

BAB 4

HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

Sistem Informasi Pencatatan Debit Air Bendungan Berbasis Web adalah aplikasi yang dirancang untuk membantu petugas dalam melakukan pencatatan, penyimpanan, dan pemantauan data debit air secara lebih terstruktur dan tepat. Sistem ini memungkinkan pencatatan data debit air harian pada beberapa waktu pengukuran, menampilkan informasi dalam bentuk tabel dan grafik untuk mempermudah analisis pola debit air, serta menyediakan fitur pembuatan laporan dalam format PDF. Selain itu, sistem memiliki fungsi pengelolaan data bendungan, kementren, pengguna, dan peran untuk mendukung pengaturan hak akses secara efektif. Aplikasi ini dikembangkan menggunakan framework Django dengan database MySQL, serta dapat diakses melalui web browser untuk mempermudah penyebaran informasi dan mendukung upaya mitigasi bencana banjir.

4.2 PERANCANGAN ANTARMUKA PENGGUNA

Perancangan antarmuka pengguna bertujuan untuk menghasilkan tampilan sistem yang sederhana, mudah dipahami, dan sesuai kebutuhan pengguna. Desain antarmuka mencakup elemen seperti form input, tabel data, dan grafik, agar proses pencatatan dan pemantauan debit air dapat dilakukan secara efisien.

4.2.1 Desain Halaman *Login*

Halaman *login* dirancang untuk memfasilitasi proses autentikasi pengguna sebelum mengakses sistem informasi pencatatan debit air. Pada gambar 4.1 menampilkan desain antarmuka halaman *login* secara keseluruhan yang muncul sebagai layar pertama saat pengguna mengakses sistem berikut.



Gambar 4.1 Halaman *Login*

Gambar 4.1 menampilkan halaman *login* yang menyajikan dua kolom input utama, yakni username dan password, serta dilengkapi dengan dua tombol aksi yaitu *Login* dan *Cancel*. Desain ini juga menampilkan ikon tetesan air dan teks sambutan "Silahkan *login* ke akun anda" sebagai elemen pendukung visual yang memperkuat konteks dan memberikan nuansa yang ramah bagi pengguna. Antarmuka dirancang secara sederhana dan mudah dibaca, sehingga memudahkan pengguna dalam memahami proses untuk masuk ke dalam sistem.

4.2.2 Desain Halaman *Dashboard*

Dashboard merupakan tampilan utama yang muncul setelah pengguna berhasil masuk ke sistem. Antarmuka ini dirancang untuk menyajikan informasi penting secara ringkas, interaktif, dan mudah dimengerti. Tujuan utamanya adalah untuk membantu petugas memantau kondisi debit air secara langsung serta mendukung pengambilan keputusan secara cepat. Desain halaman *dashboard* ditunjukkan pada Gambar 4.2 berikut.



Gambar 4.2 Halaman *Dashboard*

Gambar 4.2 menampilkan halaman *dashboard* yang menyajikan informasi debit air saat ini, prediksi harian, dan perubahan dari bulan sebelumnya. Terdapat grafik fluktuasi debit air harian dan tabel data berdasarkan tanggal. Di sisi kanan, terdapat deskripsi sistem, penjelasan indikator warna debit air, serta informasi bahwa akses sistem dibatasi untuk pengguna resmi. *Dashboard* ini berfungsi sebagai pusat pemantauan digital untuk mendukung mitigasi bencana.

4.2.3 Desain Halaman Data Debit Air

Halaman ini dirancang untuk menampilkan data pencatatan debit air secara lengkap dan terstruktur. Fitur ini memudahkan petugas dalam memantau, mengelola, serta mencetak data debit air berdasarkan periode tertentu. Antarmuka ini menyajikan informasi penting dalam bentuk tabel, dilengkapi dengan filter waktu dan opsi tampilan data. Desain halaman data debit air ditunjukkan pada Gambar 4.3 berikut.

SISTEM INFORMASI PENCATATAN DEBIT AIR

Home

Data Debit Air

Data Bendung

Data Kemantren

User

Role

Logout

Pencatatan Debit Air

Tambah debit air

Periode: April 2025

Ops: Full

Tanggal	Mataair/Lokasi/Debit Bendung											
	01/04/2025	02/04/2025	03/04/2025	04/04/2025	05/04/2025	06/04/2025	07/04/2025	08/04/2025	09/04/2025	10/04/2025	11/04/2025	12/04/2025
1	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
2	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
3	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
4	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
5	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
6	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
7	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
8	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
9	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
10	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
11	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
12	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
13	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
14	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
15	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
16	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
17	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
18	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
19	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
20	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
21	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
22	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
23	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
24	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
25	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
26	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
27	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
28	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
29	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
30	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32
31	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32	0.32

Cetak PDF

Gambar 4.3 Halaman Data Debit Air

Gambar 4.3 menampilkan halaman data debit air yang berisi tabel data hasil pencatatan debit air berdasarkan tanggal dan jam pengukuran (pagi, siang, sore). Tampilan ini juga menyediakan tombol Tambah debit air untuk menambahkan data baru, serta tombol Cetak PDF untuk mencetak atau menyimpan data ke dalam format dokumen. Fungsi ini membantu petugas dalam melakukan rekapitulasi dan dokumentasi yang efisien.

4.2.4 Desain Halaman Data Bendung

Halaman ini dirancang untuk mencatat dan menampilkan setiap data bendung yang ada di Kabupaten Cilacap. Informasi ini sangat penting sebagai acuan lokasi dalam pencatatan dan pemantauan debit air secara berkala. Desain halaman data bendung ditunjukkan pada Gambar 4.4 berikut.



Gambar 4.4 Halaman Data Bendung

Gambar 4.4 memperlihatkan tampilan halaman Data Bendung yang menyajikan daftar bendung lengkap dengan informasi seperti nama STA, sungai, desa, kecamatan, kabupaten, provinsi, serta koordinat lokasi. Tersedia pula fitur pencarian dan tombol "Tambah Bendung" untuk memudahkan pengguna dalam mengelola data. Desain halaman ini mendukung pengelolaan data infrastruktur pengairan secara digital, rapi, dan terpusat.

4.2.5 Desain Halaman Data Kemantren

Halaman ini berfungsi untuk mencatat dan mengelola data kemantren yang menjadi wilayah administratif pengawasan dan pengelolaan bendung di Kabupaten Cilacap. Pendataan kemantren penting agar sistem dapat memetakan wilayah kerja dan mempermudah pelaporan berdasarkan area pengelolaan. Desain halaman data Kemantren ditunjukkan pada Gambar 4.5 berikut.



Gambar 4.5 Halaman Data Kemantren

Gambar 4.5 menampilkan desain halaman Data Kemantren yang berisi daftar kemantren seperti Bantarsari 1, Bantarsari 2, dan Majenang. Pengguna dapat menambahkan data baru melalui tombol “+ Tambah Data” serta mengelola entri yang ada menggunakan tombol “Edit” dan “Hapus”. Tersedia pula kolom pencarian untuk mempermudah pengguna dalam menemukan data tertentu. Desain ini dibuat agar manajemen wilayah dapat dilakukan secara efisien dan terorganisir.

4.2.6 Desain Halaman User

Halaman ini dirancang untuk mengelola informasi pengguna sistem, termasuk nama, jenis kelamin, jabatan, unit kerja, kontak, serta kredensial *login*. Fitur ini penting untuk memastikan hanya pengguna yang memiliki otorisasi yang dapat mengakses dan mengelola data dalam sistem. Desain halaman *user* ditunjukkan pada Gambar 4.6 berikut.

SISTEM INFORMASI PENCATATAN DEBIT AIR

Data User

+ Tambah User

No	Nama	Jenis Kelamin	Jabatan	Unit Kerja	Nomor HP	Email	Username
1.	Daryanto	Laki-Laki	Mantri O&P	Bantarsari 2	082275630872	Daryanto@gmail.com	Daryanto O&P
2.	Yuni Andrianto	Laki-Laki	POB	Bantarsari 2	081293266755	Andrian12@gmail.com	Andri POB
3.	Yulianto	Laki-Laki	POB	Bantarsari 2	085050502030	yulianto23@gmail.com	Yuli POB
4.	Suyit	Laki-Laki	Mantri O&P	Bantarsari 1	082210755435	Suyit11@gmail.com	Suyit O&P
5.	Rawono	Laki-Laki	POB	Bantarsari 1	082266719036	Rawono@gmail.com	Rawono POB
Dist.							

Gambar 4.6 Halaman *User*

Gambar 4.6 memperlihatkan tampilan halaman *Data User* yang memuat tabel berisi daftar pengguna aktif dalam sistem. Setiap entri mencakup informasi lengkap seperti nama pengguna, jabatan, unit kerja, nomor HP, *email*, dan *username*. Tersedia juga tombol “+ Tambah User” dan kolom pencarian untuk mempermudah penambahan serta pencarian data. Dengan fitur ini, admin dapat dengan mudah memverifikasi dan mengelola hak akses pengguna.

4.2.7 Desain Halaman *Role*

Tampilan ini dirancang untuk mengelola peran setiap pengguna sesuai dengan lokasi bendungan yang mereka tangani. Tujuannya adalah untuk mengatur hak akses secara terstruktur berdasarkan tanggung jawab masing-masing. Desain halaman *role* ditunjukkan pada Gambar 4.7 berikut.



Gambar 4.7 Halaman *Role*

Gambar 4.7 menunjukkan halaman *Data Role* yang berisi daftar nama *role* seperti "Admin bendung kebogoran" hingga "Admin bendung cijalau", lengkap dengan informasi lokasi tempat bertugas. Masing-masing entri dilengkapi tombol "*Edit*" dan "*Hapus*" untuk mempermudah proses pengelolaan. Di bagian atas juga tersedia fitur pencarian dan tombol "+ Tambah User" untuk menambahkan *role* baru. Elemen-elemen ini berfungsi untuk memastikan sistem berjalan dengan tertib dan akses pengguna sesuai dengan perannya.

4.2.8 Desain Halaman *Logout*

Tampilan *logout* dibuat sebagai konfirmasi akhir sebelum pengguna keluar dari sistem. Desain ini bertujuan memastikan bahwa semua aktivitas dan penyimpanan data telah selesai dilakukan. Desain halaman *logout* ditunjukkan pada Gambar 4.8 berikut.



Gambar 4.8 Halaman *Logout*

Gambar 4.8 menampilkan halaman *logout* dengan *ilustrasi* pengguna yang meninggalkan ruangan. Di bagian atas, terdapat pertanyaan “Apakah Anda yakin ingin keluar?” disertai dengan imbauan untuk memeriksa kembali penyimpanan data. Di bagian bawah, tersedia tombol “Sing Out” yang dapat ditekan untuk keluar dari aplikasi sepenuhnya.

4.3 IMPLEMENTASI DESAIN ANTARMUKA

Antarmuka sistem ini dibuat dengan memperhatikan aspek kemudahan akses, penyusunan elemen yang rapi, serta tampilan yang dapat menyesuaikan berbagai jenis perangkat. Setiap komponen antarmuka dirancang agar mudah dimengerti dan mendukung interaksi pengguna secara nyaman dan efisien.

4.3.1 Implementasi Halaman *Landing Page*

Halaman *landing page* merupakan tampilan pertama yang muncul saat pengguna mengakses sistem. Halaman ini dirancang untuk memberikan pemahaman awal mengenai peran dan tujuan dari Sistem Informasi Pencatatan Debit Air Cilacap (SIPDA Cilacap). Implementasi halaman *landing page* ditunjukkan pada Gambar 4.9 berikut.



Gambar 4.9 Implementasi Halaman *Landing Page*

Gambar 4.9 menampilkan antarmuka halaman awal yang terdiri dari dua bagian utama:

1. Bagian kiri berisi deskripsi singkat mengenai sistem yang bertujuan membantu petugas maupun instansi dalam proses pencatatan, pemantauan, dan analisis data debit air secara digital. Sistem ini juga mendukung pengambilan keputusan yang cepat dan berbasis data.
2. Bagian kanan menampilkan gambar bendungan sebagai elemen visual pendukung yang merepresentasikan konteks penggunaan sistem.

Tombol "Login" disediakan untuk mengarahkan pengguna menuju halaman masuk ke sistem. Desain ini disusun secara ringkas dan informatif agar pengguna dapat langsung memahami fungsi utama dari aplikasi.

4.3.2 Implementasi Halaman *Login*

Halaman *login* berfungsi sebagai gerbang awal bagi pengguna untuk mengakses sistem dan dirancang dengan tampilan yang sederhana, jelas, dan mudah dipahami. Implementasi halaman *login* ditunjukkan pada Gambar 4.10 berikut.



Gambar 4.10 Implementasi Halaman *Login*

Gambar 4.10 menunjukkan antarmuka *login* yang menampilkan ikon tetesan air sebagai simbol sistem, kalimat arahan “Silahkan *login* ke akun anda”, dua kolom isian untuk *username* dan *password*, serta dua tombol aksi yaitu “*Login*” dan “*Cancel*”. Jika dibandingkan dengan desain UI/UX sebelumnya, tampilan *login* ini menunjukkan perbaikan dari sisi estetika dan kenyamanan pengguna. Latar belakang gelap transparan membantu pengguna lebih fokus pada formulir *login*, sementara tata letak dan ukuran komponen yang lebih tertata membuat proses *login* lebih nyaman. Selain itu, penggunaan font yang konsisten, ikon yang menarik, serta desain tombol yang lebih modern menjadikan tampilan antarmuka ini terlihat lebih profesional dan mudah diakses di berbagai perangkat.

4.3.3 Implementasi Halaman *Dashboard*

Halaman *dashboard* berperan sebagai pusat kendali utama yang menyajikan informasi penting secara ringkas dan mudah dipahami oleh pengguna. Implementasi halaman dashboard ditunjukkan pada Gambar 4.11 berikut.



Gambar 4.11 Implementasi Halaman *Dashboard*

Gambar 4.11 menunjukkan implementasi halaman *dashboard* pada Sistem Informasi Pencatatan Debit Air yang menampilkan informasi penting seperti debit air saat ini, prediksi kondisi harian, serta persentase perubahan debit dibanding bulan sebelumnya. Di bagian tengah halaman, terdapat grafik interaktif yang menggambarkan rata-rata debit air bulanan berdasarkan bendung yang dipilih, sementara di sisi kanan tersedia keterangan warna status debit air (hijau untuk aman, kuning untuk waspada, dan merah untuk bahaya). Jika dibandingkan dengan desain UI/UX sebelumnya, tampilan dashboard ini menunjukkan peningkatan visual dan kemudahan navigasi. Susunan elemen lebih rapi, penggunaan ikon dan grafik yang lebih jelas, serta indikator warna yang informatif membantu pengguna memahami situasi debit air secara cepat. Selain itu, menu samping yang terstruktur memudahkan akses ke fitur lainnya, menjadikan halaman dashboard lebih efisien dan ramah pengguna.

4.3.4 Implementasi Halaman Data Debit Air

Halaman Data Debit Air merupakan fitur utama yang menampilkan pencatatan debit air berdasarkan tanggal, jam, dan lokasi bendung. Terdapat filter pencarian berdasarkan periode dan bendung, serta fitur Edit, Hapus, dan Cetak PDF. Pengguna juga dapat menambahkan data baru melalui tombol Tambah Debit Air dengan mengisi formulir berisi tanggal, nama bendung, jam pengukuran, tinggi muka air (H), dan debit (Q). Dibandingkan desain sebelumnya, tampilan baru ini

lebih rapi, mudah dinavigasi, dan efisien untuk mendukung pencatatan data harian secara digital. Implementasi halaman data debit air dan input data debit ditunjukkan pada Gambar 4.12 dan 4.13 berikut.

SIPDA Cilacap

Dashboard

Data Debit Air

Data Bendung

Data Kemantran

User

Role

Sign Out

SISTEM INFORMASI PENCATATAN DEBIT AIR

Data Debit Air

+ Tambah Debit Air

Periode (Bulan & Tahun)

Tanggal 1-13

Bendung

PDAB Kebogoran

Filter

Tanggal	Muka Air Limpas dan DEK Ringkasan				Debit Pintu Intik Ringkasan								Debit Total (m³/dt)	Aksi		
	H (m)	Q (m³/dt)	H (m)	Q (m³/dt)	Jan 07-09		Jan 10-12		Jan 13-15		Jan 16-18					
01	0.14	4.10	0.14	4.10	0.14	4.10	0.17	0.24	0.26	0.126	0.17	0.24	0.26	0.126	4.000	EditHapus
02	0.12	3.317	0.12	3.317	0.12	3.317	0.17	0.24	0.26	0.126	0.17	0.24	0.26	0.126	3.900	EditHapus
03	0.13	3.317	0.13	3.317	0.13	3.317	0.17	0.24	0.26	0.126	0.17	0.24	0.26	0.126	3.800	EditHapus
04	0.16	4.107	0.16	4.107	0.16	4.107	0.17	0.24	0.27	0.144	0.17	0.24	0.27	0.144	3.400	EditHapus
05	0.21	7.870	0.21	7.870	0.21	7.870	0.17	0.24	0.27	0.144	0.17	0.24	0.27	0.144	3.000	EditHapus

Unduh PDF

Gambar 4.12 Implementasi Halaman Data Debit Air

SISTEM INFORMASI PENCATATAN DEBIT AIR

Tambah Data Debit Air (diinput dalam satuan meter (m)).

Tanggal dd/mm/yyyy Bendung Pilih Bendung

Jam 07.00
 H Muka Air Limpas Bendung H Kiri Pintu Intik Bendung H Kanan Pintu Intik Bendung
 (m) (m) (m)

Jam 12.00
 H Muka Air Limpas Bendung H Kiri Pintu Intik Bendung H Kanan Pintu Intik Bendung
 (m) (m) (m)

Jam 17.00
 H Muka Air Limpas Bendung H Kiri Pintu Intik Bendung H Kanan Pintu Intik Bendung
 (m) (m) (m)

Tutup Tambah

Gambar 4.13 Input Data Debit

Pada tahap implementasi sistem, kode yang terdapat dalam file `views.py` ini bertugas untuk menangani pemrosesan dan tampilan data debit air berdasarkan filter waktu dan lokasi bendung.

```
@login_required
def daftar_debit_air(request):
    now = timezone.now()
    periode = request.GET.get('periode', f"{now.year}-{now.month:02d}") # format: YYYY-MM
    opsi = request.GET.get('opsi', 'Full')
```

```

id_bendung = request.GET.get('bendung')

try:
    tahun, bulan = map(int, periode.split('-'))
except:
    tahun = now.year
    bulan = now.month

debit_air_qs = DebitAir.objects.all()

if id_bendung:
    debit_air_qs = debit_air_qs.filter(bendungan_id=id_bendung)

debit_air_qs = debit_air_qs.filter(tanggal__month=bulan,
                                    tanggal__year=tahun)

if opsi == 'Tanggal 1-15':
    debit_air_qs = debit_air_qs.filter(tanggal__day__lte=15)
elif opsi == 'Tanggal 16-31':
    debit_air_qs = debit_air_qs.filter(tanggal__day__gte=16)

debit_air_qs = debit_air_qs.order_by('tanggal')
bendung_list = Bendung.objects.all()

```

```

today_str = timezone.now().strftime('%Y-%m') # Contoh:
'2025-06'

context = {
    'data_debit': debit_air_qs,
    'bulan': int(bulan),
    'tahun': int(tahun),
    'opsi': opsi,
    'id_bendung': int(id_bendung) if id_bendung else None,
    'bendung_list': bendung_list,
    'today_str': today_str,
}
return render(request, 'debit_air.html', context)

def rumus_q_limpas(h):
    """Rumus untuk Q limpasan (h_07, h_12, h_17)"""
    return round(1 * 46.8 * ((2/3) * h) * math.sqrt(9.81 * ((2/3)
* h)), 3)

def rumus_q_kiri(h):
    """Rumus untuk Q pintu kiri"""
    return round(1.71 * 2 * (h ** 1.5), 3)

def rumus_q_kanan(h):
    """Rumus untuk Q pintu kanan"""
    return round(1.71 * 0.6 * (h ** 1.5), 3)

def boleh_ubah_data(user, tanggal_data):

```

```

    """Cek apakah user boleh mengedit/hapus data berdasarkan
    tanggal dan jabatan."""
    if user.is_superuser:
        return True
    if hasattr(user, 'jabatan') and user.jabatan == "Juru
    Operasional dan Pemeliharaan":
        return True

    # Cek apakah data berada di bulan dan tahun sekarang
    today = timezone.now().date()
    return tanggal_data.month == today.month and
    tanggal_data.year == today.year

```

Fungsi `daftar_debit_air` digunakan untuk mengambil data sesuai dengan bulan, tahun, rentang tanggal tertentu (1–15 atau 16–31), serta bendung yang dipilih, lalu hasilnya ditampilkan pada template `debit_air.html`. Terdapat pula tiga fungsi perhitungan debit air yaitu `rumus_q_limpas`, `rumus_q_kiri`, dan `rumus_q_kanan`, masing-masing digunakan untuk menghitung nilai debit berdasarkan tinggi muka air dengan formula tertentu. Fungsi tambahan `boleh_ubah_data` digunakan untuk memastikan apakah seorang pengguna memiliki izin untuk mengubah atau menghapus data, dengan kriteria tertentu seperti status superuser, jabatan sebagai “Juru Operasional dan Pemeliharaan,” atau data masih dalam bulan dan tahun berjalan. Secara keseluruhan, fungsi-fungsi ini membantu pengelolaan data debit air agar berjalan aman dan sesuai dengan otoritas pengguna.

4.3.5 Implementasi Halaman Data Bendung

Pada gambar 4.14 dan 4.15 menggambarkan tampilan halaman Data Bendung serta input data bendung dalam sistem pencatatan debit air yang berisi daftar seluruh bendung di Kabupaten Cilacap, lengkap dengan informasi seperti nama dan nomor STA, sungai, daerah aliran (CA), desa, kecamatan, kabupaten, provinsi, serta koordinat lintang dan bujur. Setiap data juga dikaitkan dengan kemantren tertentu dan dilengkapi dengan opsi untuk mengedit atau menghapus melalui tombol aksi. Penambahan data bendung baru dapat dilakukan melalui tombol “+ Tambah Bendung” yang akan mengarahkan pengguna ke form input berisi kolom-kolom penting seperti nama bendung, lokasi administratif, dan titik koordinat. Fitur ini dirancang agar proses pencatatan data bendung dapat dilakukan secara terstruktur dan efisien, Seperti gambar berikut.

No.	Nama STA	Nomor STA	Sungai	CA	Desa	Kecamatan	Kabupaten	Provinsi	Koordinat LS	Koordinat BT	Kemantren	Aksi
1	PDAB Kebogoran	-	Cimeneng	-	Kamulyan	Bantarsari	Cilacap	Jawa Tengah	-7,50782, 108, 87814, 35,9m, 295°	-	Bantarsari 2	Edit Hapus
2	PDAB Cijati	-	Cijati	-	Cijati	Majenang	Cilacap	Jawa Tengah	-	-	Majenang	Edit Hapus
3	PDAB Cijalu	-	Cijalu	-	Jenang	Majenang	Cilacap	Jawa Tengah	-	-	Majenang	Edit Hapus
4	PDAB Cilumuh	-	Cilumuh	-	Rejodadi	Cimanggu	Cilacap	Jawa Tengah	-	-	Majenang	Edit Hapus

Gambar 4.14 Implementasi Halaman Data Bendung

Gambar 4.15 Input Data Bendung

Berikut kode yang merupakan bagian dari file `views.py` yang berfungsi untuk mengatur proses penambahan data bendung pada sistem. Fungsi `tambah_bendung` menggunakan decorator `login_required` untuk memastikan hanya pengguna yang telah melakukan *login* yang dapat mengakses fitur ini.

```
@login_required
def tambah_bendung(request):
    if request.method == 'POST':
        nama_sta = request.POST.get('nama_sta')
        nomor_sta = request.POST.get('nomor_sta')
        sungai = request.POST.get('sungai')
        ca = request.POST.get('ca')
        desa = request.POST.get('desa')
        kecamatan = request.POST.get('kecamatan')
```

```

    kabupaten = request.POST.get('kabupaten')
    provinsi = request.POST.get('provinsi')
    koordinat_ls = request.POST.get('koordinat_ls')
    koordinat_bt = request.POST.get('koordinat_bt')
    kemantren_id = request.POST.get('kemantren')

    kemantren = get_object_or_404(Kemantren, id=kemantren_id)

    Bendung.objects.create(
        nama_sta=nama_sta,
        nomor_sta=nomor_sta,
        sungai=sungai,
        ca=ca,
        desa=desa,
        kecamatan=kecamatan,
        kabupaten=kabupaten,
        provinsi=provinsi,
        koordinat_ls=koordinat_ls,
        koordinat_bt=koordinat_bt,
        kemantren=kemantren
    )

    return JsonResponse({'message': 'Data bendung berhasil
    ditambahkan!'})
    return HttpResponseBadRequest('Invalid request')

```

Saat pengguna mengirimkan data melalui metode POST, sistem akan mengambil seluruh informasi yang diisikan dari form, seperti nama dan nomor STA, nama sungai, wilayah daerah aliran sungai (CA), lokasi administratif (desa, kecamatan, kabupaten, provinsi), hingga koordinat lintang dan bujur. Selain itu, ID kemantren yang dipilih juga akan diproses untuk diambil datanya menggunakan fungsi `get_object_or_404` dari model `Kemantren`. Setelah semua data berhasil dikumpulkan, fungsi ini akan menyimpan data baru ke dalam database melalui model `Bendung` dengan metode `objects.create()`. Jika proses penambahan berhasil, sistem akan mengirimkan respon dalam format JSON yang menyatakan bahwa data telah berhasil ditambahkan. Namun jika metode yang digunakan bukan POST, maka sistem akan mengembalikan respon `HttpResponseBadRequest` sebagai penanda bahwa permintaan tidak valid.

4.3.6 Implementasi Halaman Data Kemantren

Pada gambar 4.16 memperlihatkan tampilan halaman "Data Kemantren" pada Sistem Informasi Pencatatan Debit Air. Halaman ini digunakan untuk manajemen data kemantren yang ada di wilayah Kabupaten Cilacap. Informasi

ditampilkan dalam bentuk tabel yang berisi daftar nama kemantren beserta tombol Edit dan Hapus untuk setiap baris data. Selain itu, pengguna dapat menambahkan kemantren baru melalui tombol Tambah Kemantren yang tersedia di bagian atas. Seperti pada gambar berikut.



Gambar 4.16 Implementasi Halaman Data Kemantren

Fungsi-fungsi yang digunakan untuk mengelola data kemantren dalam sistem ini diimplementasikan di file `views.py`, mencakup proses penampilan, penambahan, pembaruan, dan penghapusan data.

```
@login_required
def daftar_kemantren(request):
    kemantren_list = Kemantren.objects.all()
    return render(request, 'kemantren.html', {'kemantren_list':
kemantren_list})

@login_required
def tambah_kemantren(request):
    if request.method == 'POST':
        nama = request.POST.get('nama')

        Kemantren.objects.create(
            nama = nama,
        )

        return JsonResponse({'message': 'Data kemantren berhasil
ditambahkan!'})
    return HttpResponseRedirect('Invalid request')

@login_required
def edit_kemantren(request, id):
    kemantren = get_object_or_404(Kemantren, id=id)

    if request.method == 'POST':
        kemantren.nama = request.POST.get('nama')
```

```

        kemantren.save()
        return JsonResponse({'message': 'Data kemantren berhasil
diperbarui!'})
        return HttpResponseRedirect('Invalid request')

@login_required
@csrf_exempt
def hapus_kemantren(request, id):
    if request.method == 'DELETE':
        kemantren = get_object_or_404(Kemantren, id=id)
        kemantren.delete()
        return JsonResponse({'message': 'Data kemantren berhasil
dihapus'})
        return HttpResponseRedirect('Invalid request')

```

Fungsi `daftar_kemantren` digunakan untuk mengambil seluruh data kemantren dari database dan menampilkannya melalui template `kemantren.html`. Sementara itu, fungsi `tambah_kemantren` memungkinkan pengguna menambahkan entri baru melalui metode POST dengan hanya mengisi nama kemantren. Untuk mengubah data yang sudah ada, digunakan fungsi `edit_kemantren`, yang akan memperbarui nama kemantren berdasarkan ID yang diterima. Terakhir, penghapusan data dilakukan melalui fungsi `hapus_kemantren` menggunakan metode `DELETE`, di mana data yang sesuai ID akan dihapus dari basis data. Seluruh fungsi ini dilindungi oleh dekorator `login_required`, sehingga hanya pengguna yang sudah masuk ke sistem yang dapat mengaksesnya.

4.3.7 Implementasi Halaman User

Pada gambar 4.17 menunjukkan halaman *Daftar User* yang menampilkan data pengguna seperti *username*, nama, jabatan, kemantren, *email*, nomor HP, dan *role*. *Admin* dapat menambah, mengedit, atau menghapus *user* melalui tombol aksi yang tersedia, serta memanfaatkan fitur pencarian untuk memudahkan pengelolaan data. Seperti pada gambar berikut.



Gambar 4.17 Implementasi Halaman User

Kode fungsi dibawah ini digunakan untuk mengelola proses penambahan data akun pengguna ke sistem. Fungsi ini ditulis dalam file *views.py* dan aktif saat sistem menerima permintaan POST.

```
@login_required
@csrf_exempt
def tambah_user(request):
    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')
        nama = request.POST.get('nama')
        jenis_kelamin = request.POST.get('jenis_kelamin')
        jabatan = request.POST.get('jabatan')
        unit_kerja_id = request.POST.get('unit_kerja')
        email = request.POST.get('email')
        no_hp = request.POST.get('no_hp')
        role_id = request.POST.get('role')

        # Cek apakah username sudah ada
        if User.objects.filter(username=username).exists():
            return JsonResponse({'error': 'Username sudah digunakan'}, status=400)

        try:
            User.objects.create(
                username=username,
                password=make_password(password), # amankan password
                nama=nama,
                jenis_kelamin=jenis_kelamin,
                jabatan=jabatan,
                unit_kerja_id=unit_kerja_id,
                email=email,
                no_hp=no_hp,
                role_id=role_id
            )
```

```

        return JsonResponse({'message': 'User berhasil
        ditambahkan'})
    except Exception as e:
        return JsonResponse({'error': str(e)}, status=500)

    return HttpResponseRedirect('Invalid request')

```

Fungsi `tambah_user` bertugas menyimpan informasi pengguna baru ke basis data. Fungsi ini terlebih dahulu memverifikasi bahwa username belum digunakan. Jika valid, data seperti nama lengkap, jabatan, jenis kelamin, unit kerja, alamat *email*, nomor handphone, dan peran (*role*) akan disimpan. *Password* pengguna juga dienkripsi menggunakan `make_password` demi keamanan. Jika proses berjalan lancar, sistem memberikan respons sukses, namun jika terjadi kesalahan, akan muncul pesan *error*.

4.3.8 Implementasi Halaman *Role*

Pada gambar 4.18 memperlihatkan tampilan halaman Daftar *Role* pada Sistem Informasi Pencatatan Debit Air. Halaman ini digunakan untuk mengatur peran pengguna yang terhubung dengan bendungan tertentu. Setiap baris mencantumkan nama *role*, lokasi bendung, serta dilengkapi dengan tombol *Edit* dan Hapus untuk mempermudah proses pengelolaan data. Selain itu, terdapat tombol + tambah *role* yang memungkinkan penambahan *role* baru ke dalam sistem. Seperti pada gambar berikut.



Gambar 4.18 Implementasi Halaman *Role*

Kode dibawah ini pada file *views.py* ini berfungsi untuk mengelola data role dalam Sistem Informasi Pencatatan Debit Air.

```

@login_required
def role_list(request):
    roles = Role.objects.select_related('tempat').all()
    bendung_list = Bendung.objects.all()
    return render(request, 'role.html', {
        'role_list': roles,
        'bendung_list': bendung_list
    })

@login_required
def tambah_role(request):
    if request.method == 'POST':
        nama_role = request.POST.get('nama_role')
        id_tempat = request.POST.get('tempat')
        tempat = get_object_or_404(Bendung, id=id_tempat)

        Role.objects.create(nama_role=nama_role, tempat=tempat)

        return JsonResponse({'message': 'Role berhasil
ditambahkan'})
    return HttpResponseRedirect('Invalid request')

@login_required
def edit_role(request, id):
    role = get_object_or_404(Role, id=id)
    if request.method == 'POST':
        role.nama_role = request.POST.get('nama_role')
        id_tempat = request.POST.get('tempat')
        role.tempat = get_object_or_404(Bendung, id=id_tempat)
        role.save()
        return JsonResponse({'message': 'Role berhasil
diperbarui'})
    return HttpResponseRedirect('Invalid request')

@login_required
@csrf_exempt
def hapus_role(request, id):
    if request.method == 'DELETE':
        role = get_object_or_404(Role, id=id)
        role.delete()
        return JsonResponse({'message': 'Role berhasil dihapus'})
    return HttpResponseRedirect('Invalid request')

```

Fungsinya meliputi menampilkan daftar role (*role_list*), menambahkan *role* baru (*tambah_role*), mengedit data role (*edit_role*), dan menghapus *role* (*hapus_role*). Seluruh fungsi dibatasi oleh *@login_required*, yang artinya hanya dapat diakses oleh pengguna yang telah login. Setiap role dikaitkan dengan data bendung sebagai lokasi tugasnya, dan proses pengolahan datanya dilakukan menggunakan metode POST dan *DELETE* melalui model *Role* dan *Bendung*.

4.3.9 Implementasi Halaman Logout

Tampilan *logout* berfungsi sebagai akhir dari sesi pengguna, dirancang agar proses keluar dari sistem berjalan dengan aman. Pengguna diingatkan untuk memastikan data telah disimpan sebelum meninggalkan sistem. Implementasi halaman *logout* ditunjukkan pada Gambar 4.19 berikut.

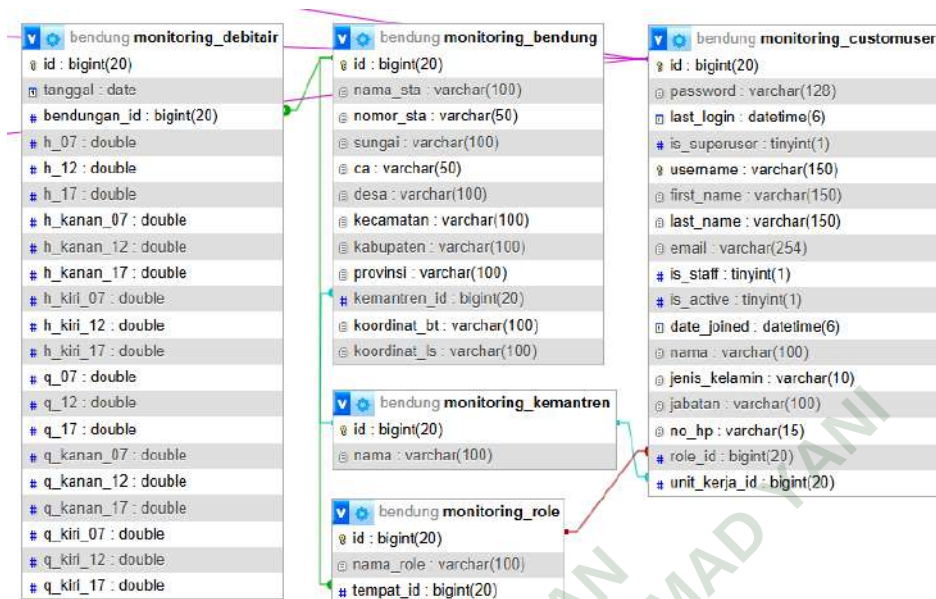


Gambar 4.19 Implementasi Halaman Logout

Gambar 4.19 menunjukkan antarmuka logout yang menyertakan ilustrasi, pesan singkat, dan tombol "Sign Out". Desain ini dibuat ringkas agar pengguna memahami dengan jelas tindakan yang akan dilakukan saat keluar dari aplikasi.

4.4 BASIS DATA

Sistem informasi ini menggunakan basis data relasional yang terdiri dari beberapa tabel, seperti tabel *user*, *role*, *bendung*, *data debit air*, dan *kemantren*. Setiap tabel memiliki atribut yang telah disesuaikan dengan kebutuhan sistem. Relasi antar tabel dirancang untuk memudahkan integrasi data dan proses pencatatan secara efisien. Gambar 4.20 berikut menunjukkan diagram relasi antar tabel dalam basis data sistem.



Gambar 4.20 Basis Data

Basis data pada Sistem Informasi Pencatatan Debit Air Bendungan Berbasis Web dirancang menggunakan model relasional yang terstruktur untuk mendukung kebutuhan pencatatan, penyimpanan, dan pengelolaan data secara konsisten dan terintegrasi. Gambar di atas menampilkan relasi antar tabel utama dalam sistem ini, yang dapat dijelaskan sebagai berikut:

1. Tabel monitoring debit air
 - 1) Berfungsi mencatat hasil pengukuran debit air harian.
 - 2) Memuat atribut seperti tanggal pencatatan, tinggi muka air (`h_07`, `h_12`, `h_17`), serta nilai debit air (`q_07`, `q_12`, `q_17`) pada waktu pengukuran pagi, siang, dan sore.
 - 3) Kolom `bendungan_id` menjadi foreign key yang menghubungkan data debit air dengan tabel `monitoring_bendung`, memastikan setiap pencatatan terkait dengan bendung tertentu.
2. Tabel monitoring bendung
 - 1) Menyimpan data detail mengenai bendungan.
 - 2) Atributnya meliputi `nama_sta`, `nomor_sta`, `sungai`, `desa`, `kecamatan`, `kabupaten`, `provinsi`, serta koordinat geografis (`koordinat_bt` dan `koordinat_ls`).

- 3) Kolom `kemantren_id` adalah foreign key yang terhubung ke tabel `monitoring_kemantren`, untuk mengelompokkan bendung berdasarkan wilayah administratif.
3. Tabel monitoring kemantren
 - 1) Menyimpan informasi wilayah kemantren atau distrik administratif.
 - 2) Hanya memiliki atribut `id` dan `nama`, yang digunakan untuk mengelompokkan bendung pada wilayah tertentu.
 - 3) Relasi ini mendukung pemetaan area pengawasan secara lebih terstruktur.
4. Tabel monitoring customuser
 - 1) Menyimpan data pengguna sistem, baik petugas maupun admin.
 - 2) Memiliki atribut login standar (`username`, `password`, `email`), informasi pribadi (`nama`, `jenis kelamin`, `jabatan`, `nomor HP`), serta data manajemen akun (`is_active`, `is_staff`, `is_superuser`).
 - 3) Kolom `role_id` adalah foreign key ke tabel `monitoring_role` untuk menentukan hak akses pengguna.
 - 4) Kolom `unit_kerja_id` adalah foreign key ke `monitoring_kemantren`, untuk mengaitkan pengguna dengan wilayah tugas tertentu.
5. Tabel monitoring role
 - 1) Menyimpan informasi mengenai peran (role) pengguna dalam sistem.
 - 2) Memiliki atribut `nama_role` untuk mendefinisikan peran seperti admin, superadmin, atau petugas.
 - 3) Kolom `tempat_id` adalah foreign key ke tabel `monitoring_kemantren`, yang memungkinkan pembatasan peran berdasarkan wilayah kerja.

Relasi antar tabel tersebut dirancang untuk menjaga integritas data dan memastikan keterhubungan yang jelas antar entitas. Struktur basis data ini mendukung sistem agar mampu menyimpan data debit air secara detail, mengaitkan catatan debit dengan bendungan dan wilayah administrasi, serta mengelola akses pengguna sesuai peran dan lokasi kerja masing-masing.

4.5 PENGUJIAN SISTEM (*BLACK BOX TESTING*)

Pengujian sistem bertujuan untuk memastikan bahwa seluruh fungsi dalam aplikasi berjalan sesuai dengan kebutuhan pengguna. Teknik yang digunakan adalah *black box testing*, yakni metode pengujian yang mengevaluasi keluaran sistem berdasarkan masukan tertentu tanpa melihat struktur internal program. Proses pengujian ini mencakup fitur-fitur utama seperti autentikasi pengguna, pencatatan debit air, pengelolaan data bendung, manajemen pengguna, serta proses keluar dari sistem. Melalui pengujian ini, dapat diketahui apakah sistem memberikan respon yang tepat terhadap setiap input, sehingga hasilnya menjadi acuan untuk perbaikan guna menjamin performa sistem yang maksimal.

4.5.1 Pengujian Fitur Login dan Logout

Pengujian fitur login dan logout dapat dilihat pada table 4.1 berikut.

Tabel 4.1 Tabel Pengujian Login dan Logout

Fitur yang diuji	Input	Langkah pengujian	Output yang diharapkan	Hasil
Login	Username dan password valid	Pengguna mengisi username dan password dengan benar dan menekan tombol "Login"	Pengguna diarahkan ke halaman dashboard	✓
	Username salah	Pengguna mengisi username yang tidak terdaftar dan menekan tombol "Login"	Muncul pesan kesalahan "username atau password salah"	✓
	Password salah	Pengguna mengisi username benar dan password salah	Muncul pesan kesalahan "username atau password salah"	✓
	Kolom input kosong	Pengguna menekan tombol "Login" tanpa mengisi username atau password	Muncul pesan peringatan "please fill out this field"	✓

Fitur yang diuji	Input	Langkah pengujian	Output yang diharapkan	Hasil
Logout	-	Pengguna menekan tombol “Logout” di halaman dashboard	Pengguna diarahkan kembali ke halaman login	✓

4.5.2 Pengujian Fitur Dashboard

Pengujian fitur dashboard dapat dilihat pada table 4.2 berikut.

Tabel 4.2 Tabel Pengujian Dashboard

Fitur yang Diuji	Input	Langkah Pengujian	Output yang Diharapkan	Hasil
Akses Halaman	Pengguna login sebagai admin	Admin berhasil login dan diarahkan ke halaman dashboard	Halaman dashboard ditampilkan lengkap dengan grafik dan indikator visual	✓
Visualisasi Data	Data debit tersedia	Grafik ditampilkan berdasarkan data debit air dari database	Grafik garis muncul sesuai fluktuasi debit air harian	✓
Indikator Warna Kuning	Nilai debit dalam aliran kecil	Debit air berada di bawah ambang waspada	Indikator kuning muncul sebagai kondisi air dalam aliran kecil	✓
Indikator Warna Merah	Nilai debit melebihi batas aman	Debit air melampaui ambang bahaya	Indikator merah muncul sebagai tanda kondisi bahaya	✓
Indikator Warna Hijau	Nilai debit dalam kondisi aman	Debit air berada di bawah ambang waspada	Indikator hijau muncul sebagai tanda kondisi aman	✓

4.5.3 Pengujian Input Data Debit Air

Pengujian input data debit air dapat dilihat pada table 4.3 berikut.

Tabel 4.3 Tabel Pengujian Debit Air

Fitur yang Diuji	Input	Langkah Pengujian	Output yang Diharapkan	Hasil
Akses Halaman Data Debit	Pengguna login dan klik menu “Data Debit Air”	Pengguna memilih menu data debit di sidebar	Halaman data debit ditampilkan lengkap dengan tabel data	✓
Tambah Data Debit	Data valid: tanggal, bendung, tinggi muka air	Mengisi form tambah data debit dan klik tombol simpan	Data baru muncul pada tabel dan tersimpan di database	✓
Edit Data Debit	Perubahan nilai debit	Klik tombol edit, ubah data lalu simpan	Perubahan tersimpan dan ditampilkan kembali dengan nilai terbaru	✓
Hapus Data Debit	Pilih data lalu klik hapus	Pengguna memilih salah satu data dan menghapusnya	Data berhasil dihapus dari tampilan dan basis data	✓
Filter Data Berdasarkan Periode	Periode bulan dan tahun tertentu	Memilih filter bulan dan tahun pada dropdown	Hanya data pada periode terpilih yang ditampilkan	✓
Perhitungan Debit Otomatis	Input tinggi muka air (h)	Sistem menghitung debit (Q) secara otomatis menggunakan rumus setelah input h	Nilai debit Q muncul otomatis di kolom form sesuai rumus	✓

4.5.4 Pengujian Input Data Bendung

Pengujian input data bendung dapat dilihat pada table 4.4 berikut.

Tabel 4.4 Tabel Pengujian Data Bendung

Fitur yang Diuji	Input	Langkah Pengujian	Output yang Diharapkan	Hasil
Akses Halaman Data Bendung	Pengguna login	Klik menu “Data Bendung” pada sidebar	Halaman bendung tampil dengan daftar bendung	✓
Tambah Data Bendung	Nama STA, nomor, sungai, koordinat, dsb	Isi form lalu klik tombol “Simpan”	Data bendung baru tersimpan dan muncul di daftar	✓
Edit Data Bendung	Perubahan data bendung	Klik tombol “Edit”, ubah data, lalu klik “Simpan”	Data bendung diperbarui dan tampil sesuai perubahan	✓
Hapus Data Bendung	Klik tombol hapus pada satu data	Klik “Hapus” dan konfirmasi penghapusan	Data bendung berhasil dihapus dari daftar	✓

4.5.5 Pengujian Input Data Kemantren

Pengujian input data kemantren dapat dilihat pada table 4.5 berikut.

Tabel 4.5 Tabel Pengujian Data Kemantren

Fitur yang Diuji	Input	Langkah Pengujian	Output yang Diharapkan	Hasil
Akses Halaman Data Kemantren	Pengguna login	Klik menu “Data Kemantren” pada sidebar	Halaman kemantren tampil dengan daftar data	✓
Tambah Data Kemantren	Nama kemantren	Isi form “Tambah Kemantren”, klik tombol “Simpan”	Data kemantren tersimpan dan muncul di daftar	✓

Fitur yang Diuji	Input	Langkah Pengujian	Output yang Diharapkan	Hasil
Edit Data Kemantren	Perubahan nama	Klik tombol “Edit”, ubah nama, lalu klik “Simpan”	Nama kemantren diperbarui dan tampil sesuai input	✓
Hapus Data Kemantren	Klik tombol hapus	Pilih salah satu data dan klik “Hapus”	Data kemantren terhapus dan hilang dari tampilan	✓

4.5.6 Pengujian Input User dan Role

Pengujian input data user dan role dapat dilihat pada table 4.6 berikut.

Tabel 4.6 Tabel Pengujian User dan Role

Fitur yang Diuji	Input	Langkah Pengujian	Output yang Diharapkan	Hasil
Akses Halaman User & Role	Pengguna login sebagai admin	Klik menu “User” atau “Role” pada sidebar	Halaman tampil menampilkan daftar user dan role	✓
Tambah Data User/Role	Isi form nama, jabatan, username, dsb	Klik tombol “Tambah”, isi data, lalu klik “Simpan”	Data user atau role tersimpan dan tampil di daftar	✓
Edit Data User/Role	Ubah nama, jabatan, role, dsb	Klik tombol “Edit”, lakukan perubahan, lalu klik “Simpan”	Data diperbarui sesuai perubahan yang dimasukkan	✓
Hapus Data User/Role	Pilih salah satu data dari daftar	Klik tombol “Hapus” lalu konfirmasi	Data berhasil dihapus dari sistem	✓

4.6 PEMBAHASAN

Bagian pembahasan ini menjelaskan bagaimana hasil perancangan dan implementasi sistem informasi pencatatan debit air berbasis web mampu menjawab permasalahan yang telah dirumuskan sebelumnya. Sistem ini dikembangkan sebagai solusi terhadap pencatatan manual yang masih digunakan oleh petugas lapangan, yang sering kali memunculkan kendala seperti kesalahan pencatatan dan

proses pengelolaan data yang lambat, terutama saat musim hujan atau debit air tidak stabil. Pengembangan dilakukan berdasarkan kebutuhan riil di lapangan yang diperoleh melalui wawancara dan observasi langsung.

Melalui metode pengujian *black box*, sistem terbukti menjalankan seluruh fitur utamanya dengan baik, seperti autentikasi pengguna, pencatatan dan pengelolaan data debit, bendung, kemantren, pengguna, serta peran. Validasi data melalui indikator warna dan fitur visualisasi grafik juga memberikan kemudahan dalam interpretasi informasi dan membantu mengurangi kesalahan input. Arsitektur sistem yang memanfaatkan *Django* dan *MySQL* mendukung pengolahan data secara efisien dan stabil. Hal ini diperkuat dengan hasil wawancara dan tanggapan dari pengguna yang menunjukkan peningkatan efisiensi dan kemudahan dalam proses pencatatan serta penyusunan laporan.

4.6.1 Evaluasi Sebelum Sistem Diterapkan

Dari hasil wawancara dengan petugas lapangan, diketahui bahwa pencatatan debit air dilakukan secara manual menggunakan blangko 0.6 dan 0.8, dengan frekuensi tiga kali sehari pada kondisi normal dan setiap jam saat banjir. Setelah dicatat, data diberikan kepada staf untuk direkap dua minggu sekali sebagai bahan analisis kebutuhan air. Tantangan utama meliputi kondisi cuaca yang menyebabkan kerusakan pada kertas, sedimentasi saluran yang memengaruhi keakuratan data, serta lamanya waktu pencatatan yang dapat mencapai 3–6 jam tergantung banyaknya pintu air yang harus dimonitor. Data disimpan secara gabungan antara bentuk fisik dan digital.

4.6.2 Evaluasi Setelah Sistem Diterapkan

Setelah sistem berbasis web digunakan, dilakukan pengisian survei melalui Google Form. Hasilnya menunjukkan bahwa sistem telah membantu mempercepat proses pencatatan dan penyusunan data, meskipun beberapa data awal tetap harus dicatat manual sebelum diinput. Penggunaan grafik tren debit dan indikator warna dinilai sangat bermanfaat dalam memantau kondisi debit secara harian dan bulanan. Fitur validasi sistem juga memberikan pengaruh dalam menurunkan kesalahan

input. Sebagian besar responden menyatakan sistem sudah sesuai dengan kebutuhan dan belum membutuhkan pengembangan tambahan untuk saat ini.

4.6.3 Relevansi dan Keterbatasan Sistem

Sistem pencatatan debit air berbasis web ini dianggap cocok untuk diterapkan di Dinas PSDA karena mampu menggantikan proses manual dengan sistem digital yang lebih praktis. Kehadiran fitur seperti grafik visual, indikator warna, dan filter data mempermudah pemantauan serta penyusunan laporan.

Namun, sistem masih memiliki kekurangan, seperti belum terhubung langsung dengan alat pengukur debit otomatis dan masih memerlukan pencatatan awal secara manual. Selain itu, keterbatasan jaringan di lapangan juga dapat menghambat input data secara real time. Pengembangan selanjutnya bisa diarahkan pada integrasi alat pengukur dan dukungan akses tanpa internet.

Melalui hasil pengujian dan tanggapan dari pengguna lapangan, dapat disimpulkan bahwa sistem ini berhasil menjawab rumusan masalah penelitian, yakni menyediakan platform pencatatan debit air yang mudah diakses, akurat, dan mendukung pengelolaan data secara efisien.