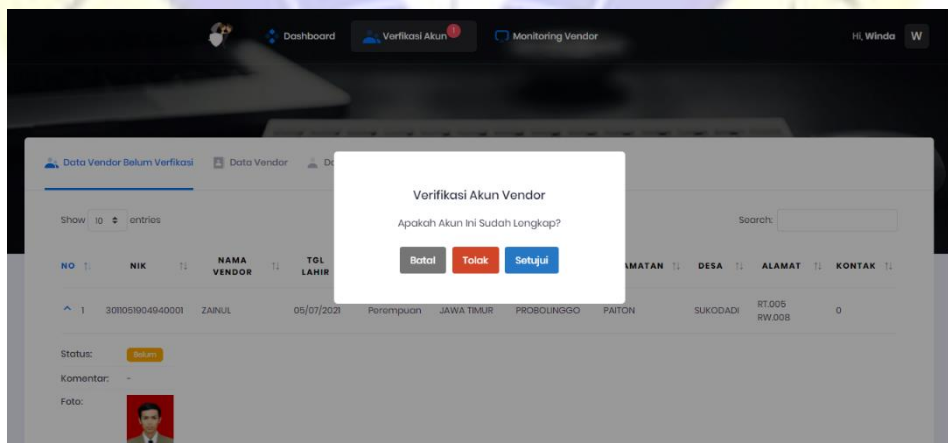
Pada halaman data Vendor belum verifikasi Admin akan ditampilkan data Vendor untuk di cek dan diverifikasi. Berikut ini adalah tampilan data Vendor belum verifikasi dan tampilan untuk memverifikasi:



Gambar 4.31 Halaman Data Vendor Belum Verifikasi

Segmen program 4.3 Halaman Data Vendor Belum Verifikasi

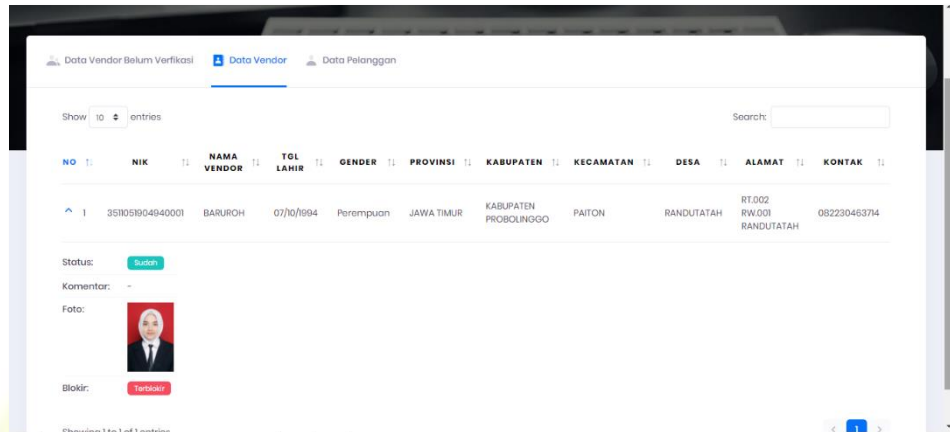
```
def verifikasi(request):  
    if request.session["nik"] == None:  
        return render(request, "pages/index.html")  
    else:  
        return render(request, "pages/verifikasi.html", {"url": "verifikasi"})
```



Gambar 4.32 Verifikasi Akun Vendor

4.3.4 Halaman Data Vendor

Pada halaman Data Vendor, Admin akan ditampilkan seluruh data Vendor yang ada. Berikut ini adalah tampilan dari Data Vendor:



Gambar 4.33 Data Vendor

Segmen program 4.4 Data Vendor

```
@csrf_exempt
def tblvndor(request):
    status = request.POST.get("status")
    sub = User.objects.filter(verifikasi=status, status='V')
    sub = sub.all().order_by('-createdAt').order_by('nama_user')
    data = []
    out = []
    no = 1
    for val in sub.iterator():
        nom = "<center>" + str(no) + "</center>"
        nik = val.nik
        nama = val.nama_user.upper()
        tgl = val.tgl_lahir.strftime('%d/%m/%Y')
        if val.jenis_kelamin == 'L':
            gender = 'Laki-laki'
        else:
            gender = 'Perempuan'
        pro = val.provinsi.upper()
```

Segmen program 4.4 Data Vendor(lanjutan)

```
kab = val.kabupaten.upper()
kec = val.kecamatan.upper()
desa = val.desa.upper()
alamat = val.alamat.upper()
kontak = val.no_telphone

if val.komentar in [None, '']:
    komentar = '-'
else:
    komentar = val.komentar

if val.verifikasi == "0":
    status = '<a href="javascript:verifikasi('+str(
        val.id)+')"><span class="label label-
warning label-pill label-inline mr-2">Belum</span></a>'
    elif val.verifikasi == "1":
        status = '<span class="label label-
success label-pill label-inline mr-2">Sudah</span>'
    else:
        status = '<span class="label label-
danger label-pill label-inline mr-2">Ditolak</span>'
    # print(val.blokir)
    if val.blokir == 0:
        blokir = '<a href="javascript:upblokir('+str(
            val.id)+',0)"><span class="label label-
success label-pill label-inline mr-2">Terbuka</span></a>'
    else:
        blokir = '<a href="javascript:upblokir('+str(
            val.id)+',1)"><span class="label label-
danger label-pill label-inline mr 2">Terblokir</span></a>'

    foto = "<img src='"+MEDIA_URL+"foto/"+val.foto+"' h
eight='100'/">"
```


Segmen program 4.4 Data Vendor(lanjutan)

```
        out = [nom, nik, nama, tgl, gender, pro, kab, kec,
desa,
                alamat, kontak, status, komentar, foto, blok
ir]

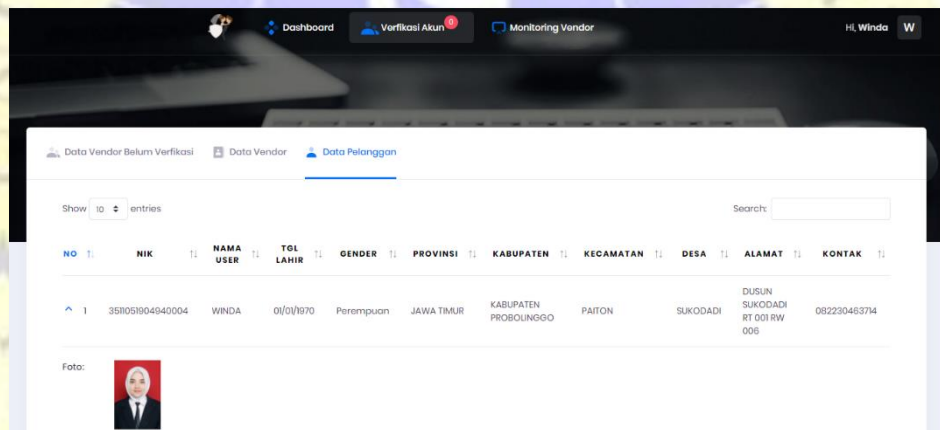
        data.append(out)

        no += 1


    output = json.dumps({"data": data})
    return HttpResponse(output, content_type="application/j
son")
```

4.3.5 Halaman Data Pelanggan

Pada halaman Data Pelanggan, Admin akan ditampilkan seluruh data User yang ada. Berikut ini adalah tampilan dari Data Pelanggan:



NO	NIK	NAMA USER	TGL LAHIR	GENDER	PROVINSI	KABUPATEN	KECAMATAN	DESA	ALAMAT	KONTAK
1	351051904640004	WINDA	01/01/1970	Perempuan	JAWA TIMUR	KABUPATEN PROBOLINGGO	PAITON	SUKODADI	DUSUN SUKODADI RT 001 RW 006	082230463714

Foto: 

Gambar 4.34 Data Pelanggan

Segmen program 4.5 Data Pelanggan

```
@csrf_exempt

def tbluser(request):

    sub = User.objects.filter(status='U').all().order_by(
        '-createdAt').order_by('nama_user')

    data = []

    out = []

    no = 1
```

Segmen program 4.5 Data Pelanggan

```
for val in sub.iterator():

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

    nik = val.nik

    nama = val.nama_user.upper()

    tgl = val.tgl_lahir.strftime('%d/%m/%Y')

    if val.jenis_kelamin == 'L':
        gender = 'Laki-laki'
    else:
        gender = 'Perempuan'

    pro = val.provinsi.upper()
    kab = val.kabupaten.upper()
    kec = val.kecamatan.upper()
    desa = val.desa.upper()
    alamat = val.alamat.upper()
    kontak = val.no_telphone

    foto = "<img src='"+MEDIA_URL+"foto/"+val.foto+"' height='100' / >"

    out = [nom, nik, nama, tgl, gender, pro, kab, kec, desa,
          alamat, kontak, foto]

    data.append(out)

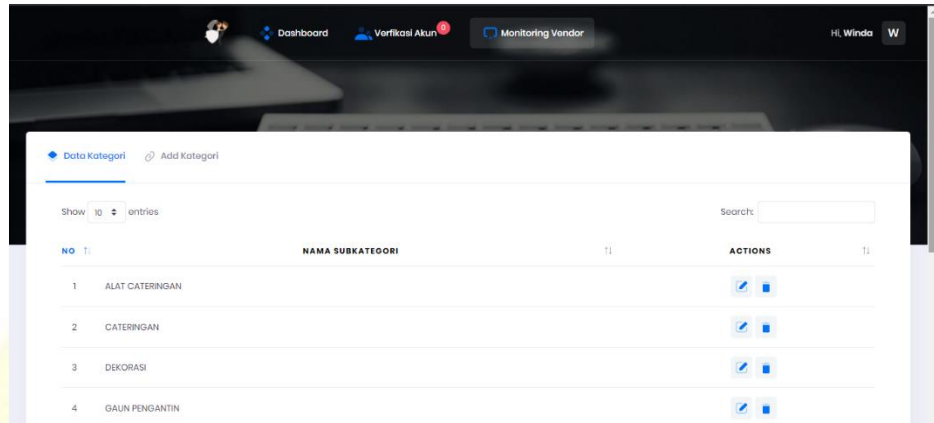
    no += 1

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

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

4.3.6 Halaman Data Kategori

Pada halaman Data Kategori, Admin akan ditampilkan seluruh Data Kategori yang ada, Admin juga dapat menghapus atau mengedit nama kategori. Berikut ini adalah tampilan dari Data Kategori:



Gambar 4.35 Data Kategori

Segmen program 4.6 Data Kategori

```
@csrf_exempt
def dkategori(request):
    id = request.POST.get("id")
    kat = Kategori.objects.filter(id=id).all()
    data = []
    value = {}
    for val in kat.iterator():
        value = {"nama": val.nama_kategori}
        data.append(value)
    output = json.dumps({"data": data})
    return HttpResponse(output, content_type="application/js
on")
```

Segmen program 4.6 Data Kategori(lanjutan)

```
@csrf_exempt

def hkategori(request):

    if request.method == "POST":

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

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

        sql = data.delete()

        if sql:

            arr = 1

        else:

            arr = 0

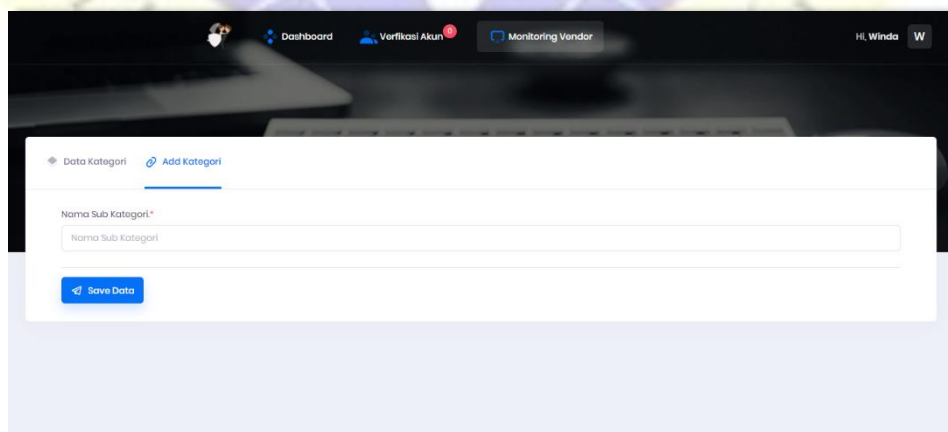
    else:

        arr = 0

    return HttpResponse(arr)
```

4.3.7 Halaman *Add Kategori*

Pada halaman *Add Kategori*, Admin akan ditampilkan untuk menginput atau menambah kategori pada produk. Berikut ini adalah tampilan dari *Add Kategori*:

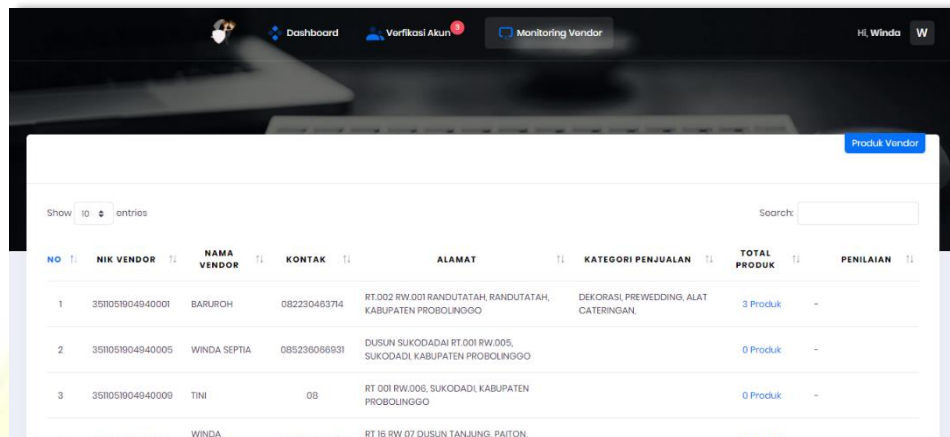


The screenshot shows a web application interface. At the top, there is a navigation bar with a user profile icon, a 'Dashboard' link, a 'Verifikasi Akun' link with a red notification badge, and a 'Monitoring Vendor' link. The user's name 'H. Winda' and a 'W' icon are visible in the top right corner. Below the navigation bar, there is a main content area with a white background. It features a breadcrumb trail with 'Data Kategori' and 'Add Kategori'. The 'Add Kategori' link is highlighted. The main form area contains a single text input field with the label 'Nama Sub Kategori*'. Below the input field is a blue button with a white arrow icon and the text 'Save Data'.

Gambar 4.36 *Add Kategori*

4.3.8 Halaman Produk Vendor

Pada halaman Produk Vendor, Admin akan ditampilkan seluruh Data Produk Vendor beserta penilaian produk tersebut, Berikut ini adalah tampilan dari Data Produk Vendor:



NO	NIK VENDOR	NAMA VENDOR	KONTAK	ALAMAT	KATEGORI PENJUALAN	TOTAL PRODUK	PENILAIAN
1	351051904940001	BARUROH	082230463714	RT.002 RW.001 RANDUTATAH, RANDUTATAH, KABUPATEN PROBOLINGGO	DEKORASI PREWEDDING, ALAT CATERINGAN,	3 Produk	-
2	351051904940005	WINDA SEPTIA	0852360566931	DUSUN SUKODADAI RT.001 RW.005, SUKODADI, KABUPATEN PROBOLINGGO		0 Produk	-
3	351051904940009	TINI	08	RT.001 RW.006, SUKODADI, KABUPATEN PROBOLINGGO		0 Produk	-

Gambar 4.37 Produk Vendor

Segmen program 4.6 Produk Vendor

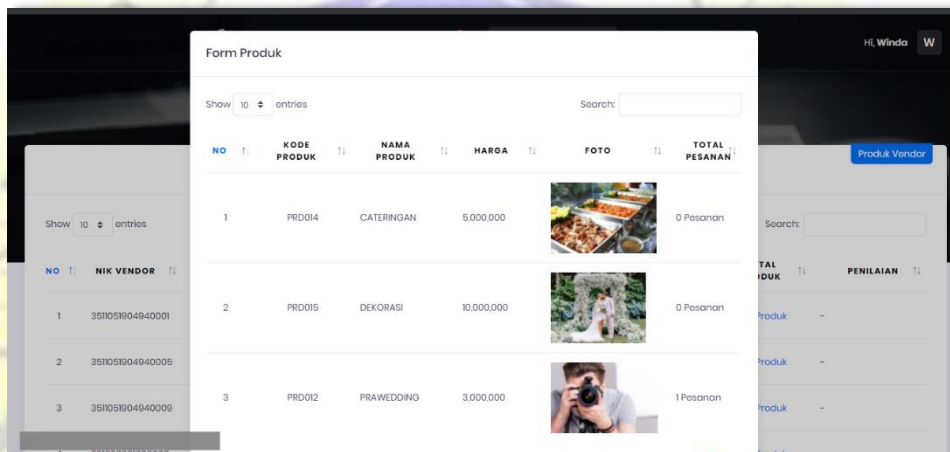
```
@csrf_exempt
def vproduk(request):
    pk = request.POST.get("nik")
    pro = Produk.objects.filter(nik_vendor=pk).all().order_by('nama_produk')
    data = []
    out = []
    no = 1
    for val in pro.iterator():
        nom = "<center>" + str(no) + "</center>"
        kode = "<center>" + val.kode_produk.upper() + "</center>"
        nama = val.nama_produk.upper()
        harga = intcomma(int(val.harga))
        total = DetailTransaksi.objects.filter(kode_produk=val.kode_produk).all().count()
        pesanan = str(total) + ' Pesanan'
        foto = '<img src="'+ANDROID_URL+'produk/foto/'+val.foto+'>' + '</img>'
        out.append(nom + kode + nama + harga + total + pesanan + foto)
    return out
```

Segmen program 4.6 Produk Vendor(lanjutan)

```
out = [nom, kode, nama, harga, foto, pesanan]
data.append(out)
no += 1
output = json.dumps({"data": data})
return HttpResponse(output, content_type="application/json")
```

4.3.9 Halaman Form Produk

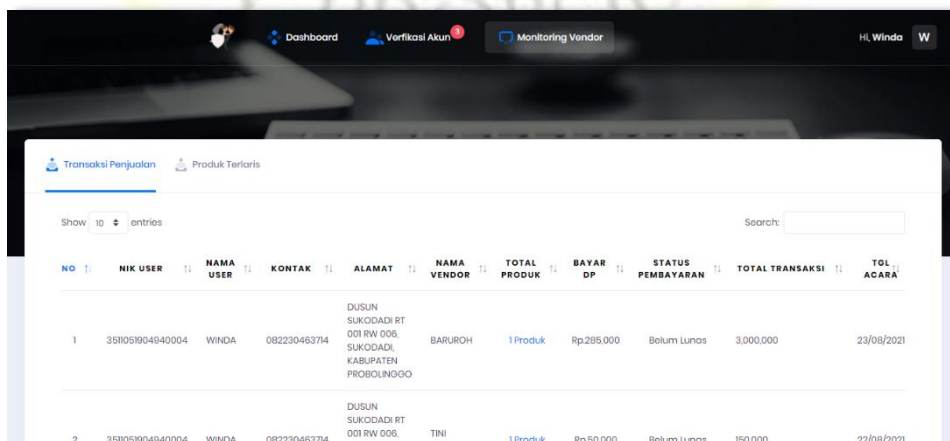
Pada halaman Form Produk, Admin akan ditampilkan seluruh Data Produk harga produk, total pesanan produk tersebut, Berikut ini adalah tampilan dari Form Produk:



Gambar 4.38 Form Produk

4.3.10 Halaman Transaksi Penjualan

Pada halaman Transaksi Penjualan, Admin akan ditampilkan seluruh Data Transaksi Penjualan, Berikut ini adalah tampilan dari Data Transaksi Penjualan:



Gambar 4.39 Transaksi Penjualan

Segmen program 4.6 Transaksi Penjualan

```
@csrf_exempt

def tbltransaksi(request):

    with connection.cursor() as con:

        con.execute("SELECT t.id,t.kode_transaksi,t.status,u
.nama_user,v.nama_user as nama_vendor,u.no_telphone as konta
k,u.desa,u.kabupaten,u.alamat,u.nik,t.tgl_persetujuan as tgl
FROM app_transaksi t LEFT JOIN app_user u ON t.user=u.nik L
EFT JOIN app_user v ON t.vendor=v.nik GROUP BY t.kode_transa
ksi")

        data = []

        out = []

        no = 1

        produk = ""

        harga = ""

        for val in con.fetchall():

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

            kode = "<center>" + val[1] + "</center>"

            nik = "<center>" + val[9] + "</center>"

            tgl = "<center>" + \

                val[10].strftime('%d/%m/%Y,<br>Jam %H:%M') +

                "</center>"

            user = val[3].upper()

            vendor = val[4].upper()

            kontak = val[5].upper()

            alamat = val[8].upper()+", "+val[6].upper()+", "

            +val[7].upper()

        with connection.cursor() as total:

            total.execute(
```

Segmen program 4.6 Transaksi Penjualan(lanjutan)

```
"SELECT COALESCE(SUM(jumlah),0) as jum,
COALESCE(SUM(jumlah*harga),0) as total FROM app_detailtransa
ksi WHERE kode_transaksi= %s", [val[1]])

ttl = total.fetchone()

produk = "<center><a href='javascript:dTrans
("+str(val[0])+")'>" + \

str(ttl[0])+" Produk" + "<a/></center>"

harga = intcomma(int(ttl[1]))

out = [nom, nik, user, kontak, alamat, vendor, p
roduk, harga, tgl]

data.append(out)

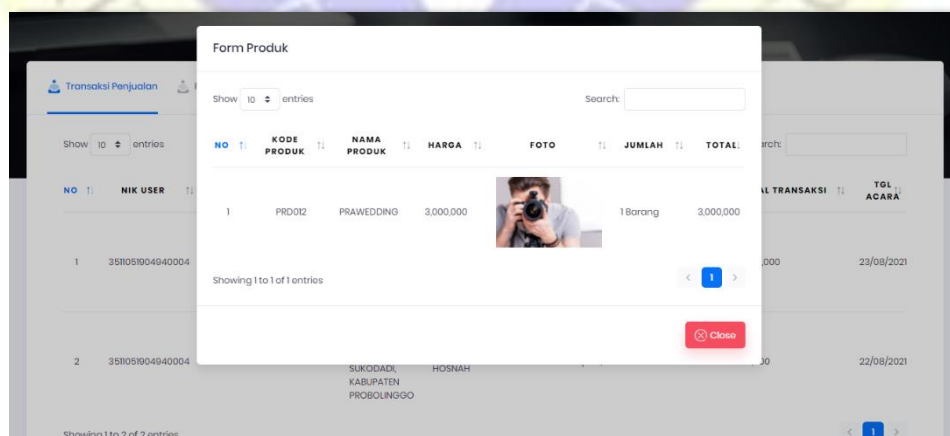
no += 1


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

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

4.3.11 Halaman Form Produk Transaksi Penjualan

Pada halaman Form Produk Transaksi Penjualan, Admin akan ditampilkan seluruh Data Produk Transaksi Penjualan, Berikut ini adalah tampilan dari Data Form Produk Transaksi Penjualan:

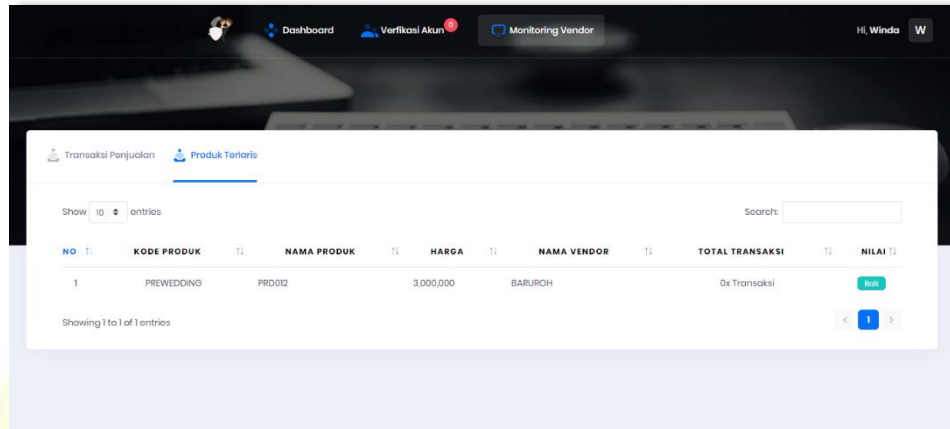


NO	KODE PRODUK	NAMA PRODUK	HARGA	FOTO	JUMLAH	TOTAL
1	PRODZ	PRAWEDDING	3,000,000		1Barang	3,000,000

Gambar 4.40 Form Produk Transaksi Penjualan

4.3.12 Halaman Produk Terlaris

Pada halaman Produk Terlaris, Admin akan ditampilkan seluruh Produk Terlaris menurut penilaian yang diberikan oleh pelanggan, Berikut ini adalah tampilan dari Data Produk Terlaris



NO	KODE PRODUK	NAMA PRODUK	HARGA	NAMA VENDOR	TOTAL TRANSAKSI	NILAI
1	PREWEDDING	PRO012	3,000,000	BARURUH	0x Transaksi	0x

Gambar 4.41 Produk Terlaris

Segmen program 4.7 Produk Terlaris

```
@csrf_exempt
def vprodukterlaris(request):
    with connection.cursor() as con:
        con.execute("SELECT d.*,u.nama_user as vendor,u.nik
FROM app_detailtransaksi d LEFT JOIN app_transaksi t ON t.ko
de_transaksi=d.kode_transaksi LEFT JOIN app_user u ON t.vend
or=u.nik GROUP BY d.kode_produk ORDER BY d.jumlah DESC")
        produk = {}
        data = []
        jum = ""
        for vl in con.fetchall():
            with connection.cursor() as total:
                total.execute(
                    "SELECT COALESCE(SUM(jumlah),0) FROM app
_detailtransaksi WHERE kode_produk=%s", [vl[2]])
                ttl = total.fetchone()
                jum = int(ttl[0])
                produk = {"label": vl[1].upper(), "value": jum}
                data.append(produk)
        output = json.dumps(
            {"data": data})
```

Segmen program 4.7 Produk Terlaris(lanjutan)

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

4.3.13 Halaman Laporan Penjualan

Pada halaman Laporan Penjualan, Admin akan ditampilkan seluruh Laporan Penjualan vendor, Berikut ini adalah tampilan dari Data Laporan Penjualan.

The screenshot shows a web interface for generating sales reports. At the top right, there is a 'Print Laporan' button. Below it, the 'Transaksi Penjualan' section has three radio buttons: 'Harian' (selected), 'Bulanan', and 'Tahunan'. The 'Tanggal Transaksi' section contains two date input fields, both set to '2021-09-10', and a dropdown menu for location set to 'BARURUH'. A 'Print Data' button is located below the filters. At the bottom right of the form area, there is a 'Laporan Penjualan' button. Below the form is a table with a search bar and a 'Show 10 entries' indicator. The table header includes columns for 'NO', 'NIK USER', 'NAMA USER', 'KONTAK', 'ALAMAT', 'NAMA VENDOR', 'TOTAL PRODUK', 'BAYAR DP', 'STATUS PEMBAYARAN', 'TOTAL TRANSAKSI', and 'TGL AGARA'. The table body is currently empty.

Gambar 4.42 Form Laporan Penjualan

4.4 Uji Coba

Pada penelitian ini terdapat dua proses tes uji sistem. Adapun dua proses tes uji sistem tersebut di antaranya adalah *Testing Internal* dan *Testing External*. Pihak yang menangani *Testing Internal* diajukan kepada pihak yang ahli dalam bidang komputer. Sedangkan pada *Testing External* tidak lain adalah dari pihak *Wedding Organizer*. Berikut merupakan perolehan hasil dari uji coba dengan keterangan sebagai berikut :

4.4.1 Hasil Internal Testing

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

Tabel 4.6. Draft Hasil *Internal Testing*

No	Event	Hasil Yang Diharapkan	Pernyataan	
			Ya	Tidak
1	Apakah sistem Verifikasi akun sesuai dengan kebutuhan ?	Sistem Verifikasi akun sesuai dengan kebutuhan	√	
2	Apakah memblokir akun vendor sesuai dengan kebutuhan ?	Sistem memblokir akun vendor sesuai dengan kebutuhan	√	
3	Apakah sistem menampilkan data vendor sesuai dengan kebutuhan ?	Sistem menampilkan data vendor sesuai dengan kebutuhan	√	
4	Apakah sistem menampilkan data user sesuai dengan kebutuhan?	Sistem menampilkan data user sesuai dengan kebutuhan	√	
5	Apakah sistem menampilkan produk vendor sesuai dengan kebutuhan?	Sistem menampilkan produk vendor sesuai dengan kebutuhan	√	
6	Apakah sistem menampilkan transaksi pemesanan sesuai dengan kebutuhan?	Sistem menampilkan transaksi pemesanan sesuai dengan kebutuhan	√	
7	Apakah sistem menambah dan menghapus kategori produk sesuai dengan kebutuhan?	Sistem menambah dan menghapus kategori produk sesuai dengan kebutuhan	√	

Tabel 4.6. Draft Hasil *Internal Testing*(lanjutan)

No	Event	Hasil Yang Diharapkan	Pernyataan	
			Ya	Tidak
8	Apakah sistem menampilkan grafik transaksi penjualan terlaris menurut penilaian pelanggan?	Sistem menampilkan grafik transaksi penjualan terlaris menurut penilaian pelanggan	√	
9	Apakah sistem menampilkan statistik produk terlaris dari semua Vendor?	Sistem menampilkan statistik produk terlaris dari semua Vendor	√	
10	Apakah sistem menampilkan data produk terlaris sesuai dengan kebutuhan?	Sistem menampilkan data produk terlaris sesuai dengan kebutuhan	√	

4.4.2 Hasil *External Testing*

Pengujian yang dilakukan langsung kepada Wedding Organizer sebanyak 5 WO dengan dibagikan kuesioner secara langsung. Terdapat 5 pernyataan pada tabel pengujian ini yang dapat dilihat pada tabel dibawah ini :

Tabel 4.7. Draft Hasil *External Testing*

No	Draf Pertanyaan	Jawaban				
		SB	B	C B	TB	STB
1.	Apakah dengan adanya aplikasi ini pihak WO serta masyarakat lebih mudah dalam penyewaan WO?	5				

Tabel 4.7. Draft Hasil *External Testing*(lanjutan)

No	Draf Pertanyaan	Jawaban				
		SB	B	C B	TB	STB
2	Apakah fitur-fitur yang digunakan telah sesuai dengan kebutuhan?	1	4			
3	Apakah sistem ini membutuhkan fitur-fitur baru untuk mendukung keadaan sistem?		4	1		
4	Apakah sistem ini telah memenuhi standart keamanan?	3	2			
5	Apakah sistem ini telah layak digunakan?	1	4			

Tabel 4.8. Perhitungan skala likert

No	Skala Jawaban	Skor
1	Sangat Baik (SB)	5 x 10 = 50
2	Baik (B)	4 x 14 = 56
3	Cukup Baik(C)	3 x 1 = 3
4	Tidak Baik (TB)	2 x 0 = 0
5	Sangat Tidak Baik (STB)	1 x 0 = 0
		Total = 109

Skor tertinggi(Y) = skor tertinggi x jumlah responden

$$Y = 5 \times 5 = 25$$

Skor terendah(X) = skor terendah x jumlah responden

$$X = 1 \times 0 = 0$$

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

$$5 \times 5 \times 5 = 125$$

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

Tabel 4.9. Kriteria Interpretasi persen

No	Persentase	Keterangan
1	0% - 19,99%	Sangat Tidak Baik
2	20% - 39,99%	Tidak Baik
3	40% - 59,99%	Cukup
4	60% - 79,99%	Baik
5	80% - 100%	Sangat Baik

Rumus penentuan nilai presentase dari kuesioner yang peneliti lakukan adalah Penentuan nilai presentase = (Total Skor yang diperoleh / Skor ideal) x 100%, Sehingga di dapat nilai presentase = (109/125) x 100% = 87,2%. Berdasarkan tabel presentase nilai diatas dapat diambil kesimpulan bahwa Aplikasi sistem informasi monitoring dan evaluasi *Wedding Organizer* berbasis web *framework django* ini sangat baik.

4.5 Pemeliharaan (*Maintenance*)

Setelah observasi permasalahan, analisis sistem yang dilanjut dengan pengujian terhadap sistem baru, maka tahap terakhir dalam suatu penelitian adalah melakukan perawatan (*Maintenance*) terhadap sistem tersebut. Hal ini bertujuan untuk mengetahui adanya kesalahan dan kekurangan yang kemungkinan tidak dijumpai sebelumnya. Selain itu juga berguna dalam penambahan fitur baru nantinya sebagai bentuk pengembangan sistem.