

BAB 4

HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

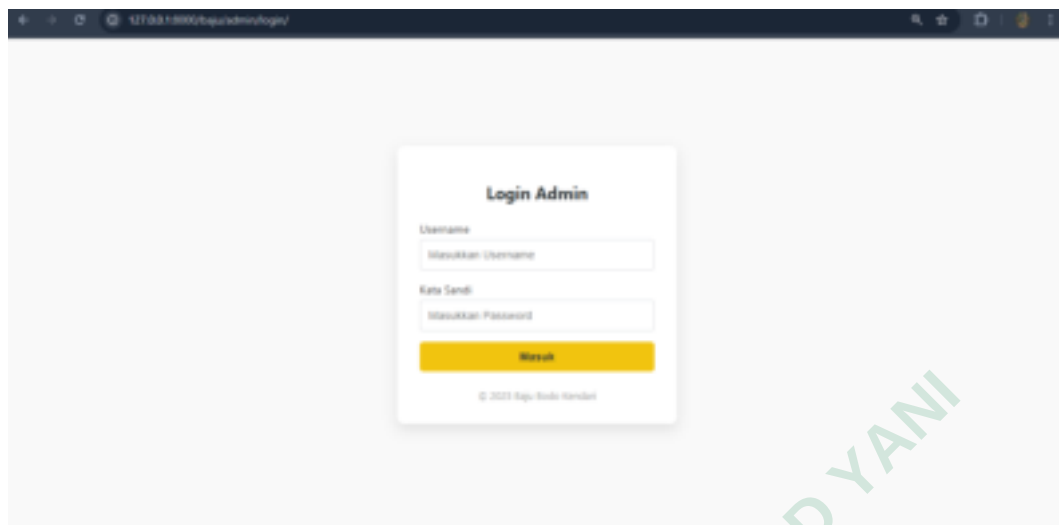
Penelitian ini menghasilkan sistem informasi penyewaan pakaian adat berbasis web interaktif untuk Toko Baju Bodo Kendari sebagai solusi atas kendala operasional yang selama ini dilakukan secara manual, seperti pencatatan melalui WhatsApp dan Excel serta pengecekan stok tanpa sistem digital. Sistem dikembangkan menggunakan Django dan MySQL, serta dilengkapi dengan fitur manajemen stok otomatis, reservasi *online*, dan chatbot berbasis *Natural Language Processing (NLP)* untuk melayani pertanyaan pelanggan secara otomatis. Proses pengembangan mengikuti metode Waterfall yang mencakup analisis kebutuhan, desain, implementasi, dan pengujian. Hasil pengujian menunjukkan sistem berjalan dengan baik dan efektif dalam meningkatkan efisiensi operasional, keakuratan data, serta kualitas pelayanan penyewaan.

4.2 IMPLEMENTASI DESAIN ANTARMUKA

Desain antarmuka sistem informasi penyewaan pakaian adat berbasis web pada Toko Baju Bodo Kendari dikembangkan menggunakan framework Django dengan bahasa pemrograman Python serta database MySQL. Antarmuka ini dirancang agar sederhana, responsif, dan mudah digunakan oleh pengguna dengan pembagian akses antara admin dan pelanggan. Berikut adalah implementasi dari beberapa halaman penting dalam sistem.

4.2.1 Implementasi Halaman Login Admin

Halaman pertama yang muncul dalam sistem adalah tampilan login admin. Pada halaman ini, admin diminta untuk mengisi *username* dan kata sandi untuk dapat mengakses dashboard pengelolaan. Tampilan ini memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses area administratif. Gambar antarmuka login admin dapat dilihat pada Gambar 4.1.



Gambar 4.1 Halaman Login Admin

Cuplikan kode login admin yang digunakan dalam sistem ini dapat dilihat sebagai berikut:

```
# views.py
def login_admin(request):
    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')

        admin_user =
Admin.objects.filter(Username=username).first()

        if admin_user:
            if check_password(password, admin_user.Password):
                request.session['admin_id'] = admin_user.id
                request.session['admin_username'] =
admin_user.Username
                return redirect('Dashboard')
            else:
                messages.error(request, "Password salah.")
        else:
            messages.error(request, "Username tidak ditemukan.")
    return render(request, 'baju/login_admin.html')
```

Kode di atas bertugas memproses permintaan login admin. Fungsi `check_password()` digunakan untuk mencocokkan *password* yang dimasukkan dengan *hash* yang disimpan di basis data. Jika valid, sesi disimpan dan pengguna diarahkan ke halaman Dashboard. Struktur model untuk tabel admin adalah sebagai berikut:

```
# models.py
class Admin(models.Model):
    id = models.AutoField(primary_key=True)
    name = models.CharField(max_length=100)
    Username = models.CharField(max_length=100)
    Password = models.CharField(max_length=100)
    image = models.ImageField(upload_to='admin/', blank=True,
null=True)

    class Meta:
        db_table = 'admin'
        managed = False
```

Model ini merepresentasikan data administrator, termasuk informasi login dan foto profilnya.

4.2.2 Implementasi Halaman Belum Dibayar Atau Sewa Menunggu Pembayaran

Setelah login, admin dapat mengakses menu “Sewa > Belum Dibayar” yang menampilkan daftar penyewaan yang belum melakukan pembayaran. Informasi yang ditampilkan meliputi kode sewa, nama baju, tanggal sewa, total biaya, nama penyewa, status pembayaran, serta opsi untuk mengubah status. Gambar 4.2 menunjukkan tampilan halaman ini.

No	Kode Sewa	Baju	Tgl. Mulai	Tgl. Selesai	Total	Pengira	Bukti Transfer	Status	Aksi
1	ADL000001	Baju Anak Perempuan Baju Boking Haju	June 16, 2025	June 17, 2025	Rp140.000	akibat	Belum ada	Menunggu Pembayaran	Ubah stat.

Gambar 4.2 Halaman Sewa Menunggu Pembayaran

Cuplikan kode yang menangani halaman ini adalah sebagai berikut:

```
# views.py
def sewa_bayar(request):
    sewa_list = Booking.objects.filter(status='Menunggu
    Pembayaran').select_related('id_baju', 'id_user')
    for s in sewa_list:
        s.total_harga = s.durasi * s.id_baju.harga
    return render(request, 'penyewaan/sewa_bayar.html',
    {'sewa_list': sewa_list})
```

Fungsi ini mengambil data penyewaan yang belum dibayar, menghitung total harga berdasarkan durasi sewa dan harga baju, kemudian menampilkannya di halaman sewa_bayar.html.

4.2.3 Implementasi Halaman Sedang Diswa Atau Sewa Menunggu Konfirmasi

Setelah pelanggan mengunggah bukti pembayaran, status penyewaan akan berpindah ke halaman “Sewa Menunggu Konfirmasi”. Di halaman ini, admin dapat memverifikasi pembayaran, menetapkan status sewa menjadi disetujui atau menambahkan denda jika diperlukan. Gambar 4.3 menunjukkan tampilan antarmuka halaman ini.



Gambar 4.3 Halaman Sewa Menunggu Konfirmasi

Cuplikan kode program untuk menampilkan halaman ini:

```
# views.py
def sewa_konfirmasi(request):
    sewa_list =
    Booking.objects.filter(status='Diterima').select_related('id_baju'
    , 'id_user')
    for s in sewa_list:
        s.total_harga = s.durasi * s.id_baju.harga
```

```
return render(request, 'penyewaan/sewa_konfirmasi.html',
{'sewa_list': sewa_list})
```

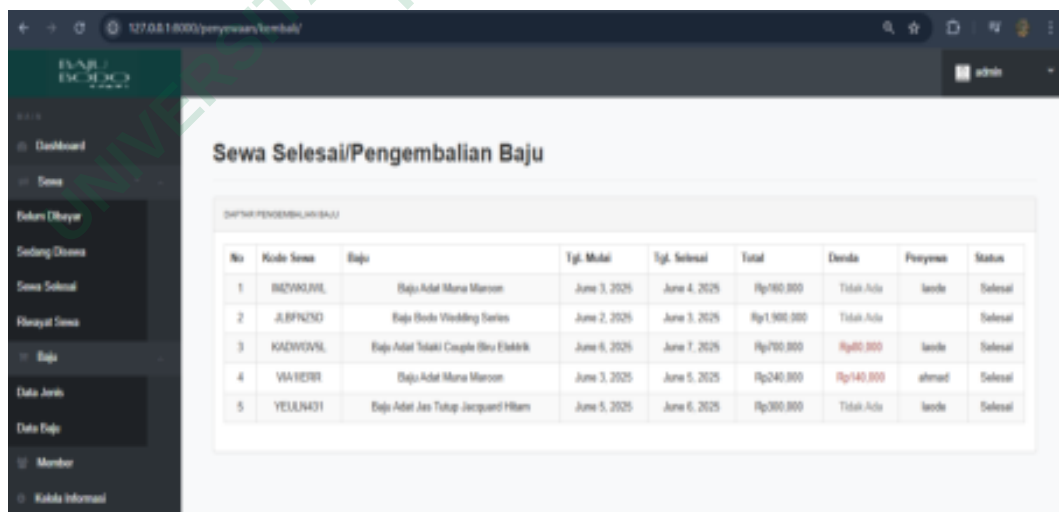
Kode untuk menyimpan denda manual bila terjadi keterlambatan pengembalian:

```
# views.py
@require_POST
def simpan_denda(request, kode_booking):
    booking = get_object_or_404(Booking,
    kode_booking=kode_booking)
    denda_per_hari = int(request.POST.get('denda_per_hari', 0))
    hari_terlambat = (timezone.now().date() -
    booking.tgl_selesai).days
    booking.denda = max(0, denda_per_hari * hari_terlambat)
    booking.save()
```

Fungsi `simpan_denda` akan menghitung jumlah hari keterlambatan, lalu mengalikan dengan denda harian untuk menghasilkan total denda. Jika tidak terlambat, denda akan tetap 0.

4.2.4 Implementasi Halaman Sewa Selesai atau Pengembalian Baju

Halaman ini menampilkan daftar pengembalian baju dari pelanggan, termasuk informasi mengenai kode sewa, nama baju, tanggal peminjaman, tanggal kembali, biaya sewa, denda jika ada, serta status penyewaan. Admin dapat memantau penyewaan yang telah diselesaikan dan mencatat denda secara manual jika keterlambatan terjadi. Gambar 4.4 memperlihatkan tampilan halaman ini.



No	Kode Sewa	Baju	Tgl. Mulai	Tgl. Selesai	Total	Denda	Pembayaran	Status
1	BAGVKNJL	Baju Adet Muna Maroon	June 3, 2025	June 4, 2025	Rp180.000	Tidak Ada	lunas	Selesai
2	JLBPXNSD	Baju Bodo Wedding Series	June 2, 2025	June 3, 2025	Rp1.900.000	Tidak Ada	lunas	Selesai
3	KADWYVSL	Baju Adet Tolaki Couple Biru Elektrik	June 6, 2025	June 7, 2025	Rp700.000	Rp600.000	lunas	Selesai
4	VAAHEDRI	Baju Adet Muna Maroon	June 3, 2025	June 5, 2025	Rp240.000	Rp140.000	ahmad	Selesai
5	YELUN431	Baju Adet Jas Tugap Jacquard Hitam	June 5, 2025	June 6, 2025	Rp300.000	Tidak Ada	lunas	Selesai

Gambar 4.4 Halaman Pengembalian Baju

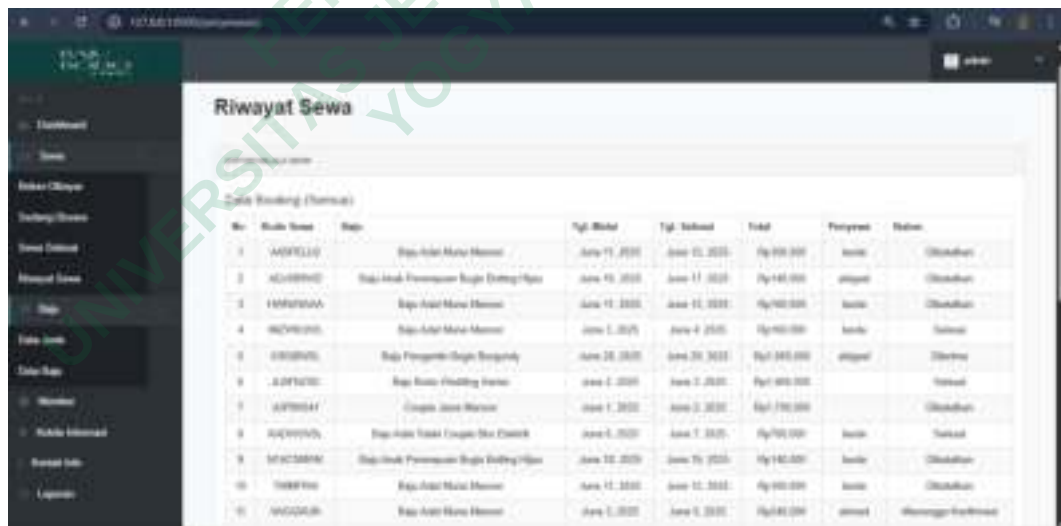
Kode program yang digunakan:

```
# views.py
def sewa_kembali(request):
    sewa_list =
Booking.objects.filter(status='Selesai').select_related('id_baju',
'id_user')
    for s in sewa_list:
        s.total_harga = s.durasi * s.id_baju.harga
    return render(request, 'penyewaan/sewa_kembali.html',
{'sewa_list': sewa_list})
```

Halaman ini berfungsi untuk memudahkan admin melakukan pemantauan terhadap transaksi yang telah ditutup dan mencatat penyewa yang telah mengembalikan baju.

4.2.5 Implementasi Halaman Riwayat Sewa

Admin dapat melihat seluruh riwayat pemesanan dalam halaman “Riwayat Sewa”. Halaman ini berisi daftar lengkap penyewaan, termasuk yang dibatalkan, masih berlangsung, atau telah selesai. Setiap entri mencakup kode sewa, nama baju, tanggal penyewaan, nama penyewa, total biaya, dan status pemesanan. Halaman ini mendukung pemantauan menyeluruh terhadap seluruh aktivitas sewa. Tampilan halaman dapat dilihat pada Gambar 4.5.



No	Kode Sewa	Baju	Tgl. Mulai	Tgl. Selesai	Total	Pembayaran	Status
1	A000110	Baju Adas Mada Hitam	Juni 11, 2020	Juni 12, 2020	Rp 100.000	Selesai	Selesai
2	A000090	Baju Adas Pemasukan Baju Daring Hitam	Juni 10, 2020	Juni 11, 2020	Rp 140.000	Selesai	Selesai
3	A000000A	Baju Adas Mada Hitam	Juni 11, 2020	Juni 12, 2020	Rp 100.000	Selesai	Selesai
4	A000000	Baju Adas Mada Hitam	Juni 1, 2020	Juni 4, 2020	Rp 400.000	Selesai	Selesai
5	A000000	Baju Pemasukan Baju Daring Hitam	Juni 20, 2020	Juni 20, 2020	Rp 140.000	Selesai	Selesai
6	A000000	Baju Baju Pemasukan Baju Daring Hitam	Juni 1, 2020	Juni 2, 2020	Rp 140.000	Selesai	Selesai
7	A000000	Celana Jeans Hitam	Juni 1, 2020	Juni 2, 2020	Rp 100.000	Selesai	Selesai
8	A000000	Baju Adas Mada Celana Jeans Hitam	Juni 1, 2020	Juni 2, 2020	Rp 100.000	Selesai	Selesai
9	A000000	Baju Adas Pemasukan Baju Daring Hitam	Juni 10, 2020	Juni 11, 2020	Rp 140.000	Selesai	Selesai
10	A000000	Baju Adas Mada Hitam	Juni 11, 2020	Juni 12, 2020	Rp 100.000	Selesai	Selesai
11	A000000	Baju Adas Mada Hitam	Juni 1, 2020	Juni 2, 2020	Rp 100.000	Selesai	Selesai

Gambar 4.5 Halaman Kelola Sewa

Kode program yang digunakan:

```
# views.py
def sewa(request):
    # Pembatalan otomatis jika lebih dari 6 jam belum dibayar
```

```

    batas_exp = timezone.now() - timedelta(hours=6)
    Booking.objects.filter(
        status='Menunggu Pembayaran',
        bukti_bayar='',
        tgl_booking__lte=batas_exp.date()
    ).update(status='Dibatalkan')

    sewa_list = Booking.objects.all().select_related('id_baju',
    'id_user')
    for s in sewa_list:
        s.total_harga = s.durasi * s.id_baju.harga

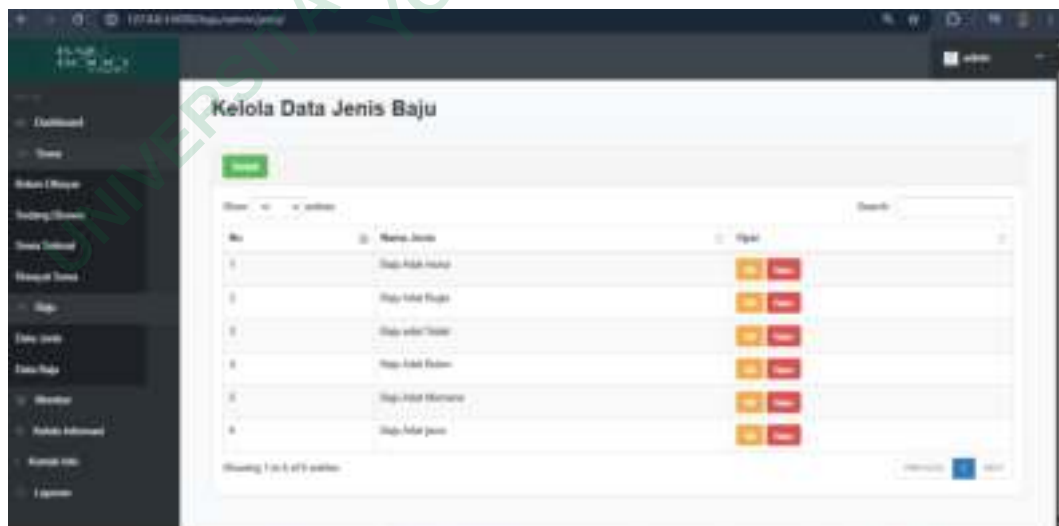
    return render(request, 'penyewaan/sewa.html', {'sewa_list':
    sewa_list})

```

Fungsi ini menampilkan semua data penyewaan dari semua status: dibatalkan, belum dibayar, sedang disewa, atau sudah selesai. Juga terdapat logika pembatalan otomatis berdasarkan waktu.

4.2.6 Implementasi Halaman Kelola Data Jenis Baju

Halaman ini digunakan oleh admin untuk mengelola kategori jenis baju adat yang tersedia dalam sistem. Admin dapat menambahkan jenis baru, mengedit jenis yang sudah ada, atau menghapus entri yang tidak lagi digunakan. Setiap data ditampilkan dalam bentuk tabel dengan kolom nomor, nama jenis baju, dan aksi. Gambar 4.6 menunjukkan tampilan halaman ini.



Gambar 4.6 Halaman Kelola Baju

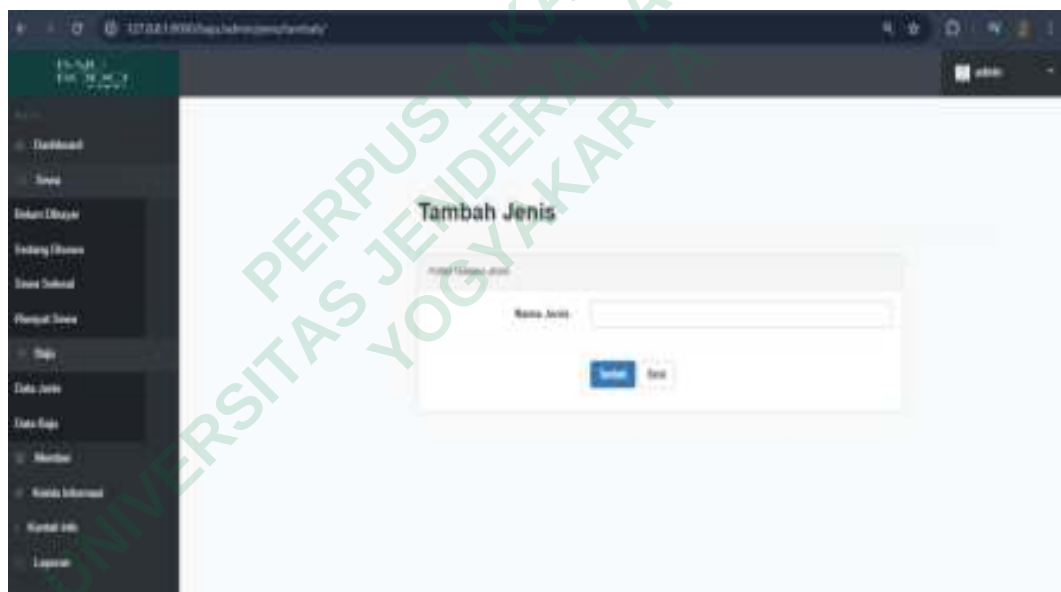
Cuplikan kode berikut digunakan untuk menampilkan data jenis baju:

```
# views.py
def daftar_jenis(request):
    data_jenis = Jenis.objects.all()
    return render(request, 'baju/jenis.html', {'data_jenis':
data_jenis})
```

Fungsi di atas mengambil semua data dari model Jenis dan menampilkannya melalui template HTML.

4.2.7 Implementasi Halaman Tambah Jenis Baju

Untuk menambahkan kategori jenis baju baru, sistem menyediakan formulir sederhana berupa input teks untuk nama jenis. Setelah data diisi, admin cukup menekan tombol “Tambah” untuk menyimpan data baru ke dalam sistem. Antarmuka halaman ini ditampilkan pada Gambar 4.7.



Gambar 4.7 Halaman Tambah Jenis Baju

Kode program untuk fitur ini:

```
# views.py
def tambah_jenis(request):
    if request.method == 'POST':
        form = JenisForm(request.POST)
        if form.is_valid():
            form.save()
            messages.success(request, "Jenis baju berhasil
ditambahkan.")
            return redirect('jenis')
```

4.2.8 Implementasi Halaman Edit Jenis Baju

Kode program halaman edit jenis:

```
# views.py
def edit_jenis(request, id_jenis):
    jenis = get_object_or_404(Jenis, pk=id_jenis)
    if request.method == 'POST':
        form = JenisForm(request.POST, instance=jenis)
        if form.is_valid():
            form.save()
            messages.success(request, "Data jenis berhasil diperbarui.")
            return redirect('jenis')
    else:
        form = JenisForm(instance=jenis)
        return render(request, 'baju/jenis_edit.html', {'form': form, 'jenis': jenis})
```

Fungsi ini memuat data awal, memproses input dari user, dan menyimpan hasil edit ke database.

4.2.9 Implementasi Halaman Kelola Data Baju

Halaman ini menampilkan daftar lengkap baju adat yang tersedia untuk disewa. Informasi yang disajikan meliputi nama baju, jenis, kategori, harga sewa per hari, serta tombol aksi seperti detail, ubah, dan hapus. Halaman ini membantu admin dalam memantau dan mengatur seluruh inventaris baju. Tampilan halaman ini disajikan pada Gambar 4.9.

No.	Nama Baju	Jenis	Kategori	Harga Sewa	Aksi
1	Baju Adat Melayu Melayu	Baju Adat Melayu	Ornamen Peranginan	Rp 50.000	Detail Ubah Hapus
2	Baju Adat Melayu	Baju Adat Melayu	Ornamen Peranginan	Rp 50.000	Detail Ubah Hapus
3	Baju Adat Melayu	Baju Adat Melayu	Ornamen Peranginan	Rp 50.000	Detail Ubah Hapus
4	Baju Adat Melayu	Baju Adat Melayu	Ornamen Peranginan	Rp 50.000	Detail Ubah Hapus
5	Baju Adat Melayu	Baju Adat Melayu	Ornamen Peranginan	Rp 50.000	Detail Ubah Hapus
6	Baju Adat Melayu	Baju Adat Melayu	Ornamen Peranginan	Rp 50.000	Detail Ubah Hapus
7	Baju Adat Melayu	Baju Adat Melayu	Ornamen Peranginan	Rp 50.000	Detail Ubah Hapus
8	Baju Adat Melayu	Baju Adat Melayu	Ornamen Peranginan	Rp 50.000	Detail Ubah Hapus
9	Baju Adat Melayu	Baju Adat Melayu	Ornamen Peranginan	Rp 50.000	Detail Ubah Hapus

Gambar 4.9 Halaman Kelola Data Baju

Cuplikan kode berikut digunakan untuk mengambil data baju:

```
# views.py
def daftar_baju(request):
    data_baju = Baju.objects.all()
    return render(request, 'baju/baju.html', {'data_baju':
data_baju})
```

Model Baju memiliki relasi ke Jenis dan menampilkan data lengkap, termasuk stok ukuran dan gambar.

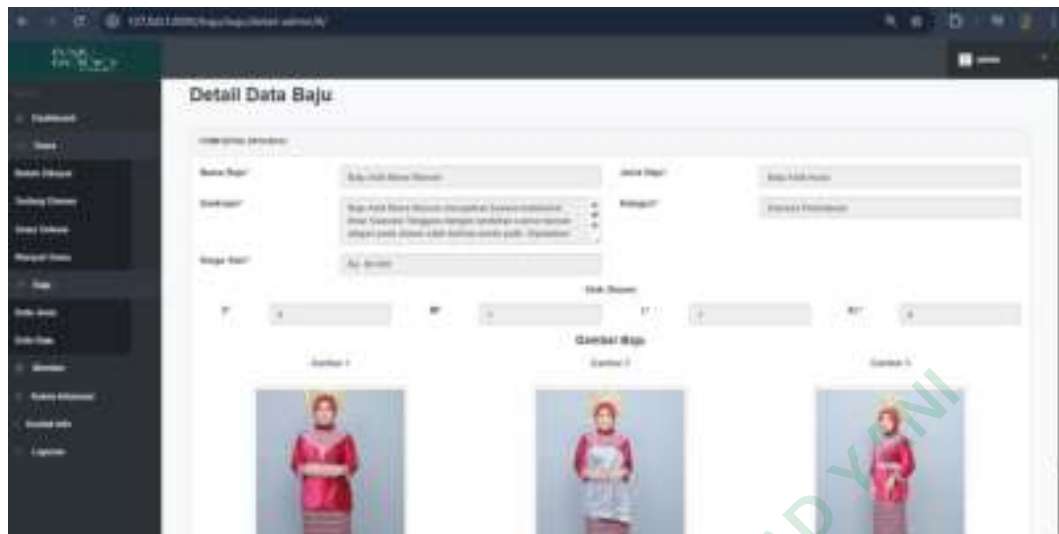
4.2.10 Implementasi Halaman Tambah Data Baju

Halaman ini menyediakan formulir bagi admin untuk menambahkan data baju ke sistem, mencakup nama, deskripsi, harga, jenis, kategori, stok per ukuran (S, M, L, XL), dan unggahan gambar. Tampilan formulir ditunjukkan pada Gambar 4.10.

Kode program berikut menangani proses tambah data baju:

Form BajuForm merupakan turunan ModelForm dari model Baju, dan mengatur atribut penting seperti gambar, harga, serta kategori.

Halaman ini menampilkan informasi lengkap dari satu baju adat, seperti nama, deskripsi, harga, jenis, kategori, stok ukuran, serta tiga gambar pendukung. Tampilan ini ditunjukkan pada Gambar 4.11.



Gambar 4.11 Halaman Detail Data Baju

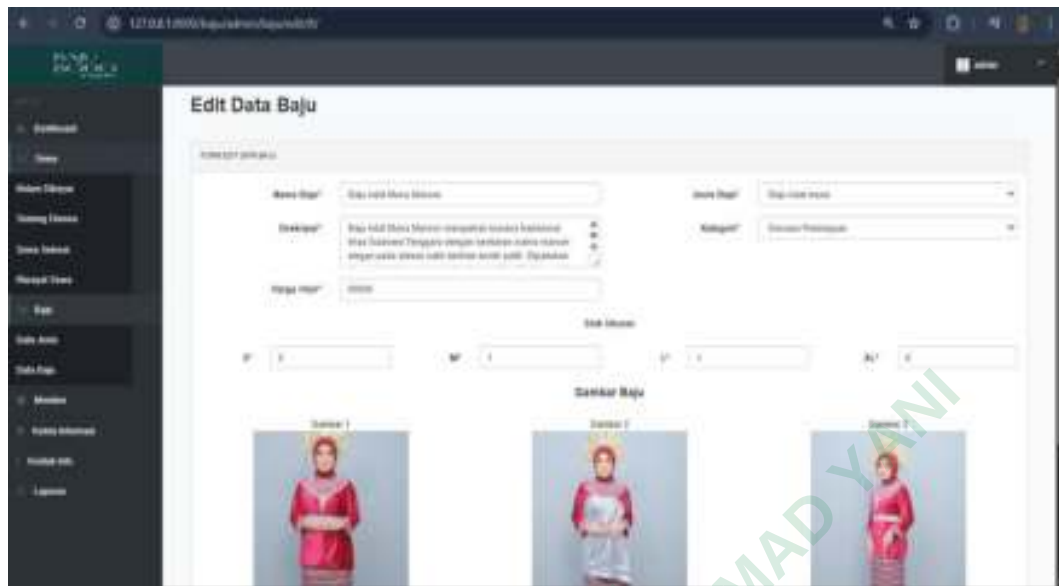
Cuplikan kode untuk menampilkan detail baju adalah sebagai berikut:

```
# views.py
def detail_baju_admin(request, id_baju):
    baju = get_object_or_404(Baju, id_baju=id_baju)
    return render(request, 'baju/detail_baju.html', {'baju': baju})
```

Kode ini mengambil data berdasarkan ID baju, kemudian menampilkannya di halaman detail_baju.html dengan seluruh informasi yang terkait.

4.2.12 Implementasi Halaman Edit Data Baju

Halaman ini digunakan oleh admin untuk memperbarui informasi baju adat yang telah terdaftar dalam sistem. Perubahan dapat dilakukan apabila terdapat pembaruan harga, penyesuaian deskripsi, perubahan kategori, atau penambahan stok ukuran tertentu. Selain itu, admin juga dapat mengganti gambar baju agar sesuai dengan kondisi terbaru di lapangan. Seluruh atribut seperti nama baju, jenis, kategori, harga sewa per hari, ukuran (S, M, L, XL), dan gambar dapat diubah melalui formulir yang tersedia. Fitur ini sangat penting untuk menjaga keakuratan dan keterbaruan informasi dalam sistem penyewaan, sehingga pelanggan selalu mendapatkan data yang relevan dan aktual saat melakukan pemesanan. Tampilan halaman edit data baju ditunjukkan pada Gambar 4.12.



Gambar 4.12 Halaman Edit Data Baju

Cuplikan kode yang digunakan dalam proses edit data baju adalah sebagai berikut:

```
# views.py
def edit_baju(request, id_baju):
    baju = get_object_or_404(Baju, pk=id_baju)
    if request.method == 'POST':
        form = BajuForm(request.POST, request.FILES, instance=baju)
        if form.is_valid():
            form.save()
            messages.success(request, "Data baju berhasil diperbarui.")
            return redirect('baju')
        else:
            form = BajuForm(instance=baju)
    return render(request, 'baju/baju_edit.html', {'form': form, 'baju': baju})
```

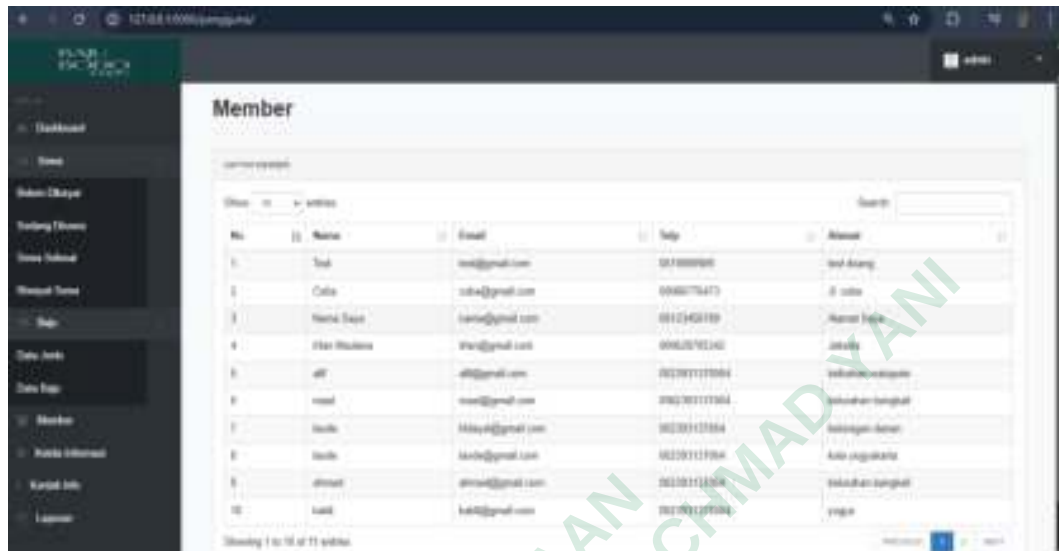
Formulir yang digunakan untuk mengelola input data adalah:

```
# forms.py
class BajuForm(forms.ModelForm):
    class Meta:
        model = Baju
        fields = 'all'
```

4.2.13 Implementasi Halaman Data *Member*

Halaman ini berfungsi untuk menampilkan daftar semua pengguna (*member*) yang telah mendaftar ke dalam sistem. Data disajikan dalam bentuk tabel yang mencakup nama, email, nomor telepon, dan alamat. Halaman ini sangat

berguna bagi admin untuk memantau data pengguna secara menyeluruh. Gambar 4.13 menampilkan antarmuka halaman data *member*.



Gambar 4.13 Halaman Data *Member*

Cuplikan kode yang digunakan untuk menampilkan data *member*:

```
# views.py
def data_member(request):
    members = Member.objects.all()
    return render(request, 'admin/member_list.html', {'members':
members})
```

Model untuk menyimpan informasi *member* adalah sebagai berikut:

```
# models.py
class User(models.Model):
    nama_user = models.CharField(max_length=100)
    email = models.EmailField(unique=True)
    telp = models.CharField(max_length=15)
    alamat = models.TextField()
```

4.2.14 Implementasi Halaman Kelola Informasi

Halaman kelola informasi digunakan oleh admin untuk memperbarui konten yang bersifat umum dan tampil secara publik pada halaman utama aplikasi. Konten ini meliputi syarat dan ketentuan penyewaan, aturan penggunaan baju, serta informasi rekening pembayaran. Halaman ini menyediakan form teks editor yang memudahkan pembaruan informasi secara berkala. Gambar 4.14 menunjukkan tampilan halaman kelola informasi.

Gambar 4.14 Halaman Kelola Informasi

Cuplikan kode program untuk menangani proses kelola informasi adalah sebagai berikut:

```
# views.py
def kelola_informasi(request):
    info, created = KelolaInformasi.objects.get_or_create(pk=1)
    if request.method == 'POST':
        form = KelolaInformasiForm(request.POST, instance=info)
        if form.is_valid():
            form.save()
            messages.success(request, 'Informasi berhasil disimpan.')
        else:
            messages.error(request, 'Terjadi kesalahan saat menyimpan.')
    else:
        form = KelolaInformasiForm(instance=info)
    return render(request, 'pages/kelola_informasi.html', {'form': form})
```

Form yang digunakan untuk mengelola informasi umum adalah sebagai berikut:

```
# forms.py
class KelolaInformasiForm(forms.ModelForm):
    class Meta:
        model = KelolaInformasi
        fields = ['syarat_ketentuan', 'aturan_pemakaian', 'rekening']
        widgets = {
            'syarat_ketentuan': forms.Textarea(attrs={'rows': 6}),
            'aturan_pemakaian': forms.Textarea(attrs={'rows': 6}),
            'rekening': forms.Textarea(attrs={'rows': 3}),
        }
```

Struktur model KelolaInformasi sebagai sumber data:

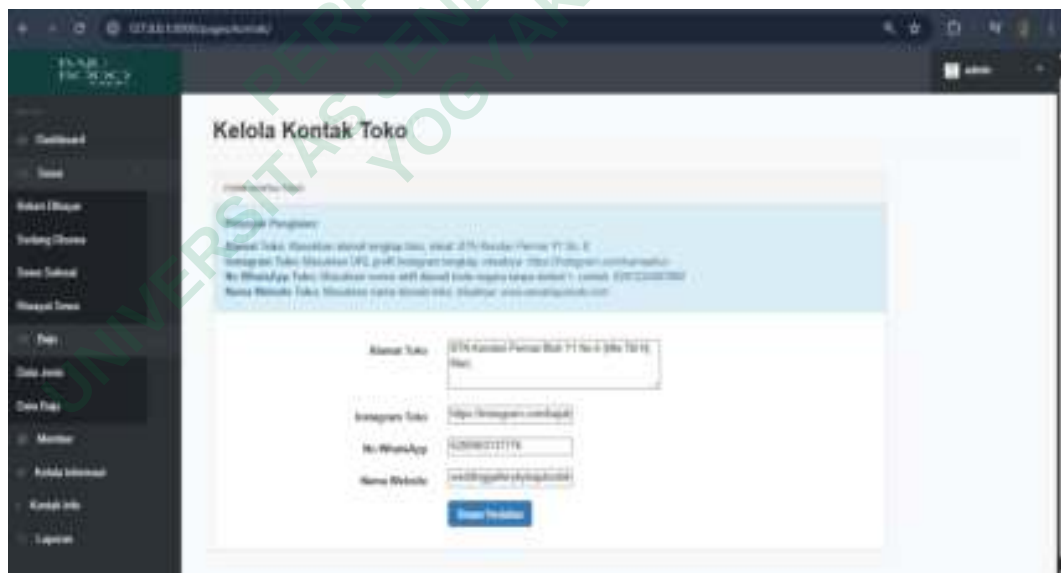
```
# models.py

class KelolaInformasi(models.Model):
    syarat_ketentuan = models.TextField()
    aturan_pemakaian = models.TextField()
    rekening = models.CharField(max_length=100)

    class Meta:
        db_table = 'kelola_informasi'
```

4.2.15 Implementasi Halaman Kontak Toko

Halaman ini memberikan fasilitas bagi admin untuk mengelola informasi kontak toko yang akan ditampilkan kepada pengguna di sisi *front-end*. Informasi yang dapat diatur meliputi alamat toko, akun media sosial seperti Instagram, nomor WhatsApp, serta alamat website resmi. Dengan adanya fitur ini, admin dapat memastikan bahwa data kontak yang ditampilkan selalu terkini, sehingga pelanggan dapat dengan mudah menghubungi toko untuk kebutuhan konsultasi, konfirmasi penyewaan, atau pertanyaan lainnya. Tampilan antarmuka halaman ini ditunjukkan pada Gambar 4.15.



Gambar 4.15 Halaman Kontak Toko

Cuplikan kode program yang digunakan untuk mengelola informasi kontak toko:

```
# views.py

def kelola_kontak(request):
    kontak = KontakToko.objects.first()
```

```

    if request.method == 'POST':
        form = KontakTokoForm(request.POST, instance=kontak)
        if form.is_valid():
            form.save()
            messages.success(request, 'Informasi kontak berhasil
diperbarui.')
            return redirect('kelola_kontak')
        else:
            form = KontakTokoForm(instance=kontak)
            return render(request, 'pages/kelola_kontak.html', {'form':
form})

```

Form input yang digunakan untuk halaman ini adalah:

```

# forms.py
class KontakTokoForm(forms.ModelForm):
    class Meta:
        model = KontakToko
        fields = '__all__'
        widgets = {
            'alamat_toko': forms.Textarea(attrs={'rows': 3}),
        }

```

Struktur model KontakToko yang menyimpan informasi kontak:

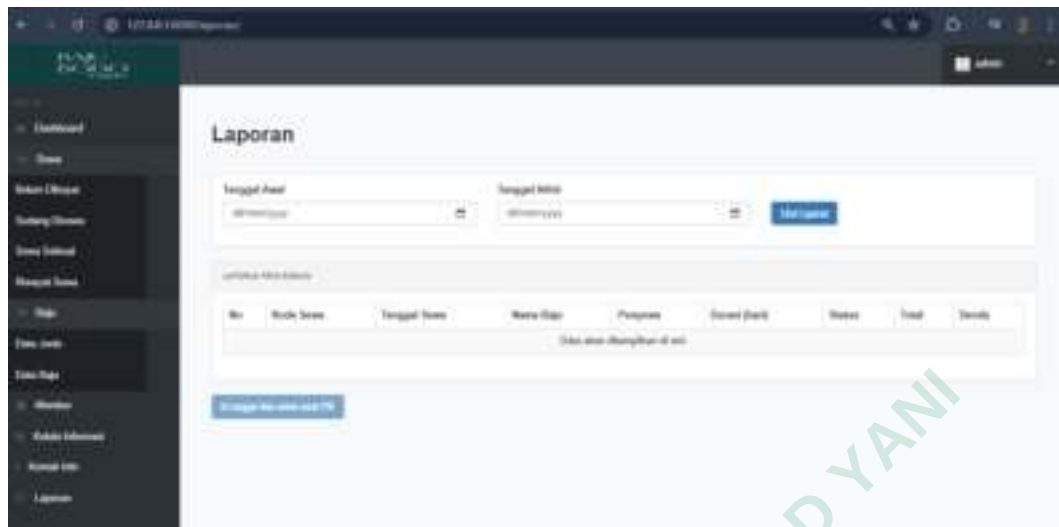
```

# models.py
class KontakToko(models.Model):
    alamat_toko = models.TextField()
    instagram_toko = models.CharField(max_length=255)
    no_whatsapp_toko = models.CharField(max_length=20)
    nama_website_toko = models.CharField(max_length=255)

```

4.2.16 Implementasi Halaman Laporan Penyewaan

Halaman laporan digunakan oleh admin untuk menampilkan dan mencetak data penyewaan berdasarkan periode tertentu, seperti harian, mingguan, atau bulanan. Sistem menyediakan input rentang tanggal yang dapat disesuaikan, dan hasil laporan ditampilkan dalam bentuk tabel transaksi lengkap, termasuk nama penyewa, kode *booking*, tanggal sewa, dan total biaya. Admin juga dapat mengekspor laporan ke dalam format PDF, menjadikannya praktis untuk kebutuhan arsip, pelaporan usaha, maupun audit internal. Tampilan halaman ini ditunjukkan pada Gambar 4.16.



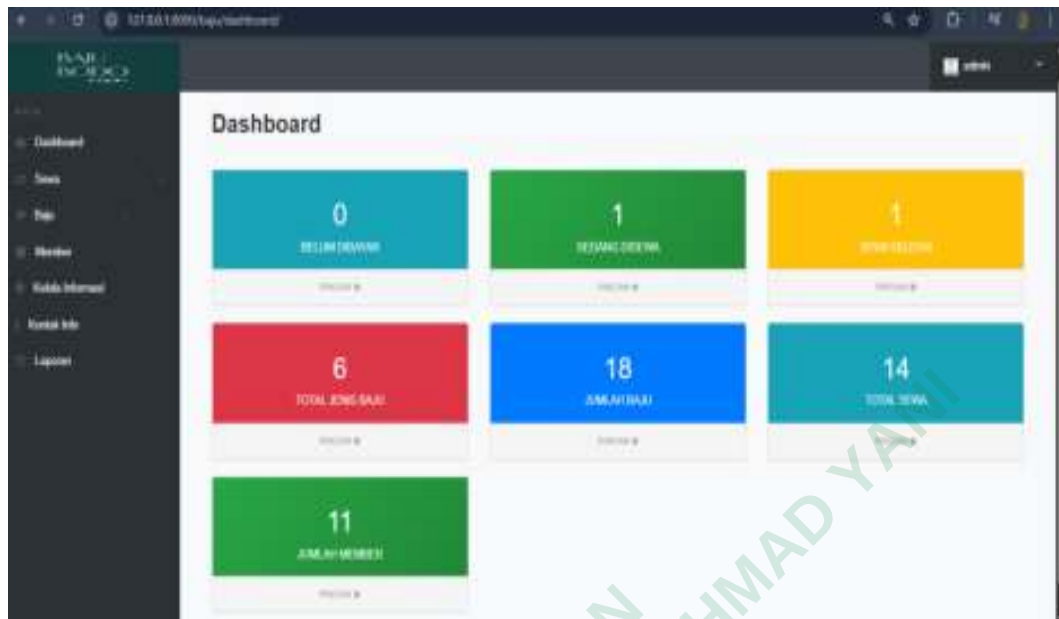
Gambar 4.16 Halaman Laporan Penyewaan

Cuplikan kode laporan penyewaan:

```
# views.py
def laporan_sewa(request):
    start_date = request.GET.get('start')
    end_date = request.GET.get('end')
    laporan =
Booking.objects.filter(tgl_booking__range=(start_date, end_date))
    return render(request, 'penyewaan/laporan.html', {'laporan':
laporan})
```

4.2.17 Implementasi Halaman Dashboard

Halaman dashboard merupakan tampilan utama setelah admin login, menampilkan statistik penyewaan seperti jumlah transaksi yang belum dibayar, sedang disewa, dan selesai. Selain itu, ditampilkan pula total jenis baju, jumlah stok, total transaksi, dan jumlah *member*. Setiap indikator dilengkapi tombol “Rincian” untuk melihat data lengkap. Desain visual yang berwarna memudahkan pemantauan sistem secara cepat. Gambar 4.17 menunjukkan tampilan halaman ini.



Gambar 4.17 Halaman Dashboard

Cuplikan kode program untuk menampilkan data Dashboard adalah sebagai berikut:

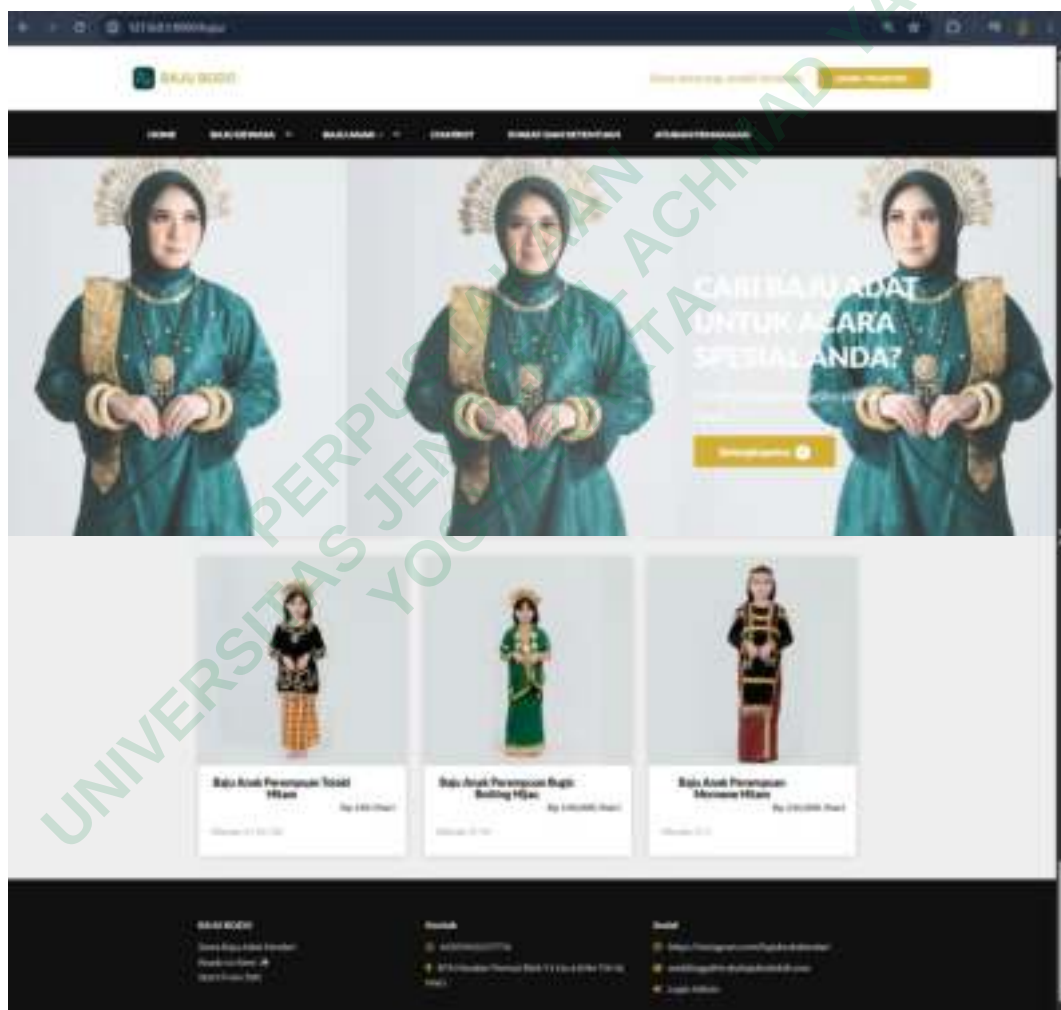
```
# views.py
def Dashboard_admin(request):
    total_baju = Baju.objects.count()
    total_member = Member.objects.count()
    total_booking = Booking.objects.count()
    total_selesai =
Booking.objects.filter(status='Selesai').count()
    total_belum_bayar = Booking.objects.filter(status='Menunggu
Pembayaran').count()
    total_diterima =
Booking.objects.filter(status='Diterima').count()

    return render(request, 'admin/Dashboard.html', {
        'total_baju': total_baju,
        'total_member': total_member,
        'total_booking': total_booking,
        'total_selesai': total_selesai,
        'total_belum_bayar': total_belum_bayar,
        'total_diterima': total_diterima,
    })
```

Kode di atas mengambil statistik dari beberapa model utama seperti Baju, Booking, dan User, kemudian dikirimkan ke template Dashboard.html untuk ditampilkan secara visual.

4.2.18 Implementasi Halaman Beranda Pelanggan

Halaman beranda merupakan tampilan awal yang dilihat oleh pengguna sebelum melakukan login. Halaman ini menyajikan antarmuka visual yang menarik dengan menampilkan gambar model baju adat serta slogan promosi. Terdapat menu navigasi yang mengarahkan pengguna ke kategori baju dewasa, baju anak, chatbot, serta halaman syarat dan ketentuan. Pengguna juga dapat memilih untuk masuk (login) atau mendaftar (*register*) melalui tombol khusus di bagian kanan atas. Gambar 4.18 menampilkan tampilan beranda.



Gambar 4.18 Halaman Beranda Sebelum Login

Cuplikan kode program:

```
# views.py
def beranda(request):
    baju_terbaru = Baju.objects.all().order_by('-id_baju')[:6]
```

```

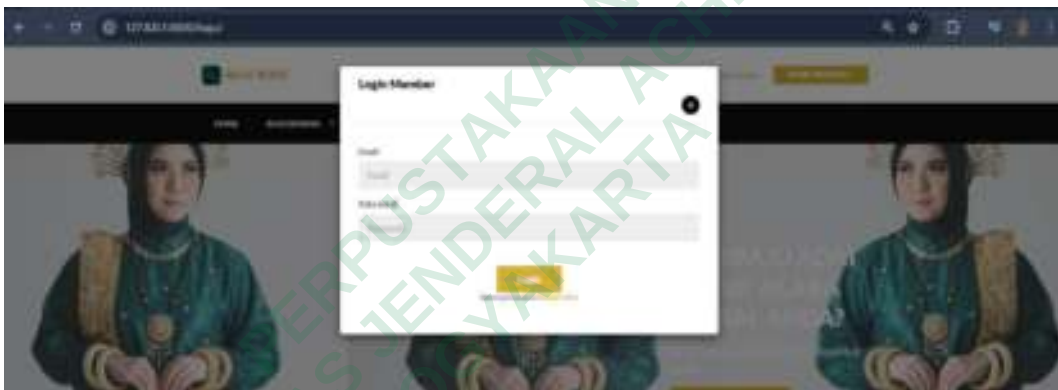
return render(request, 'baju/index.html', {
    'baju_terbaru': baju_terbaru
})

```

Fungsi di atas menampilkan daftar 6 baju terbaru di beranda, yang menjadi daya tarik utama bagi pengunjung pertama kali.

4.2.19 Implementasi Halaman Login Pelanggan

Untuk mengakses fitur penyewaan, pengguna diwajibkan melakukan login. Halaman login *member* ditampilkan dalam bentuk *modal popup* dengan input email dan kata sandi. Apabila pengguna belum memiliki akun, tersedia tautan untuk melakukan pendaftaran. Tampilan ini mendukung autentikasi sederhana namun efektif bagi pengguna. Gambar 4.19 memperlihatkan tampilan login *member*.



Gambar 4.19 Halaman Login *Member*

Cuplikan kode program untuk autentikasi:

```

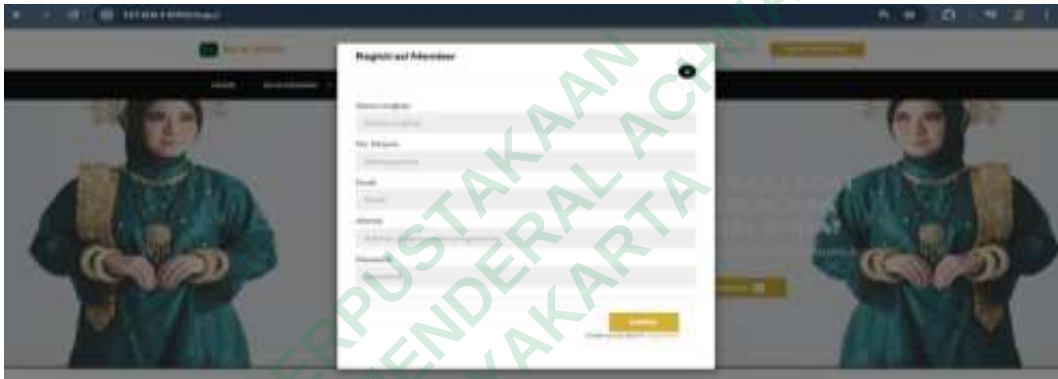
# views.py
def login_member(request):
    if request.method == 'POST':
        email = request.POST['email']
        password_input = request.POST['password']
        try:
            member = Member.objects.get(email=email)
            if check_password(password_input, member.password):
                request.session['id_user'] = member.id_user
                request.session['nama_user'] = member.nama_user
                return redirect('beranda')
            else:
                messages.error(request, 'Password salah.')
        except Member.DoesNotExist:
            messages.error(request, 'Email tidak ditemukan.')
    return redirect('beranda')

```

Autentikasi memanfaatkan `check_password` untuk membandingkan input dengan *password* yang tersimpan dalam database. Jika berhasil, sesi login disimpan dan pengguna diarahkan ke halaman utama

4.2.20 Implementasi Halaman Registrasi Pelanggan

Pengguna yang belum memiliki akun dapat mendaftar melalui formulir registrasi *member*. Formulir ini mencakup input untuk nama lengkap, nomor telepon, email, alamat, dan *password*. Proses registrasi dirancang agar cepat dan mudah, untuk mempercepat konversi pengguna menjadi pelanggan aktif. Tampilan formulir registrasi ditunjukkan pada Gambar 4.20.



Gambar 4.20 Halaman Registrasi *Member*

Cuplikan kode program registrasi:

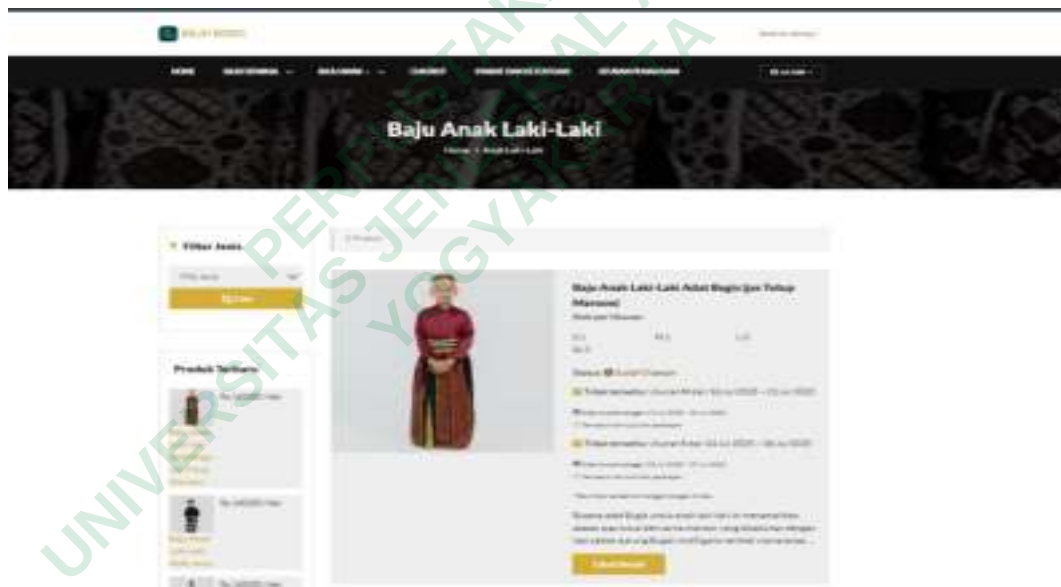
```
# forms.py
class RegisterForm(forms.ModelForm):
    password = forms.CharField(widget=forms.PasswordInput)
    class Meta:
        model = Member
        fields = ['nama_user', 'email', 'telp', 'alamat',
'password']
# views.py
def register_member(request):
    if request.method == 'POST':
        form = RegisterForm(request.POST)
        if form.is_valid():
            email = form.cleaned_data['email']
            if Member.objects.filter(email=email).exists():
                messages.error(request, 'Email sudah terdaftar.')
            else:
                Member.objects.create(
                    nama_user=form.cleaned_data['nama_user'],
                    email=email,
```

```
password=make_password(form.cleaned_data['password']),
                        telp=form.cleaned_data['telp'],
                        alamat=form.cleaned_data['alamat']
                    )
                    messages.success(request, 'Registrasi berhasil!')
                    return redirect('beranda')
                return redirect('beranda')
```

Registrasi dilakukan dengan menyimpan *password* yang sudah di-hash untuk keamanan. Sistem juga memvalidasi agar email tidak ganda.

4.2.21 Implementasi Halaman Katalog Baju

Setelah login, pengguna dapat mengakses halaman katalog berisi daftar baju adat yang tersedia lengkap dengan nama, harga sewa, dan ukuran. Tampilan grid interaktif memudahkan pencarian, dan desain responsif memungkinkan akses dari berbagai perangkat. Gambar 4.21 menunjukkan tampilan katalog.



Gambar 4.21 Halaman Katalog Baju

Cuplikan kode program:

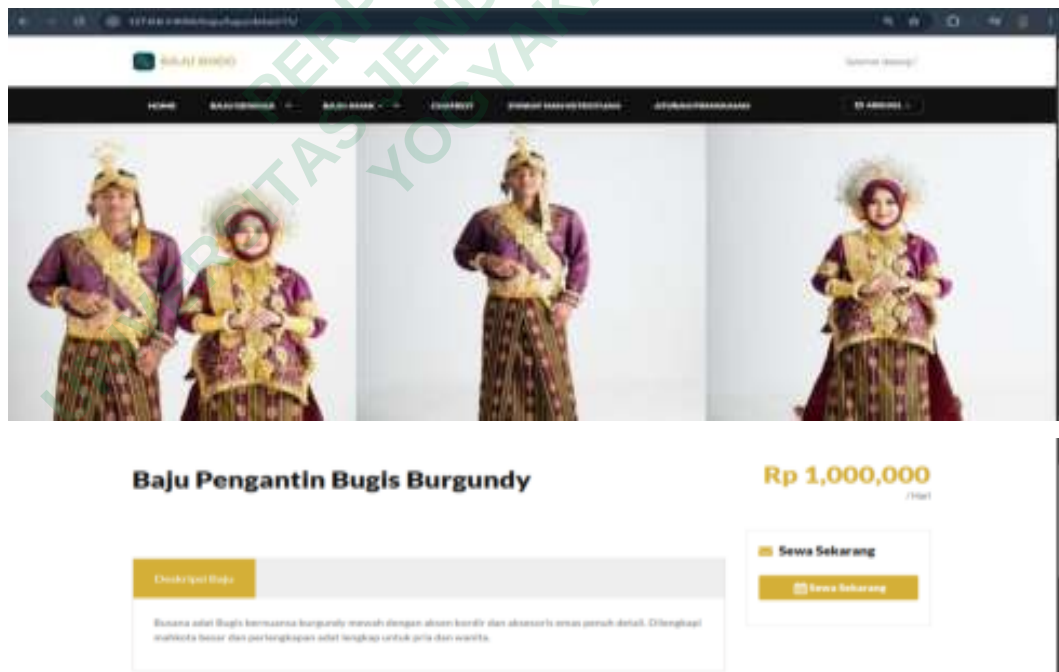
```
# views.py
def daftar_semua_baju(request):
    jenis_list = Jenis.objects.all()
    baju_list = Baju.objects.all().order_by('-id_baju')
    id_jenis = request.GET.get('jenis')
    if id_jenis:
        baju_list = baju_list.filter(id_jenis_id=id_jenis)
```

```
return render(request, 'baju/semua_baju.html', {
    'baju_list': baju_list,
    'jenis_list': jenis_list
})
```

Kode di atas memungkinkan pengguna memfilter baju berdasarkan jenis tertentu. Semua data ditampilkan secara interaktif dan responsif.

4.2.22 Implementasi Halaman Detail Baju

Halaman ini menyajikan informasi lengkap mengenai baju adat yang dipilih oleh pengguna. Data yang ditampilkan meliputi nama baju, deskripsi, harga sewa, kategori, jenis, dan pilihan ukuran yang tersedia. Untuk memudahkan proses pemesanan, disediakan tombol “Sewa Sekarang” yang akan mengarahkan pengguna ke halaman reservasi. Selain itu, gambar baju ditampilkan dalam ukuran besar untuk memberikan gambaran visual secara menyeluruh. Halaman ini dirancang agar pengguna dapat mempertimbangkan pilihan dengan lebih matang sebelum melakukan transaksi. Gambar 4.22 menunjukkan tampilan halaman detail baju.



Gambar 4.22 Halaman Detail Baju

Cuplikan kode program:

```
# views.py
def detail_baju(request, id_baju):
    baju = get_object_or_404(Baju, id_baju=id_baju)
    baju_terkait = Baju.objects.exclude(id_baju=id_baju)[:4]
    return render(request, 'baju/detail_baju_user.html', {
        'baju': baju,
        'baju_terkait': baju_terkait
    })
```

Kode di atas juga menampilkan beberapa baju lain sebagai rekomendasi untuk pengguna.

4.2.23 Implementasi Halaman Form Pemesanan dan Cek Ketersediaan

Jika pengguna memilih untuk menyewa baju, mereka diarahkan ke halaman form pemesanan. Pengguna akan diminta memilih tanggal mulai dan selesai, ukuran baju, serta metode pengambilan (*pickup*). Form ini secara otomatis menghitung durasi dan menampilkan biaya sewa. Gambar 4.23 menunjukkan tampilan form pemesanan baju.



Gambar 4.23 Halaman Form Pemesanan

Cuplikan kode program:

```
# views.py
def booking_view(request, id_baju):
    if not request.session.get('id_user'):
        messages.warning(request, 'Silakan login terlebih dahulu.')
        return redirect('detail_baju', id_baju=id_baju)

    baju = get_object_or_404(Baju, id_baju=id_baju)
```

```

ukuran_tersedia = []
if baju.s > 0: ukuran_tersedia.append('S')
if baju.m > 0: ukuran_tersedia.append('M')
if baju.l > 0: ukuran_tersedia.append('L')
if baju.xl > 0: ukuran_tersedia.append('XL')

if request.method == 'POST':
    form = BookingForm(request.POST,
available_sizes=ukuran_tersedia)
    if form.is_valid():
        tgl_mulai = form.cleaned_data['tgl_mulai']
        tgl_selesai = form.cleaned_data['tgl_selesai']
        durasi = max(1, (tgl_selesai - tgl_mulai).days)
        total_harga = baju.harga * durasi

        request.session['booking_data'] = {
            'id_baju': baju.id_baju,
            'ukuran': form.cleaned_data['ukuran'],
            'tgl_mulai': str(tgl_mulai),
            'tgl_selesai': str(tgl_selesai),
            'durasi': durasi,
            'pickup': form.cleaned_data['pickup'],
            'total_harga': total_harga
        }
        return redirect('booking_ready')
    else:
        form = BookingForm(available_sizes=ukuran_tersedia)

return render(request, 'baju/booking.html', {
    'form': form,
    'baju': baju,
    'semua_habis': not ukuran_tersedia
})

```

4.2.24 Implementasi Halaman Ketersediaan Baju

Setelah pengguna mengisi formulir pemesanan, sistem secara otomatis akan memproses data dan menampilkan halaman konfirmasi yang menunjukkan status ketersediaan baju. Jika pakaian tersedia sesuai tanggal dan ukuran yang dipilih, maka pengguna diberikan opsi untuk melanjutkan proses pemesanan dengan mengklik tombol “Sewa”. Selain status stok, halaman ini juga menampilkan rincian biaya total sewa berdasarkan durasi penggunaan, sehingga memberikan transparansi sebelum transaksi dikonfirmasi. Tampilan halaman ini membantu memastikan bahwa tidak terjadi tabrakan pemesanan dan memperkuat pengalaman pengguna. Halaman ini dapat dilihat pada Gambar 4.24.



Gambar 4.24 Halaman Konfirmasi Ketersediaan

Cuplikan kode program:

```
# views.py
def booking_ready(request):
    data = request.session.get('booking_data')
    if not data:
        messages.error(request, 'Data booking tidak ditemukan.')
        return redirect('beranda')

    baju = get_object_or_404(Baju, id_baju=data['id_baju'])

    return render(request, 'baju/booking_ready.html', {
        'baju': baju,
        'tgl_mulai': data['tgl_mulai'],
        'tgl_selesai': data['tgl_selesai'],
        'ukuran': data['ukuran'],
        'pickup': data['pickup'],
        'durasi': data['durasi'],
        'total_harga': data['total_harga']
    })
```

Halaman ini memberikan transparansi kepada pengguna tentang detail pemesanan sebelum proses dikonfirmasi lebih lanjut.

4.2.25 Implementasi Halaman Detail Sewa

Pengguna dapat melihat rincian transaksi yang telah mereka lakukan dalam halaman detail sewa. Informasi yang ditampilkan mencakup kode sewa, nama baju, tanggal sewa, durasi, dan total biaya. Tampilan ini dirancang sederhana dan informatif. Gambar 4.25 menunjukkan tampilan halaman detail sewa.



Gambar 4.25 Halaman Detail Sewa

Cuplikan kode program:

```
# views.py
def booking_detail(request, kode_booking):
    booking = get_object_or_404(Booking,
                                kode_booking=kode_booking)
    total_harga = booking.durasi * booking.id_baju.harga
    info = KelolaInformasi.objects.first()
    rekening = info.rekening if info else 'Belum ada data
    rekening'

    return render(request, 'baju/booking_detail.html', {
        'booking': booking,
        'total_harga': total_harga,
        'rekening': rekening,
    })
```

Data *booking* diambil berdasarkan kode unik dan dihitung total biayanya secara langsung berdasarkan durasi sewa.

4.2.26 Implementasi Halaman Riwayat Sewa

Semua transaksi yang dilakukan oleh pengguna akan tercatat dalam halaman riwayat sewa. Tabel yang ditampilkan berisi informasi kode sewa, nama baju, tanggal, durasi, status, serta opsi untuk mengunggah bukti pembayaran atau melihat detail. Halaman ini sangat membantu pengguna dalam melacak aktivitas penyewaannya. Gambar 4.26 menampilkan halaman riwayat sewa pengguna.

KODE NO SEWA	BAJU	TGL. MULAI	TGL. SELESAI	DURASI	BIAYA	STATUS	OPSI
1 AD8RPPWD	Baju Asat Perempaan Bugis (Beking) Baja	June 16, 2025	June 17, 2025	1 hari	Rp. 140.000	Dibatalan	Detail
2 K068RWSL	Baju Pengantin Bugis Burgundy	June 20, 2025	June 20, 2025	1 hari	Rp. 1.000.000	Dibatalan	Detail
3 L250TQVK	Baju Pengantin Bugis Burgundy	June 16, 2025	June 17, 2025	1 hari	Rp. 1.000.000	Menunggu Pembayaran	Detail Choose File No file chosen Upload Bukti

Gambar 4.26 Halaman Riwayat Sewa

Cuplikan kode program:

```
# views.py
def riwayat_sewa(request):
    id_user = request.session.get('id_user')
    if not id_user:
        messages.error(request, "Anda harus login.")
        return redirect('login_member')

    bookings = Booking.objects.filter(id_user=id_user).order_by('-
    tgl_booking')
    for b in bookings:
        b.total_harga = b.id_baju.harga * b.durasi

    return render(request, 'baju/riwayat_sewa.html', {'bookings':
    bookings})
```

Fungsi ini menyusun daftar riwayat berdasarkan user yang sedang login dan menghitung total harga untuk tiap transaksi.

4.2.27 Implementasi Halaman Profil Pelanggan

Halaman ini memungkinkan pengguna untuk melihat dan memperbarui informasi profil mereka secara mandiri. Informasi yang dapat diedit meliputi nama lengkap, alamat email, nomor telepon, dan alamat tempat tinggal. Seluruh data akan ditampilkan secara otomatis sesuai dengan data yang telah dimasukkan saat proses registrasi, sehingga memudahkan pengguna dalam melakukan pembaruan tanpa

harus menginput ulang seluruh informasi dari awal. Fitur ini berfungsi untuk menjaga keakuratan data pribadi pengguna, terutama ketika terjadi perubahan kontak atau alamat. Dengan kemudahan ini, sistem memastikan bahwa informasi yang tersimpan tetap relevan dan terkini. Gambar 4.27 memperlihatkan tampilan halaman profil pengguna.



Gambar 4.27 Halaman Profil Pengguna

Cuplikan kode program:

```
# forms.py
class ProfileForm(forms.ModelForm):
    class Meta:
        model = User
        fields = ['nama_user', 'email', 'telp', 'alamat']
        widgets = {
            'email': forms.EmailInput(attrs={'readonly': True}),
            ...
        }
# views.py
def profile_settings(request):
    if not request.session.get('id_user'):
        messages.error(request, "Anda harus login terlebih dahulu.")
        return redirect('login_member')

    user = get_object_or_404(Member,
id_user=request.session['id_user'])

    if request.method == 'POST':
        form = ProfileForm(request.POST, instance=user)
        if form.is_valid():
            form.save()
```

```

        messages.success(request, "Profil berhasil
diperbarui.")
        return redirect('profile_settings')
    else:
        form = ProfileForm(instance=user)

        return render(request, 'baju/profile_settings.html', {'form':
form})

```

Pengguna dapat mengelola data pribadi dengan cepat tanpa harus mengisi ulang seluruh informasi.

4.2.28 Implementasi Halaman Ubah *Password*

Untuk menjaga keamanan akun, sistem menyediakan halaman khusus untuk mengganti kata sandi. Pengguna diminta memasukkan kata sandi lama untuk verifikasi, lalu mengisi dan mengonfirmasi kata sandi baru. Fitur *show/hide password* disertakan untuk meminimalkan kesalahan input. Halaman ini menjadi bagian penting dalam mencegah akses yang tidak sah. Gambar 4.28 menunjukkan tampilan halaman ubah *password*.



Gambar 4.28 Halaman Ubah *Password*

Cuplikan kode program:

```

# views.py
def update_password(request):
    if not request.session.get('id_user'):
        messages.error(request, "Silakan login terlebih dahulu.")
        return redirect('login_member')

    user = get_object_or_404(Member,
id_user=request.session['id_user'])

```

```

if request.method == 'POST':
    current = request.POST.get('password')
    new = request.POST.get('newpassword')
    confirm = request.POST.get('confirmpassword')

    if not check_password(current, user.password):
        messages.error(request, "Password lama salah.")
    elif new != confirm:
        messages.error(request, "Password baru dan konfirmasi
tidak cocok.")
    else:
        user.password = make_password(new)
        user.save()
        messages.success(request, "Password berhasil
diperbarui.")
        return redirect('update_password')

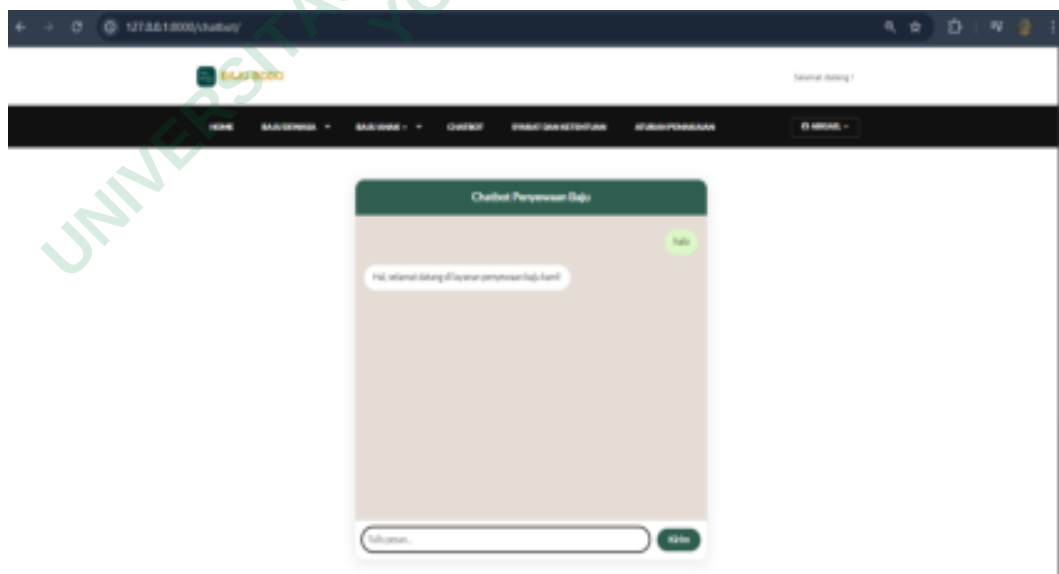
return render(request, 'baju/update_password.html')

```

Keamanan pengguna ditingkatkan dengan validasi *password* lama dan penggunaan hash saat menyimpan *password* baru.

4.2.29 Implementasi Halaman Chatbot

Sistem menyediakan chatbot berbasis NLP untuk menjawab pertanyaan dasar seputar penyewaan, prosedur pemesanan, dan layanan. Namun, karena belum terhubung ke basis data, chatbot belum dapat menjawab pertanyaan spesifik atau real-time. Gambar 4.29 menampilkan antarmukanya.



Gambar 4.29 Halaman Chatbot

Cuplikan kode chatbot:

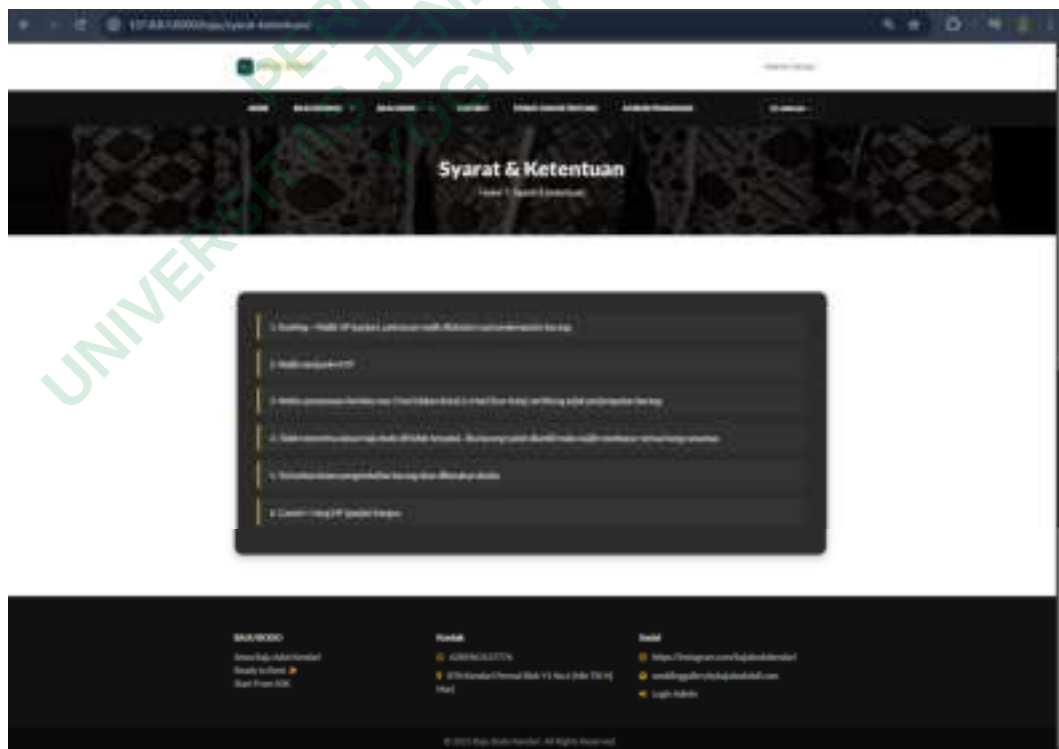
```
# bot.py
def get_response(ints, intents_json):
    tag = ints[0]['intent']
    for i in intents_json['intents']:
        if i['tag'] == tag:
            return random.choice(i['responses'])

# views.py
def chatbot(request):
    return render(request, 'chatbot/chatbot.html')
```

Chatbot dipicu melalui antarmuka HTML yang mengirimkan input ke model klasifikasi intent, lalu memberikan respons yang sesuai dari `intents.json`.

4.2.30 Implementasi Halaman Syarat dan Ketentuan

Halaman ini menampilkan daftar syarat dan ketentuan yang berlaku bagi pengguna layanan penyewaan. Informasi seperti kewajiban DP, jaminan KTP, batas waktu peminjaman, dan sanksi keterlambatan dijelaskan secara terstruktur. Ini penting untuk mengatur hubungan antara pengguna dan penyedia layanan. Gambar 4.30 memperlihatkan tampilannya.



Gambar 4.30 Halaman Syarat dan Ketentuan

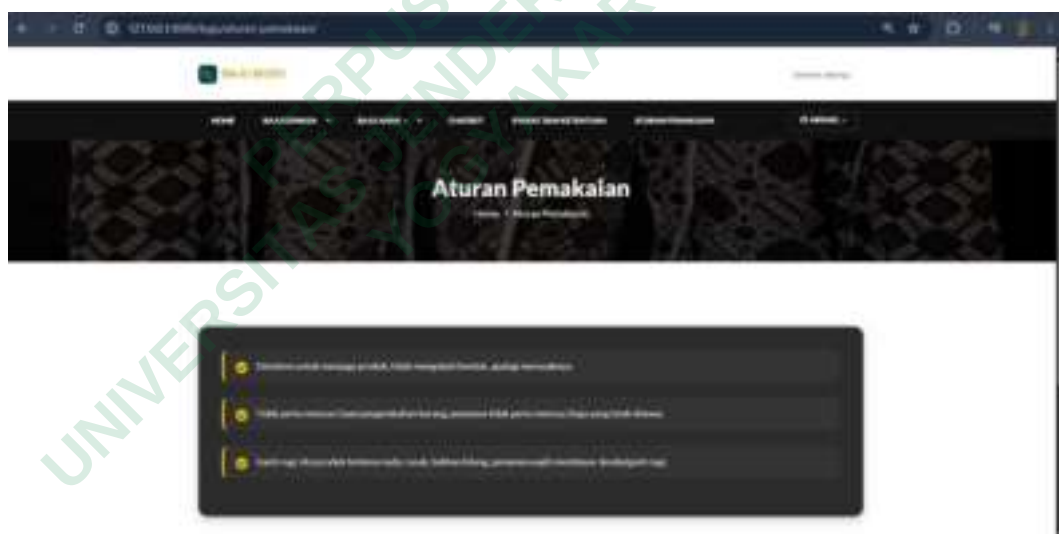
Cuplikan kode program:

```
# views.py
def syarat_ketentuan(request):
    info = KelolaInformasi.objects.first()
    syarat_list = info.syarat_ketentuan.strip().splitlines() if
info and info.syarat_ketentuan else []
    return render(request, 'baju/syarat_ketentuan.html',
{'syarat_list': syarat_list})
```

Data diambil dari field `syarat_ketentuan` pada model `KelolaInformasi` dan ditampilkan sebagai daftar poin ke pengguna.

4.2.31 Implementasi Halaman Aturan Pemakaian

Halaman ini menjelaskan etika dan tanggung jawab pengguna selama menggunakan baju adat, termasuk larangan merusak atau mencuci, serta kewajiban mengganti rugi jika terdapat kerusakan. Aturan ini membantu menjaga kelayakan baju untuk disewa oleh pengguna lain. Gambar 4.31 menampilkan halaman aturan pemakaian.



Gambar 4.31 Halaman Aturan Pemakaian

Cuplikan kode program:

```
# views.py
def aturan_pemakaian(request):
    info = KelolaInformasi.objects.first()
    aturan_list = info.aturan_pemakaian.strip().splitlines() if
info and info.aturan_pemakaian else []
```

```
return render(request, 'baju/aturan_pemakaian.html',
{'aturan_list': aturan_list})
```

Konten ini dikelola oleh admin dari panel kelola informasi dan ditampilkan secara dinamis ke pengguna.

4.3 BASIS DATA

Basis data yang digunakan dalam sistem informasi penyewaan pakaian adat ini diberi nama db_sewabaju dan terdiri atas 18 tabel, di antaranya terdapat tabel inti yang mendukung aktivitas transaksi, manajemen data pakaian, serta administrasi informasi pengguna dan penyedia layanan. Setiap tabel memiliki atribut *id* sebagai *primary key* dan beberapa tabel dihubungkan menggunakan foreign key untuk menjamin integritas relasional antartentitas. Implementasi struktur database secara umum ditunjukkan pada Gambar 4.32.

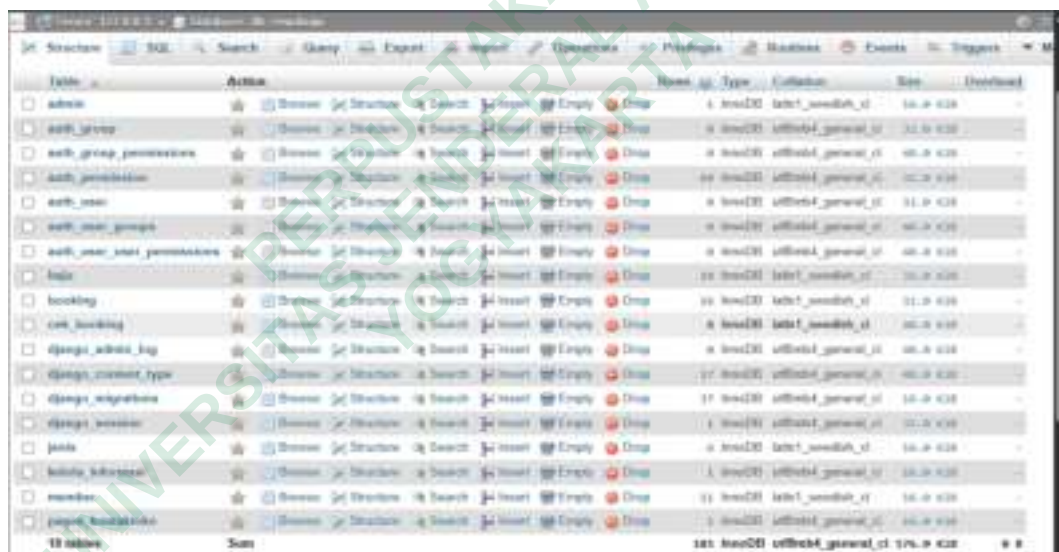


Table	Action	PK	FK	Index	Engine	Collation	Size	Download
admin	Structure	id			MyISAM	latin1_swedish_ci	16.0 K	
auth_group	Structure	id			MyISAM	latin1_swedish_ci	32.0 K	
auth_group_permissions	Structure	id	auth_group		MyISAM	latin1_swedish_ci	40.0 K	
auth_permission	Structure	id			MyISAM	latin1_swedish_ci	32.0 K	
auth_user	Structure	id			MyISAM	latin1_swedish_ci	31.0 K	
auth_user_groups	Structure	id	auth_user		MyISAM	latin1_swedish_ci	40.0 K	
auth_user_permissions	Structure	id	auth_user		MyISAM	latin1_swedish_ci	40.0 K	
baju	Structure	id			MyISAM	latin1_swedish_ci	30.0 K	
booking	Structure	id			MyISAM	latin1_swedish_ci	31.0 K	
cat_booking	Structure	id			MyISAM	latin1_swedish_ci	30.0 K	
djangan_adat_bag	Structure	id			MyISAM	latin1_swedish_ci	40.0 K	
djangan_ceremoni_type	Structure	id			MyISAM	latin1_swedish_ci	40.0 K	
djangan_migrasi	Structure	id			MyISAM	latin1_swedish_ci	30.0 K	
djangan_wisata	Structure	id			MyISAM	latin1_swedish_ci	30.0 K	
jenis	Structure	id			MyISAM	latin1_swedish_ci	16.0 K	
kalenda_kalender	Structure	id			MyISAM	latin1_swedish_ci	16.0 K	
manajemen	Structure	id			MyISAM	latin1_swedish_ci	16.0 K	
manajemen_kalender	Structure	id			MyISAM	latin1_swedish_ci	16.0 K	
18 tables	Sum						180.0 K	

Gambar 4.32 Implementasi Struktur Basis Data

4.3.1 Tabel Admin

Tabel admin digunakan untuk menyimpan informasi akun administrator yang memiliki hak akses penuh terhadap sistem. Atribut yang terdapat dalam tabel ini meliputi *name*, *username*, *password*, *updateDate*, dan *image*. Struktur tabel admin ditunjukkan pada Gambar 4.33.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
3	UserName	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
4	Password	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
5	updateDate	datetime			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change Drop More
6	image	varchar(255)	latin1_swedish_ci		No	NULL			Change Drop More

Gambar 4.33 Implementasi Tabel Admin

4.3.2 Tabel Baju

Tabel baju menyimpan data lengkap mengenai pakaian adat yang tersedia untuk disewa. Atribut yang dikelola mencakup nama baju, jenis, kategori, deskripsi, harga, serta jumlah stok berdasarkan ukuran (S, M, L, XL). Tabel ini juga memuat gambar pakaian dalam tiga kolom. Struktur tabel ini ditunjukkan pada Gambar 4.34.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_baju	int(5)			No	None		AUTO_INCREMENT	Change Drop More
2	nama_baju	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
3	id_jenis	int(5)			No	None			Change Drop More
4	kategori	varchar(10)	latin1_swedish_ci		No	None			Change Drop More
5	deskripsi	text	latin1_swedish_ci		No	None			Change Drop More
6	harga	int(3)			No	None			Change Drop More
7	s	int(5)			No	None			Change Drop More
8	m	int(5)			No	None			Change Drop More
9	l	int(5)			No	None			Change Drop More
10	xl	int(5)			No	None			Change Drop More
11	gambar1	varchar(255)	latin1_swedish_ci		Yes	NULL			Change Drop More
12	gambar2	varchar(255)	latin1_swedish_ci		Yes	NULL			Change Drop More
13	gambar3	varchar(255)	latin1_swedish_ci		Yes	NULL			Change Drop More

Gambar 4.34 Implementasi Tabel Baju

4.3.3 Tabel Booking

Tabel booking berfungsi mencatat transaksi penyewaan yang dilakukan oleh pelanggan. Informasi yang tersimpan antara lain kode booking, id baju, ukuran, tanggal sewa, durasi, email pelanggan, metode pengambilan, status penyewaan, serta bukti pembayaran. Tabel ini juga berelasi dengan id_user dari tabel member. Struktur tabel ditunjukkan pada Gambar 4.35.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1 kode_booking	varchar(8)	latin1_swedish_ci		No	None			Change Drop More
☐	2 id_baju	int(11)			No	None			Change Drop More
☐	3 ukuran	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
☐	4 tgl_mulai	date			No	None			Change Drop More
☐	5 tgl_akhir	date			No	None			Change Drop More
☐	6 durasi	int(11)			No	None			Change Drop More
☐	7 denda	int(11)			No	None			Change Drop More
☐	8 status	varchar(20)	latin1_swedish_ci		No	None			Change Drop More
☐	9 email	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
☐	10 pickup	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
☐	11 tgl_booking	date			No	None			Change Drop More
☐	12 bukti_bayar	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
☐	13 id_user	int(11)			Yes	NULL			Change Drop More

Gambar 4.35 Implementasi Tabel Booking

4.3.4 Tabel Cek Booking

Tabel cek_booking menyimpan data hasil pengecekan ketersediaan baju yang akan disewa, berdasarkan ukuran dan tanggal booking. Struktur tabel ini ditunjukkan pada Gambar 4.36.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1 id_cek	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2 kode_booking	varchar(8)	latin1_swedish_ci		No	None			Change Drop More
☐	3 id_baju	int(11)			No	None			Change Drop More
☐	4 ukuran	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
☐	5 tgl_booking	varchar(10)	latin1_swedish_ci		No	None			Change Drop More
☐	6 status	varchar(20)	latin1_swedish_ci		No	None			Change Drop More

Gambar 4.36 Implementasi Tabel Cek Booking

4.3.5 Tabel Jenis

Tabel jenis berfungsi untuk mengelompokkan pakaian adat berdasarkan jenis atau etnis (misalnya Bugis, Buton, atau Muna). Tabel ini bersifat referensial dan memiliki relasi dengan tabel baju. Implementasi tabel ini ditunjukkan pada Gambar 4.37.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1 id_jenis	int(1)			No	None		AUTO_INCREMENT	Change Drop More
☐	2 nama_jenis	varchar(50)	latin1_swedish_ci		No	None			Change Drop More

Gambar 4.37 Implementasi Tabel Jenis

4.3.6 Tabel Kelola Informasi

Tabel `kelola_informasi` digunakan untuk menyimpan data konten halaman syarat dan ketentuan, aturan pemakaian, dan informasi rekening toko. Data dari tabel ini akan ditampilkan secara dinamis di antarmuka pengguna. Struktur tabel ini ditunjukkan pada Gambar 4.38.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)			No	None		AUTO_INCREMENT	Change Drop More
2	syarat_ketentuan	longtext	utf8mb4_general_ci		No	None			Change Drop More
3	aturan_pemakaian	longtext	utf8mb4_general_ci		No	None			Change Drop More
4	rekening	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More

Gambar 4.38 Implementasi Tabel Kelola Informasi

4.3.7 Tabel Member

Tabel `member` menyimpan data pelanggan yang telah melakukan registrasi akun dalam sistem. Atribut dalam tabel ini mencakup nama, email, password, nomor telepon, dan alamat. Tabel ini berelasi dengan tabel `booking`. Struktur tabel `member` ditunjukkan pada Gambar 4.39.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id_user	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	nama_user	varchar(128)	latin1_swedish_ci		Yes	NULL			Change Drop More
3	email	varchar(100)	latin1_swedish_ci		Yes	NULL			Change Drop More
4	password	varchar(100)	latin1_swedish_ci		Yes	NULL			Change Drop More
5	telep	varchar(20)	latin1_swedish_ci		Yes	NULL			Change Drop More
6	alamat	varchar(255)	latin1_swedish_ci		Yes	NULL			Change Drop More

Gambar 4.39 Implementasi Tabel Member

4.3.8 Tabel Pages_KontakToko

Tabel `pages_kontaktoko` menyimpan informasi kontak toko yang ditampilkan pada halaman utama sistem. Informasi ini meliputi alamat toko, akun Instagram, nomor WhatsApp, dan nama website toko. Struktur tabel ditunjukkan pada Gambar 4.40.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)			No	None		AUTO_INCREMENT	Change Drop More
2	alamat_toko	longtext	utf8mb4_general_ci		No	None			Change Drop More
3	instagram_toko	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	no_whatapp_toko	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
5	nama_website_toko	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

Gambar 4.40 Implementasi Tabel Kontak Toko

4.4 FITUR-FITUR SISTEM

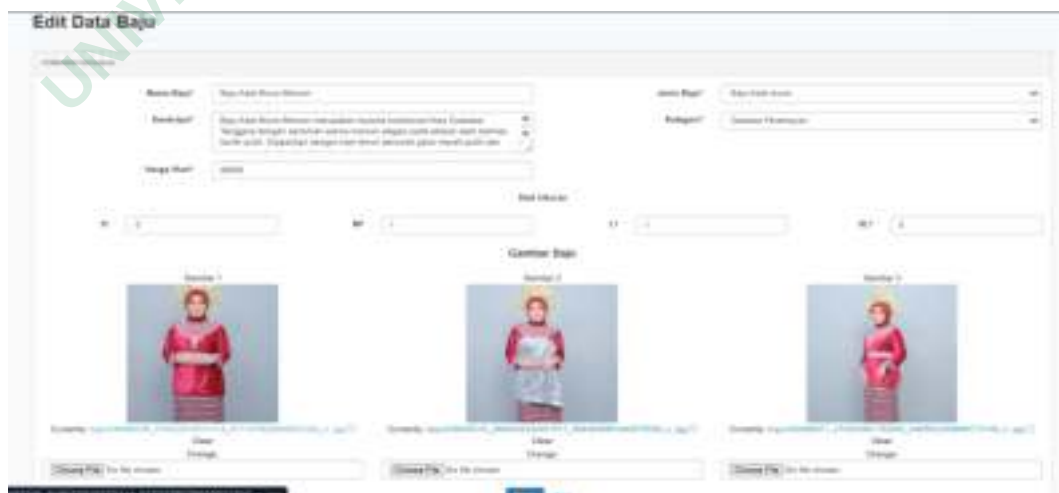
Sistem informasi penyewaan pakaian adat berbasis web yang dikembangkan dalam penelitian ini mencakup berbagai fitur utama yang dirancang untuk mendukung pengelolaan layanan penyewaan secara digital. Setiap fitur dikembangkan sesuai dengan kebutuhan pengguna, baik dari sisi administrator (admin) maupun pelanggan (*member*). Penjelasan fitur-fitur tersebut dijabarkan sebagai berikut:

4.4.1 Fitur Manajemen Data Pakaian

Fitur ini memungkinkan admin untuk menambahkan, mengubah, dan menghapus data baju adat. Informasi yang dikelola mencakup nama baju, jenis, kategori, harga sewa, deskripsi, stok berdasarkan ukuran, dan gambar pendukung. Fitur ini ditunjukkan pada Gambar 4.41 dan Gambar 4.42.



Gambar 4.41 Halaman Detail Data Baju



Gambar 4.42 Halaman Edit Data Baju

4.4.2 Fitur Reservasi dan Validasi Ketersediaan

Pelanggan dapat melakukan reservasi secara daring dengan mengisi formulir pemesanan berdasarkan tanggal, ukuran, dan metode pengambilan. Sistem secara otomatis memverifikasi ketersediaan baju. Tampilan fitur ini ditunjukkan pada Gambar 4.43 dan Gambar 4.44.

Form Pemesanan Baju

Baju Adat Mornene
Rp. 85,000 / Hari

Tanggal Mulai
DD-MM-YYYY

Tanggal Selesai
DD-MM-YYYY

Ukuran
L

Metode Pickup
Ambil Sendiri

Cek Ketersediaan

Gambar 4.43 Formulir Pemesanan dan Cek Ketersediaan

Baju Tersedia untuk disewa.

Baju Adat Mornene
Rp. 85,000 / Hari

Tanggal Mulai
2025-05-20

Tanggal Selesai
2025-05-21

Durasi
1 Hari

Ukuran
L

Metode Pickup
Ambil Sendiri

Biaya Sewa
Rp. 85,000

Sewa

Gambar 4.44 Halaman Konfirmasi Ketersediaan Baju

4.4.3 Fitur Riwayat Penyewaan dan Transaksi

Sistem menyimpan seluruh riwayat penyewaan yang dilakukan pelanggan. Admin dapat melihat detail transaksi serta mencetak laporan berdasarkan periode tertentu. Tampilan fitur ini ditunjukkan pada Gambar 4.45 dan Gambar 4.46.

Riwayat Sewa Anda							
NO	KODE SEWA	BAJU	TGL MULAI	TGL SELESAI	DURASI	BIAYA	STATUS
1	PRVDJDLX	Baju Adat Momen	June 15, 2025	June 21, 2025	1 hari	Rp. 85.000	Menunggu Pembayaran
							Detail Choose File No file chosen Upload Photo
2	ADJ08PMD	Baju Adat Peremajaan Baju Koteng Hajar	June 16, 2025	June 17, 2025	1 hari	Rp. 140.000	Dibatalkan
							Detail
3	IK45BWSL	Baju Pengantin Baju Burgundy	June 28, 2025	June 29, 2025	1 hari	Rp. 1.000.000	Dibatalkan
							Detail 
4	LT25T24X	Baju Pengantin Baju Burgundy	June 16, 2025	June 17, 2025	1 hari	Rp. 1.000.000	Menunggu Pembayaran
							Detail Choose File No file chosen Upload Photo

4.4.4 Fitur Chatbot Berbasis Natural Language Processing (NLP)

Fitur chatbot dapat memberikan jawaban otomatis atas pertanyaan pelanggan terkait layanan penyewaan, menggunakan *pendekatan Natural Language Processing*. Tampilan fitur ini ditunjukkan pada Gambar 4.47.



Formulir profil pelanggan yang mencakup:

- Nama:** abigael
- Alamat Email:** abi@gmail.com
- Telepon:** 082292345988
- Alamat:** dana
- Tombol:** Simpan Perubahan

Gambar 4.49 Halaman Profil Pelanggan



Formulir ubah password yang mencakup:

- Form Ubah Password** (Judul)
- Password Saat Ini:** [input field] [toggle visibility]
- Password Baru:** [input field] [toggle visibility]
- Konfirmasi Password:** [input field] [toggle visibility]
- Tombol:** Simpan Perubahan

Gambar 4.50 Halaman Ubah Password

4.4.6 Fitur Pengelolaan Informasi Umum dan Kontak

Admin dapat mengelola konten informasi umum seperti syarat dan ketentuan, aturan pemakaian, serta kontak toko. Tampilan fitur ini ditunjukkan pada Gambar 4.51 dan Gambar 4.52.

Kelola Informasi Umum

Panduan Pengisian Konten

Syarat & Ketentuan: Masukkan poin-poin dalam bentuk lampiran. Gunakan `<p></p>` untuk tiap baris. Jangan tambahkan tag HTML.

Aturan Penulisan: Sama seperti Syarat & Ketentuan. Setiap aturan akan tampil sebagai poin dengan ikon di halaman pengguna.

Informasi Rekening: Hanya satu baris, cukup jelas & singkat. Misal: BRI - a/n Saver Baju Bodo - 0234 5678 9012

FORMA KEMAS INFORMASI

Syarat dan Ketentuan	Booking = flight (DP pengal), pemesanan wajib dilakukan saat pengumpulan barang flight minimum 14 hari Waktu pengiriman berlaku max 3 hari (dalam kota) & 4 hari (luar kota), terhitung sejak pengumpulan barang.
Aturan Penulisan	Deskripsi untuk menjaga produk, tidak mengubah bentuk, apatag transaksinya Total perlu mencuri 500 pengembalian barang, penerima tidak perlu mengisi baju yang telah diterima. Ganti rugi jika produk terlihat rusak, rusak.
Informasi Rekening	BRI - a/n Saver Baju Bodo - 0234 5678 9012

[Simpan Perubahan](#)

Gambar 4.51 Halaman Kelola Informasi Umum

Kelola Kontak Toko

FORMA KEMAS TOKO

Panduan Pengisian:

Alamat Toko: Masukkan alamat lengkap toko misal: BTN Kencana Permai Blok Y1 No.6 (Me 18 Hg Mar)

Instagram Toko: Masukkan URL profil Instagram lengkap: <https://www.instagram.com/bajubodo>

No WhatsApp Toko: Masukkan nomor aktif di level kode negara (kode antar +), contoh: 6281234567890

Nama Website Toko: Masukkan nama domain toko, misal: www.bajubodotoko.com

Alamat Toko:

Instagram Toko:

No WhatsApp:

Nama Website:

[Simpan Perubahan](#)

Gambar 4.52 Halaman Kelola Kontak Toko

4.4.7 Fitur Dashboard Statistik Admin

Dashboard menyediakan statistik *real-time* seperti jumlah transaksi aktif, total baju, dan pengguna. Tampilan fitur ini ditunjukkan pada Gambar 4.53.



Gambar 4.53 Halaman Dashboard Statistik Admin

4.4.8 Fitur Manajemen Jenis Baju

Admin dapat menambah, memperbarui, dan menghapus jenis baju yang tersedia. Tampilan fitur ini ditunjukkan pada Gambar 4.54, Gambar 4.55, dan Gambar 4.56.



Gambar 4.54 Halaman Kelola Jenis Baju

The screenshot shows the 'Tambah Jenis' form with a single input field for 'Nama Jenis' and two buttons: 'Tambah' and 'Batal'.

Gambar 4.55 Halaman Tambah Jenis Baju



Gambar 4.56 Halaman Edit Jenis Baju

4.4.9 Fitur Pencetakan Laporan Otomatis

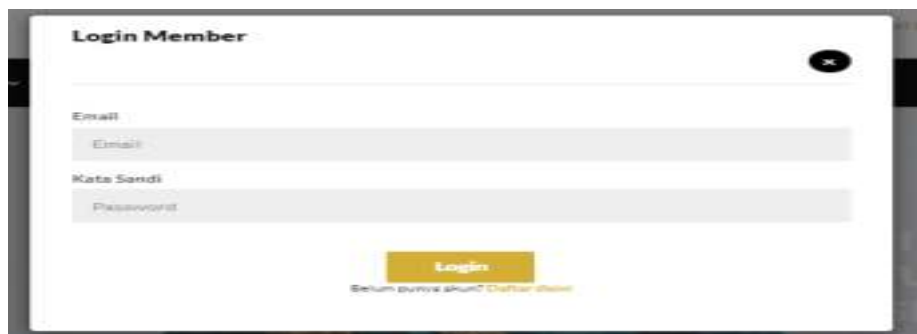
Sistem mendukung pencetakan laporan transaksi penyewaan dalam format PDF, berdasarkan tanggal yang dipilih oleh admin. Tampilan fitur ini ditunjukkan pada Gambar 4.57.



Gambar 4.57 Halaman Pencetakan Laporan Penyewaan

4.4.10 Fitur Autentikasi dan Registrasi Pelanggan

Fitur ini memungkinkan pelanggan untuk membuat akun dan masuk ke sistem menggunakan kredensial yang sah. Tampilan fitur ini ditunjukkan pada Gambar 4.58 dan Gambar 4.59.

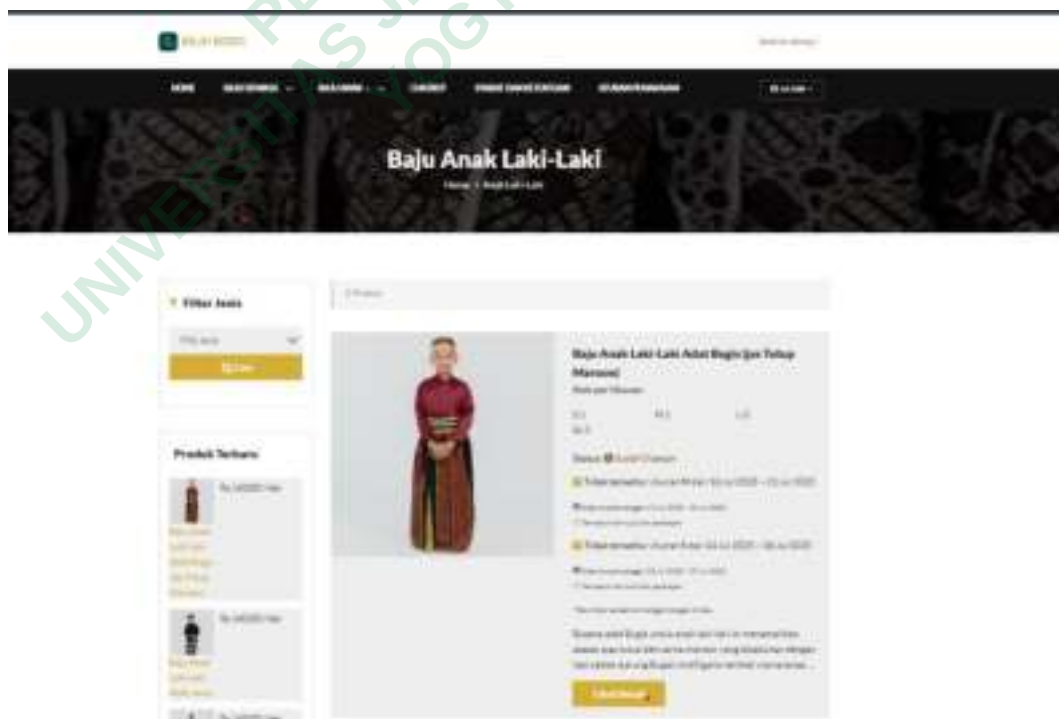


Gambar 4.58 Halaman Login Pelanggan

Gambar 4.59 Halaman Registrasi Pelanggan

4.4.11 Fitur Katalog dan Detail Baju

Pelanggan dapat menelusuri seluruh baju adat yang tersedia melalui katalog dan melihat informasi rinci sebelum menyewa. Fitur ini ditunjukkan pada Gambar 4.60 dan Gambar 4.61.



Gambar 4.60 Halaman Katalog Baju

Baju Adat Mornene

Rp 85,000
/Hari

Deskripsi Baju

Baju Adat Mornene merupakan busana tradisional khas suku Moronene di Sulawesi Tenggara. Terdiri dari atasan beludru hitam berhias ornamen emas dan numbai perak di bagian lengan serta bawah baju. Dipadukan dengan sarung kotak merah-hitam dan aksesoris kepala khas, baju ini mencerminkan identitas budaya yang kuat, anggun, dan bernilai. Cocok untuk upacara adat dan pertunjukan budaya.

 Sewa Sekarang

 Sewa Sekarang

Gambar 4.61 Halaman Detail Baju

4.4.12 Fitur Informasi Syarat dan Ketentuan serta Aturan Pemakaian

Fitur ini menampilkan informasi hukum dan ketentuan teknis terkait layanan penyewaan kepada pelanggan secara terbuka. Tampilan fitur ini ditunjukkan pada Gambar 4.62 dan Gambar 4.63.



Gambar 4.62 Halaman Syarat dan Ketentuan



Gambar 4.63 Halaman Aturan Pemakaian

4.5 PENGUJIAN SISTEM

Tahapan pengujian dalam pengembangan perangkat lunak memiliki peran esensial untuk memastikan bahwa sistem telah dibangun sesuai dengan kebutuhan fungsional dan preferensi pengguna. Proses ini dilakukan untuk menilai apakah seluruh fitur sistem bekerja sebagaimana mestinya berdasarkan standar yang telah dirancang sebelumnya. Dalam penelitian ini, pengujian sistem informasi penyewaan pakaian adat berbasis web menggunakan dua pendekatan utama, yaitu pengujian *Black Box* dan *User Acceptance Testing* (UAT). Kedua pendekatan tersebut tidak hanya bertujuan untuk memvalidasi fungsi sistem, tetapi juga untuk mengukur seberapa jauh sistem dapat diterima oleh pengguna akhir dari sisi kegunaan dan kenyamanan (Pressman & Maxim, 2014).

4.5.1 Pengujian *Black Box*

Metode *Black Box Testing* merupakan salah satu strategi evaluasi perangkat lunak yang berfokus pada pengujian output sistem berdasarkan berbagai macam input yang diberikan, tanpa melibatkan pengamatan langsung terhadap struktur internal program. Dengan kata lain, penguji tidak memerlukan akses atau pengetahuan terhadap kode sumber, melainkan cukup menilai apakah sistem memberikan respons yang tepat sesuai dengan spesifikasi yang ditetapkan. Pendekatan ini sangat berguna dalam konteks pengguna akhir karena menguji sistem sebagaimana mereka menggunakannya dalam praktik sehari-hari (Priyaungga et al., 2020).

Dalam penelitian ini, teknik *State Transition* digunakan sebagai dasar untuk melakukan pengujian *black box*. Teknik ini menilai bagaimana sistem berpindah dari satu kondisi ke kondisi lainnya berdasarkan aksi yang dilakukan, baik oleh pengguna maupun administrator. Teknik ini sangat sesuai untuk sistem penyewaan yang memiliki alur proses berlapis, seperti dimulai dari registrasi akun, proses pemesanan baju, pengunggahan bukti pembayaran, konfirmasi oleh admin, hingga penyelesaian transaksi. Evaluasi ini memungkinkan peneliti untuk memverifikasi bahwa perubahan status yang terjadi di dalam sistem berlangsung secara konsisten dan sesuai dengan desain yang telah dirancang. Hasil pengujian dengan pendekatan

state transition dirangkum dalam Tabel 4.1, yang menunjukkan berbagai skenario perpindahan status dalam sistem beserta hasil evaluasinya.

Tabel 4.1 Pengujian Sistem *Black Box Testing*

No	Pengguna	Fitur yang Diuji	Test Case / Skenario Uji	Hasil Diharapkan	Hasil Uji	Kesimpulan
1	Member	Login Member	Masukkan email dan <i>password</i> valid	Berhasil masuk ke dashboard pengguna	Sesuai	Valid
2	Member	Registrasi Akun	Mengisi semua kolom (nama, email, telp, alamat, <i>password</i>) dengan data valid	Akun berhasil dibuat dan diarahkan ke login	Sesuai	Valid
3	Member	Lihat Katalog Baju	Klik menu "Baju Dewasa" atau "Baju Anak"	Katalog tampil sesuai kategori	Sesuai	Valid
4	Member	Detail Baju	Klik salah satu baju dari katalog	Muncul halaman detail lengkap baju	Sesuai	Valid
5	Member	Form Pemesanan Baju	Isi tanggal, ukuran, metode pickup dan	Info ketersediaan dan harga tampil	Sesuai	Valid

No	Pengguna	Fitur yang Diuji	Test Case / Skenario Uji	Hasil Diharapkan	Hasil Uji	Kesimpulan
			klik “Cek Ketersediaan”			
6	Member	Konfirmasi Penyewaan	Klik “Sewa Sekarang”	Status sewa menjadi “Belum Dibayar”	Sesuai	Valid
7	Member	Upload Bukti Pembayaran	Klik tombol “Upload Bukti” dan unggah file gambar	Bukti tampil dan status berubah menjadi “Menunggu Konfirmasi”	Sesuai	Valid
8	Member	Lihat Riwayat Sewa	Klik menu “Riwayat Sewa”	Tabel penyewaan tampil sesuai histori	Sesuai	Valid
9	Member	Profil Pengguna	Klik menu “Profil” dan lihat data	Data diri tampil sesuai informasi akun	Sesuai	Valid
10	Member	Ubah Password	Isi password lama, baru, dan	Password berhasil diubah	Sesuai	Valid

No	Pengguna	Fitur yang Diuji	Test Case / Skenario Uji	Hasil Diharapkan	Hasil Uji	Kesimpulan
			konfirmasi lalu simpan			
11	Member	Chatbot	Ketik pertanyaan pada chatbot dan klik “Kirim”	Jawaban tampil otomatis di layar	Sesuai	Valid
12	Admin	Login Admin	Masukkan <i>username</i> dan <i>password</i> valid	Masuk ke dashboard admin	Sesuai	Valid
13	Admin	Dashboard Admin	Lihat statistik sewa, <i>member</i> , baju, dan status	Statistik tampil sesuai database	Sesuai	Valid
14	Admin	Konfirmasi Pembayaran	Lihat penyewaan status “Menunggu Konfirmasi” → klik konfirmasi	Status berubah menjadi “Sedang Disewa”	Sesuai	Valid
15	Admin	Tandai Sewa Selesai	Klik tombol “Selesai” pada	Status berubah menjadi “Selesai”	Sesuai	Valid

No	Pengguna	Fitur yang Diuji	Test Case / Skenario Uji	Hasil Diharapkan	Hasil Uji	Kesimpulan
			transaksi aktif			
16	Admin	Kelola Jenis Baju	Tambah / edit / hapus jenis baju	Data berhasil diperbarui	Sesuai	Valid
17	Admin	Kelola Data Baju	Tambah / edit / lihat detail baju	Baju tampil sesuai input	Sesuai	Valid
18	Admin	Kelola Data <i>Member</i>	Lihat tabel <i>member</i> dan informasi profil	Daftar <i>member</i> tampil dengan benar	Sesuai	Valid
19	Admin	Kelola Konten Informasi & Syarat	Ubah info kontak, syarat, aturan pemakaian	Data tersimpan dan tampil di halaman depan	Sesuai	Valid
20	Admin	Laporan Penyewaan	Pilih rentang tanggal dan cetak laporan	Laporan tampil sesuai <i>filter</i>	Sesuai	Valid

Berdasarkan hasil pengujian, seluruh fungsi sistem telah berjalan sesuai dengan harapan dan tidak ditemukan kesalahan fungsional pada tahap ini. Hal ini menunjukkan bahwa sistem telah memenuhi kebutuhan fungsional sebagaimana dirancang dalam spesifikasi awal.

4.5.2 Pengujian *User Acceptance Testing*

Pengujian *User Acceptance Testing* (UAT) merupakan tahap akhir dalam siklus pengembangan perangkat lunak yang dilakukan oleh pengguna akhir (*end-user*) untuk mengevaluasi sejauh mana sistem telah memenuhi kebutuhan serta spesifikasi fungsional yang ditetapkan. Tahapan ini sangat penting untuk memberikan validasi bahwa sistem benar-benar siap digunakan dalam lingkungan operasional dan dapat diterima oleh penggunanya.

Menurut Sambas & Ripai (2022), pengujian *UAT* dilaksanakan menggunakan metode angket skala *Likert* dengan 5 tingkat penilaian, yang masing-masing memiliki bobot tertentu. Bobot ini kemudian dikalikan dengan jumlah frekuensi responden untuk menghasilkan skor total. Selanjutnya skor ini dibandingkan dengan skor ideal, yang dihitung dari jumlah responden dikalikan jumlah item pertanyaan dan dikalikan skor maksimal (5).

Tabel 4.2 Skala *Likert UAT*

Skor Persentase	Nilai	Kualifikasi
80% – 100%	5	Sangat Setuju
60% – 79.99%	4	Setuju
40% – 59.99%	3	Kurang Setuju
20% – 39.99%	2	Tidak Setuju
0% – 19.99%	1	Sangat Tidak Setuju

Data dianalisis menggunakan rumus persentase sebagai berikut:

$$NP = \frac{f}{n} \times 100\%$$

Keterangan:

P = Persentase penerimaan sistem

f = Jumlah skor total dari semua responden

n = Skor ideal (jumlah responden × jumlah butir × skor maksimum)

a. Kegunaan (*Usefulness*)

Tabel 4.3 Item Pernyataan *Usefulness* untuk *Member*

No	Pernyataan
----	------------

Tabel 4.3 Item Pernyataan *Usefulness* untuk *Member*

No	Pernyataan
1	Saya dapat dengan mudah mencari baju berdasarkan kategori.
2	Fitur <i>filter</i> dan pencarian sangat membantu saat mencari baju.
3	Sistem menampilkan detail baju yang lengkap.
4	Saya bisa memilih ukuran dan tanggal sewa sesuai keinginan.
5	Sistem <i>memberitahu</i> jika baju tidak tersedia.
6	Saya dapat mencetak bukti sewa dengan mudah.
7	Proses upload bukti pembayaran berjalan lancar.
8	Riwayat penyewaan tercatat dengan baik.
9	Saya bisa mengetahui status sewa saya.
10	Informasi seperti aturan pemakaian mudah diakses.

Tabel 4.4 Hasil Perhitungan *Usefulness Member*

No	Penguji	item kuisioner <i>usefulness</i>
----	---------	----------------------------------

Tabel 4.4 Hasil Perhitungan *Usefulness Member*

[illegible]

No	Penguji	item kuisioner <i>usefulness</i>									
		1	2	3	4	5	6	7	8	9	10
10	Member 10	4	4	5	4	4	4	4	4	5	4
Total		42	47	46	43	46	47	44	45	45	45

Skor Ideal = $10 \times 10 \times 5 = 500$

$$NP = \frac{455}{500} \times 100\% = 91\%$$

Dari perhitungan diatas menunjukkan bahwa pelanggan merasa sistem sangat berguna, dengan nilai sebesar 91 %, sehingga dikategorikan sebagai Sangat Setuju.

b. Kepuasan (*Satisfaction*)

Berikut Tabel 4.5 berisi pernyataan yang mengevaluasi kepuasan pengguna terhadap sistem.

Tabel 4.5 Item Pernyataan *Satisfaction* untuk Member

No	Pernyataan
1	Saya puas dengan pengalaman menyewa melalui sistem ini.
2	Desain dan navigasi situs nyaman diakses.
3	Proses pemesanan terasa cepat dan tidak ribet.

Tabel 4.6 berikut merangkum skor dari masing-masing responden terhadap aspek ini:

Tabel 4.6 Hasil Perhitungan *Satisfaction* Member

No	Penguji	item kuisioner <i>Satisfaction</i>		
		1	2	3
1	Member 1	4	4	4
2	Member 2	5	5	5
3	Member 3	4	5	5
4	Member 4	4	3	4
5	Member 5	4	5	5
6	Member 6	4	4	3
7	Member 7	4	4	4
8	Member 8	5	4	5

No	Penguji	item kuisisioner <i>Satisfaction</i>		
		1	2	3
9	Member 9	5	5	5
10	Member 10	4	4	4
Total		43	43	44

Skor Ideal = $10 \times 3 \times 5 = 150$

$$NP = \frac{130}{150} \times 100\% = 86.7\%$$

Perhitungan diatas menunjukkan bahwa pelanggan merasa puas terhadap sistem dengan nilai 86.7%, termasuk dalam kategori Sangat Setuju.

c. Kemudahan Penggunaan (*Ease of Use*)

Tabel 4.7 memuat item pernyataan terkait kemudahan penggunaan fitur sistem.

Tabel 4.7 Item Pernyataan *Ease of Use* untuk Member

No	Pernyataan
1	Menu yang tersedia mudah dipahami oleh pengguna baru.
2	Sistem memberikan arahan jelas jika terjadi kesalahan saat input.
3	Proses login dan logout bekerja tanpa hambatan.
4	Saya merasa chatbot mudah digunakan dan responsif.

Tabel 4.8 berikut merangkum skor dari masing-masing responden terhadap aspek ini:

Tabel 4.8 Hasil Perhitungan *Ease of Use* Member

No	Penguji	item kuisisioner ease of use			
		1	2	3	4
1	Member 1	5	4	4	5
2	Member 2	5	5	4	5
3	Member 3	4	5	4	4
4	Member 4	4	3	3	4
5	Member 5	4	5	5	5
6	Member 6	4	5	3	4

No	Penguji	item kuisioner ease of use			
		1	2	3	4
7	Member 7	3	4	4	4
8	Member 8	5	5	5	5
9	Member 9	5	5	5	5
10	Member 10	4	4	4	4
Total		43	45	41	46

Skor Ideal = $10 \times 4 \times 5 = 200$

$$NP = \frac{175}{200} \times 100\% = 87.5\%$$

Perhitungan diatas menunjukkan bahwa pengguna merasa sistem mudah digunakan, dengan nilai 87.5% dan dikategorikan sebagai Sangat Setuju.

d. Kemudahan Mempelajari (*Ease of Learning*)

Tabel 4.9 menyajikan butir-butir penilaian terkait kemudahan dalam mempelajari penggunaan sistem.

Tabel 4.9 Item Pernyataan *Ease of Learning* untuk Member

No	Pernyataan
1	Saya langsung paham cara mencari dan menyewa baju pada kunjungan pertama.
2	Sistem memberikan pengalaman yang intuitif tanpa harus melihat panduan.
3	Informasi dan tombol-tombol cukup jelas untuk digunakan.

Tabel 4.10 berikut merangkum skor dari masing-masing responden terhadap aspek ini:

Tabel 4.10 Hasil Perhitungan *Ease of Learning* Member

No	Penguji	item kuisioner ease Learning		
		1	2	3
1	Member 1	5	5	5
2	Member 2	5	5	4
3	Member 3	4	5	5

No	Penguji	item kuisioner ease Learning		
		1	2	3
4	Member 4	5	5	5
5	Member 5	5	5	5
6	Member 6	5	5	5
7	Member 7	4	4	3
8	Member 8	4	4	5
9	Member 9	5	5	5
10	Member 10	3	2	4
Total		45	45	46

Skor Ideal = $10 \times 3 \times 5 = 150$

$$NP = \frac{136}{150} \times 100\% = 90.7\%$$

Perhitungan diatas menunjukkan nilai 90.7%, yang mengindikasikan sistem ini sangat mudah dipelajari dan dipahami oleh pelanggan.

2. PENGUJIAN UAT ADMIN

a. Kegunaan (*Usefulness*)

Berikut Tabel 4.11 menunjukkan daftar pernyataan untuk aspek *usefulness* yang diajukan kepada admin:

Tabel 4.11 Item Pernyataan *Usefulness* untuk Admin

No	Pernyataan
1	Saya dapat menambah, mengedit, dan menghapus data baju.
2	Saya dapat melihat dan mencetak data penyewaan secara lengkap.
3	Bisa mengubah status penyewaan sesuai kebutuhan (Diterima, Selesai, Dibatalkan).
4	Informasi toko seperti kontak, aturan pemakaian, dan syarat ketentuan mudah diperbarui.
5	Sistem menyediakan laporan penyewaan berdasarkan rentang waktu yang fleksibel.
6	Saya dapat memantau data <i>member</i> yang sudah registrasi dengan mudah.

No	Pernyataan
7	Tampilan dashboard membantu saya memahami ringkasan aktivitas secara jelas.
8	Proses verifikasi bukti bayar mudah dilakukan.
9	Proses verifikasi bukti bayar mudah dilakukan.

Berikut Tabel 4.12 adalah hasil pengisian kuesioner oleh admin terhadap aspek *usefulness*:

Tabel 4.12 Hasil Perhitungan *Usefulness* Admin

No	Penguji	item kuisisioner <i>usefulness</i>									Total
		1	2	3	4	5	6	7	8	9	
1	Admin	4	4	4	4	4	5	5	4	4	38
2	Admin 2	5	5	5	5	5	5	5	5	5	45
3	Admin 3	5	4	5	5	4	5	5	5	5	43

Skor Ideal = $3 \times 9 \times 5 = 135$

$$NP = \frac{126}{135} \times 100\% = 93.3\%$$

Perhitungan ini menunjukkan bahwa admin menilai sistem sangat berguna dan dikategorikan sebagai Sangat Setuju.

b. Kepuasan (*Satisfaction*)

Berikut Tabel 4.13 berisi pernyataan yang mengevaluasi kepuasan admin terhadap sistem:

Tabel 4.13 Item Pernyataan *Satisfaction* untuk Admin

No	Pernyataan
1	Tampilan dan tata letak halaman memudahkan saya bekerja.
2	Proses input data tidak membingungkan.
3	Fitur cetak laporan bekerja seperti yang saya harapkan.
4	Sistem ini membantu saya menghindari kesalahan pencatatan.

Berikut Tabel 4.14 merangkum hasil kuisisioner pada aspek kepuasan:

Tabel 4.14 Hasil Perhitungan *Satisfaction* Admin

No	Penguji	item kuisioner <i>Satisfaction</i>				Total
		1	2	3	4	
1	Admin	4	4	4	4	16
2	Admin 2	5	4	4	5	18
3	Admin 3	5	4	4	4	17

Skor Ideal = $3 \times 4 \times 5 = 20$

$$NP = \frac{51}{60} \times 100\% = 85.0\%$$

Berdasarkan hasil tersebut, admin merasa puas terhadap sistem dan termasuk dalam kategori Setuju.

c. Kemudahan Penggunaan (*Ease of Use*)

Tabel 4.15 memuat item pernyataan terkait kemudahan penggunaan fitur sistem oleh admin:

Tabel 4.15 Item Pernyataan *Ease of Use* untuk Admin

No	Pernyataan
1	Sistem ini membantu saya menghindari kesalahan pencatatan.
2	Saya bisa mengoperasikan seluruh fitur admin tanpa kesulitan.
3	Sistem memberikan notifikasi jelas terhadap kesalahan input.
4	Proses penghapusan data aman dan mudah.
5	Saya dapat login/logout tanpa kendala teknis.

Berikut Tabel 4.16 menunjukkan hasil pengisian oleh admin:

Tabel 4.16 Hasil Perhitungan *Ease of Use* Admin

No	Penguji	item kuisioner ease of use					Total
		1	2	3	4	5	
1	Admin	4	4	4	4	4	20
2	Admin 2	5	5	5	5	5	25
3	Admin 3	4	4	4	3	5	20

Skor Ideal = $3 \times 5 \times 5 = 75$

$$NP = \frac{65}{75} \times 100\% = 86.7\%$$

Perhitungan ini menunjukkan bahwa sistem dianggap mudah digunakan oleh admin dan termasuk dalam kategori Setuju.

d. Kemudahan Mempelajari (*Ease of Learning*)

Tabel 4.17 menyajikan butir-butir penilaian terkait kemudahan dalam mempelajari penggunaan sistem:

Tabel 4.17 Item Pernyataan *Ease of Learning* untuk Admin

No	Pernyataan
1	Saya cepat memahami penggunaan fitur pengelolaan baju.
2	Menu laporan mudah dipahami sejak awal.
3	Saya tidak membutuhkan panduan tambahan untuk menjalankan sistem.

Berikut Tabel 4.18 menampilkan skor pengisian admin:

Tabel 4.18 Hasil Perhitungan *Ease of Learning* Admin

No	Penguji	item kuisisioner ease Learning			Total
		1	2	3	
1	Admin	3	4	5	10
2	Admin 2	5	5	5	15
3	Admin 3	3	3	3	9

Skor Ideal = $3 \times 3 \times 5 = 45$

$$NP = \frac{36}{45} \times 100\% = 80\%$$

Nilai ini menunjukkan bahwa sistem cukup mudah dipelajari, namun masih ada ruang untuk penyempurnaan agar lebih intuitif bagi admin.

4.6 PEMBAHASAN

Sistem informasi penyewaan pakaian adat berbasis web interaktif yang dikembangkan pada Toko Baju Bodo Kendari secara nyata mampu menjawab permasalahan yang telah diidentifikasi dalam tahap awal penelitian. Sebelumnya, proses bisnis masih berjalan secara manual, mulai dari pencatatan transaksi menggunakan Excel, komunikasi via WhatsApp, hingga pengecekan stok yang dilakukan secara langsung tanpa bantuan sistem. Kondisi tersebut mengakibatkan

sejumlah permasalahan seperti keterlambatan pelayanan, risiko kesalahan pencatatan, hingga tabrakan pemesanan dalam jadwal yang sama. Dengan kehadiran sistem ini, proses penyewaan menjadi lebih terstruktur, efisien, dan minim kesalahan.

Fitur utama yang dihadirkan dalam sistem ini meliputi manajemen stok otomatis yang bekerja secara *real-time*, layanan reservasi *online* yang responsif, serta fitur chatbot yang ditenagai oleh algoritma Natural Language Processing (NLP). Ketiga fitur ini menjadi elemen kunci yang mendukung peningkatan kualitas pelayanan dan efisiensi operasional. Pelanggan kini dapat melakukan pemesanan secara mandiri, memeriksa ketersediaan pakaian sesuai ukuran dan tanggal, serta mendapatkan jawaban otomatis dari chatbot terkait prosedur penyewaan dan ketentuan layanan. Bagi pemilik usaha, sistem memberikan kemudahan dalam memantau seluruh aktivitas penyewaan, mengelola data pelanggan, serta mencetak laporan penyewaan dalam format digital.

Hasil pengujian *User Acceptance Testing (UAT)* menunjukkan bahwa sistem diterima dengan sangat baik oleh pengguna. Dari sisi pelanggan, sistem memperoleh tingkat kegunaan sebesar 91%, kepuasan pengguna sebesar 86,7%, kemudahan penggunaan sebesar 87,5%, dan kemudahan mempelajari sistem mencapai 90,7%. Sementara itu, dari sisi administrator, nilai kegunaan berada pada angka 93,3%, dengan tingkat kepuasan sebesar 85%, kemudahan penggunaan sebesar 86,7%, dan kemudahan mempelajari sistem sebesar 80%. Nilai-nilai ini mencerminkan bahwa sistem telah mampu memenuhi ekspektasi pengguna, baik dari aspek fungsionalitas, kenyamanan dalam penggunaan, maupun efisiensi operasional, sehingga layak digunakan dalam implementasi nyata.

Selain hasil positif, terdapat pula sejumlah masukan dari pengguna yang menjadi catatan penting dalam pengembangan lebih lanjut. Di antaranya adalah usulan untuk meningkatkan kecerdasan chatbot agar mampu menangani pertanyaan yang lebih kompleks, serta permintaan integrasi sistem pembayaran otomatis guna mempercepat proses verifikasi transaksi. Selain itu, beberapa pengguna mengusulkan adanya fitur notifikasi otomatis melalui WhatsApp atau email untuk mengingatkan pelanggan mengenai jadwal pengembalian pakaian yang disewa.

Secara umum, sistem informasi yang dikembangkan telah berhasil mengintegrasikan kebutuhan operasional dengan solusi teknologi informasi yang modern dan adaptif. Sistem ini tidak hanya menjawab permasalahan yang dihadapi Toko Baju Bodo Kendari, tetapi juga menawarkan model pengelolaan penyewaan yang dapat direplikasi oleh pelaku usaha lain, khususnya dalam sektor UMKM berbasis budaya. Dengan dukungan antarmuka yang ramah pengguna, struktur data yang terorganisasi, serta fitur layanan mandiri yang memadai, sistem ini menunjukkan potensi besar sebagai instrumen digitalisasi bisnis penyewaan pakaian adat di era transformasi digital.

PERPUSTAKAAN
UNIVERSITAS JENDERAL ACHMAD YANU
YOGYAKARTA