

BAB 4

HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

Hasil dari penelitian ini berupa pengembangan sistem informasi layanan publik berbasis web bernama SIPRIMA, yang dirancang untuk menggantikan sistem sebelumnya yang masih menggunakan Google Form. Implementasi ini bertujuan meningkatkan efisiensi, akurasi data, dan transparansi layanan di Balai Pengujian Standar Instrumen Tanaman Industri dan Penyegar (BPSI-TRI).

Penelitian menghasilkan aplikasi web yang mampu mengelola berbagai jenis layanan, seperti konsultasi, magang, kunjungan agro eduwisata, perpustakaan, serta fitur galeri dan pengelolaan informasi. Sistem dibangun dengan framework Django dan database *MySQL*, serta mengikuti pendekatan metode prototyping agar dapat divalidasi secara iteratif.

4.2 PENERAPAN DESAIN ANTARMUKA WEB SIPRIMA

Antarmuka pengguna (*User Interface/UI*) merupakan aspek penting dalam pengembangan aplikasi web karena berkaitan langsung dengan pengalaman pengguna dalam mengakses sistem. Dalam pengembangan aplikasi web SIPRIMA, antarmuka dirancang dengan prinsip sederhana, informatif, dan mudah digunakan oleh pemohon maupun petugas (admin). Desain antarmuka disusun berdasarkan hasil analisis kebutuhan dan prototipe awal yang divalidasi bersama pengguna.

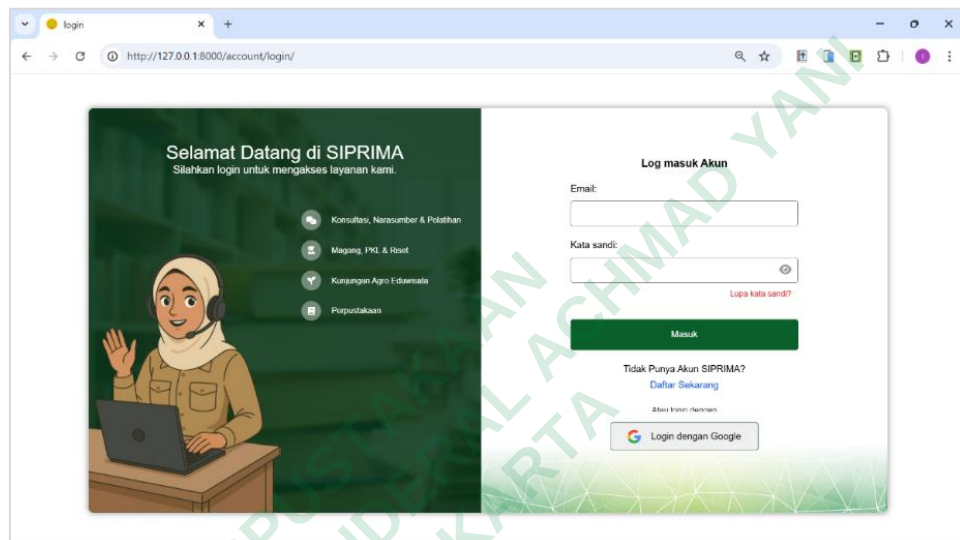
Aplikasi SIPRIMA memiliki dua peran utama, yaitu pemohon dan operator, masing-masing memiliki tampilan dan fungsi antarmuka yang berbeda sesuai hak aksesnya. Berikut ini adalah uraian antarmuka yang telah diimplementasikan.

4.2.1 Halaman Login dan Register

Sistem SIPRIMA menyediakan dua jalur login yang terpisah, yaitu untuk pemohon dan operator (admin). Pemisahan ini bertujuan untuk menjaga keamanan sistem dan memastikan bahwa masing-masing pengguna hanya memiliki akses ke fitur yang sesuai dengan peran dan tanggung jawabnya.

1. Halaman Login Pemohon

Halaman login pemohon merupakan gerbang utama bagi pengguna eksternal untuk mengakses layanan digital yang disediakan oleh sistem SIPRIMA. Desain antarmuka sama dengan rancangan awal, dimana login dibuat sederhana, dan ramah pengguna agar mudah diakses dari berbagai perangkat, termasuk ponsel pintar.



Gambar 4.1 Halaman login pemohon

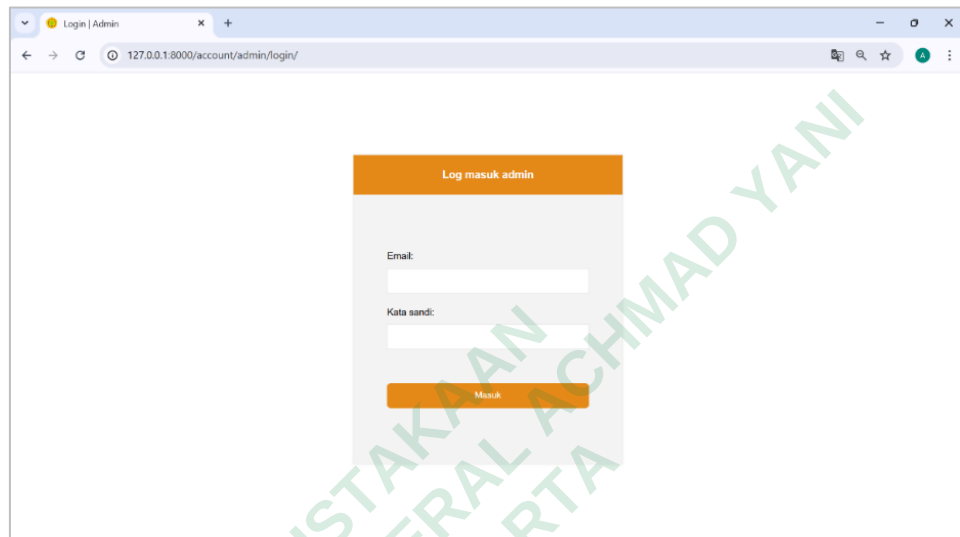
Pada halaman ini, pengguna diminta untuk memasukkan alamat email dan kata sandi yang sudah terdaftar untuk dapat mengakses akun. Selain itu, terdapat opsi login melalui akun Google sebagai alternatif yang lebih praktis.

Fitur utama halaman login pemohon:

- Input email dan kata sandi: Untuk proses autentikasi pengguna.
- Tombol "Masuk": Memproses kredensial yang dimasukkan.
- Tombol "Masuk dengan Google": Mengizinkan pengguna untuk login menggunakan akun Google.
- Link "Daftar Sekarang": Untuk pengguna baru yang belum memiliki akun SIPRIMA.
- Login dengan Google: Alternatif autentikasi berbasis OAuth.

2. Halaman Login Operator/Admin

Halaman login ini diperuntukkan bagi admin atau operator layanan. Desainnya lebih sederhana dan formal. Form login terdiri dari input email dan password, serta tombol untuk masuk ke sistem pengelolaan data layanan.



Gambar 4.2 Halaman login operator

Berikut tampilan potongan kode *backend* yang menangani proses autentikasi pengguna. Potongan kode ini mencerminkan logika pemrosesan data pada sisi server.

```

Def login_view(request, user_type=None):
if request.method == 'POST':
    form = LoginForm(request, data=request.POST)
    if form.is_valid():
        user = form.get_user()

        if user_type == 'admin' and not (user.is_staff or
user.is_superuser):
            form.add_error(None, "Akun ini bukan admin atau
staf.")
        elif user_type == 'user' and (user.is_staff or
user.is_superuser):
            form.add_error(None, "Akun ini khusus untuk
pengguna biasa.")
        else:
            login(request, user)
            if user.is_staff or user.is_superuser:
                return redirect('adminpanel:admin')
            else:

```

```

        return redirect('web:home')
    else:
        form = LoginForm()
    if user_type == 'admin':
        template = 'authusers/login_admin.html'
    else:
        template = 'authusers/login.html'
    return render(request, template, {'form': form})

```

3. Halaman Register

Halaman registrasi pemohon dirancang sebagai antarmuka untuk memungkinkan masyarakat umum mendaftarkan diri sebagai pengguna sistem layanan SIPRIMA. Registrasi ini bertujuan agar pengguna dapat mengakses berbagai layanan yang tersedia seperti konsultasi, magang, kunjungan, dan perpustakaan secara daring.

Gambar 4.3 Halaman register

Antarmuka halaman ini menampilkan form isian dengan struktur yang sederhana dan mudah dipahami, dengan penekanan pada kemudahan dan keakuratan input data oleh pengguna baru.

form registrasi terdiri atas beberapa isian berikut:

1. First Name: Nama depan pengguna.
2. Last Name: Nama belakang pengguna.
3. Email: Alamat email aktif yang akan digunakan untuk login.
4. No Handphone: Nomor kontak pengguna yang valid.

5. Password: Kata sandi yang digunakan untuk login.
6. Ulangi Password: Konfirmasi ulang kata sandi untuk memastikan keakuratan.

Form ini juga telah dilengkapi dengan validasi untuk memastikan seluruh data yang dimasukkan benar dan sesuai format, seperti:

1. Validasi email menggunakan format email standar.
2. Minimal jumlah karakter untuk password.
3. Pengecekan kesamaan antara password dan konfirmasi ulang.

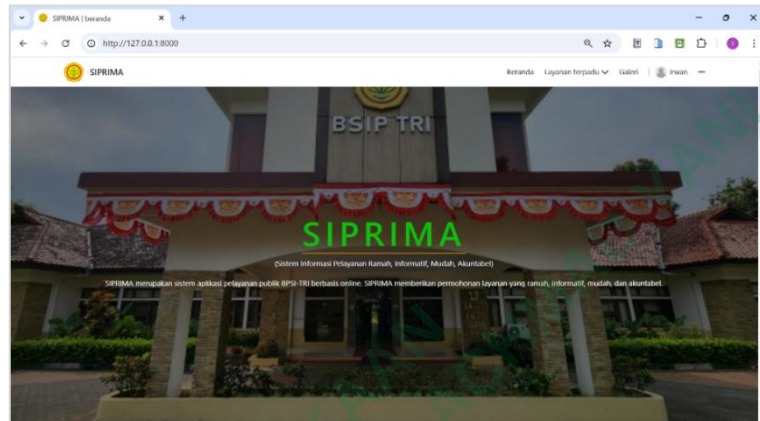
Berikut ini kode pemrograman untuk fungsi pendaftaran akun.

```
def register_view(request):
    if request.method == 'POST':
        form = RegisterForm(request.POST)
        if form.is_valid():
            user = form.save()
            user.backend = get_backends()[0].__module__ + '.' +
get_backends()[0].__class__.__name__
            login(request, user)
            return redirect('web:home')
        else:
            form = RegisterForm()
            return render(request, 'authusers/register.html', {'form':
form})
```

4.2.3 Dashboard Pemohon

1. Section Hero

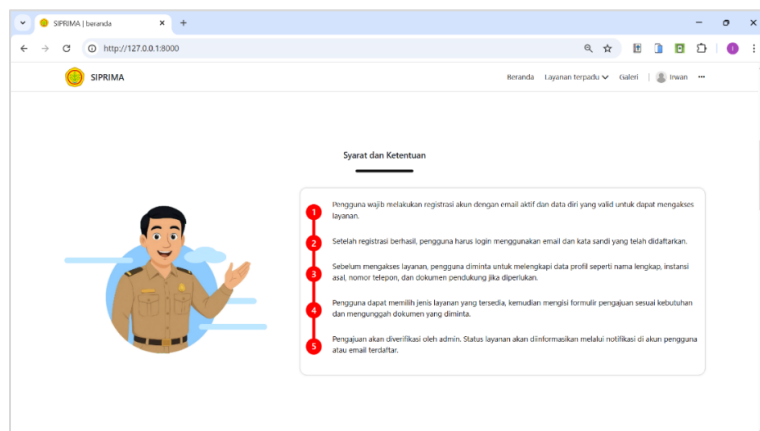
Section hero merupakan bagian awal halaman beranda yang menampilkan nama sistem dan deskripsi singkat layanan SIPRIMA seperti pada Gambar 4.5.



Gambar 4.4 Section hero

2. Section SOP

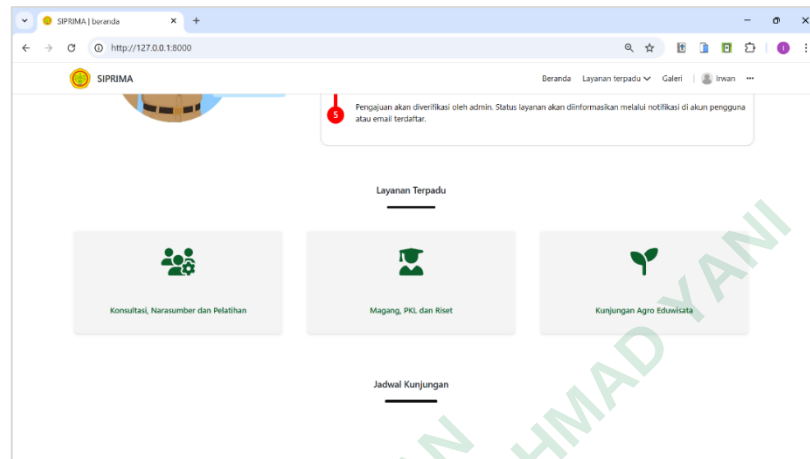
Section ini menampilkan alur prosedur penggunaan layanan dalam bentuk poin-poin vertikal bernomor yang mudah dibaca. Di bagian atas ditampilkan ilustrasi karakter ASN sebagai penguat visual dan simbol formalitas. Desain dibuat responsif agar tetap terbaca dengan baik di perangkat seluler. Tampilan antarmuka section SOP ini ditunjukkan pada Gambar 4.6.



Gambar 4.5 Section SOP

3. Section Layanan Terpadu

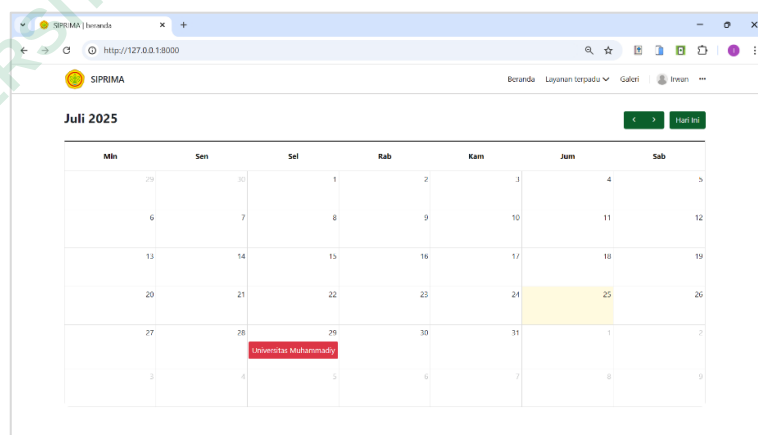
Section ini menampilkan daftar menu layanan yang dapat dipilih pengguna untuk diarahkan ke halaman layanan terkait.



Gambar 4.6 Layanan terpadu

4. Section Jadwal Kunjungan

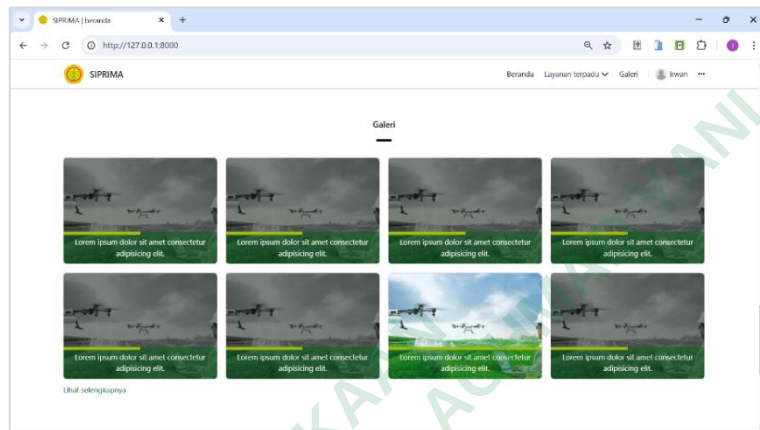
Section ini menampilkan kalender interaktif berbasis plugin *FullCalendar* yang menyajikan informasi tanggal-tanggal pelaksanaan kunjungan. Tampilan kalender dibuat responsif dan disusun secara rapi untuk memudahkan pengguna dalam memantau jadwal yang telah diajukan. Implementasi antarmuka section ini ditunjukkan pada Gambar 4.8.



Gambar 4.7 Section jadwal kunjungan

5. Section Galeri

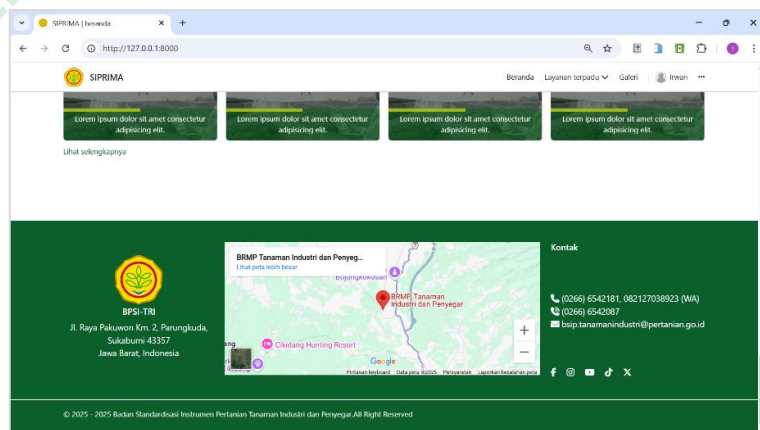
Section ini menampilkan dokumentasi visual dari berbagai kegiatan. Beberapa foto dan video terbaru, disertai tombol “Lihat Selengkapnya” untuk mengarahkan pengguna ke halaman galeri lengkap. Implementasi antarmuka section ini ditunjukkan pada Gambar 4.9.



Gambar 4.8 Section galeri

6. Footer

Bagian footer dirancang sebagai penutup halaman yang memuat informasi penting seperti alamat instansi, peta lokasi Google Maps, kontak, alamat email, serta ikon media sosial. Desainnya menggunakan latar gelap agar kontras dengan isi utama halaman, sekaligus menonjolkan identitas instansi. Implementasi antarmuka ditunjukkan pada Gambar 4.10.

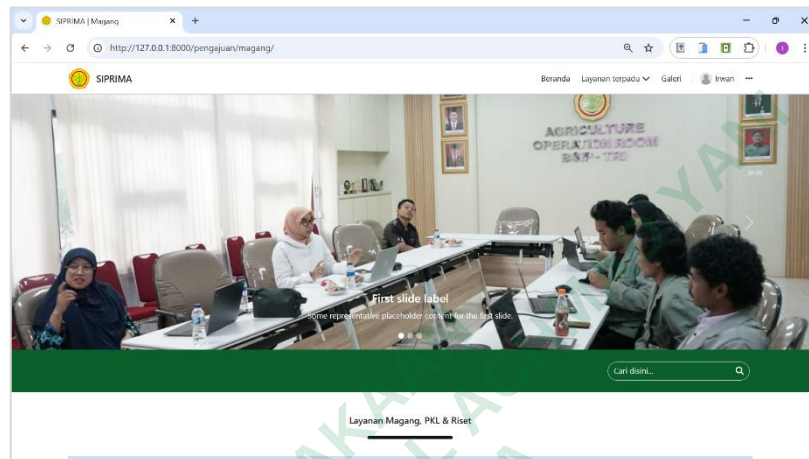


Gambar 4.9 Section footer

4.2.4 Implementasi Pengajuan Layanan

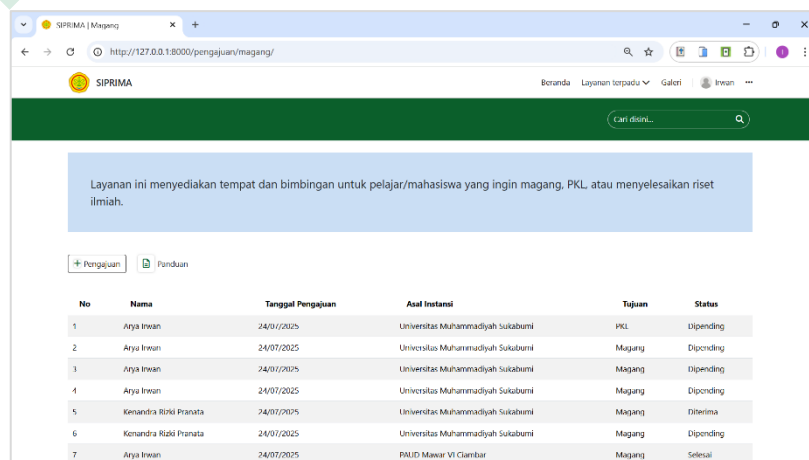
1. Halaman Pengajuan

Section hero pada halaman pengajuan berfungsi sebagai tampilan pembuka yang menampilkan banner gambar (*carousel*), judul layanan, dan kolom pencarian cepat seperti pada Gambar 4.2



Gambar 4.10 Halaman pengajuan

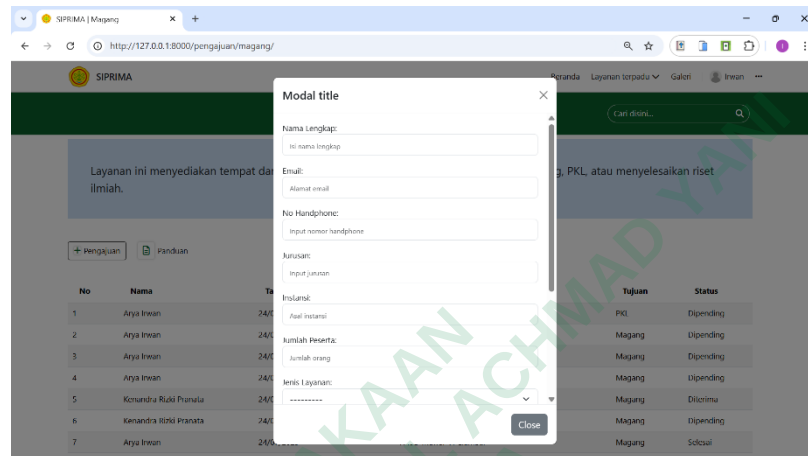
Section data pengajuan menampilkan tabel daftar pengajuan magang yang sudah dilakukan oleh pengguna. Tabel ini mencakup informasi seperti nama, tanggal pengajuan, asal instansi, tujuan, dan status. Di atas tabel, terdapat tombol “Pengajuan” untuk mengisi form permohonan baru dan tombol “Panduan” untuk mengakses petunjuk layanan. Bagian ini berfungsi sebagai pusat pemantauan dan manajemen data pengajuan oleh pengguna.



Gambar 4.11 Daftar pengajuan

2. Form Pengajuan

Form pengajuan ditampilkan dalam modal menggunakan *Bootstrap*. Form ini mencakup input data diri, waktu pelaksanaan, file pendukung dan pemilihan jenis layanan yang ingin diajukan. Antarmuka halaman ini dapat dilihat pada Gambar 4.12.



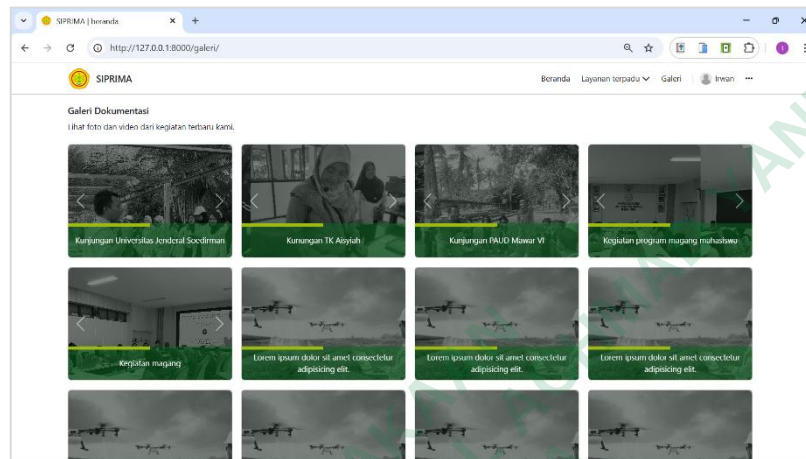
Gambar 4.12 Form pengajuan

Berikut kode untuk menampilkan halaman pengajuan.

```
@login_required
def konsultasi_dashboard(request):
    konsultasi_list = ConsultationService.objects.filter(user =
request.user).order_by('-submission_date')
    if request.method == 'POST':
        form = ConsultationForm(request.POST, request.FILES)
        if form.is_valid():
            konsultasi = form.save(commit=False)
            konsultasi.user = request.user
            konsultasi.save()
            messages.success(request, "Pengajuan berhasil
dikirim!")
            return redirect('konsultasi:konsultasi')
        else:
            print("Form errors:", form.errors)
            messages.error(request, "Terjadi kesalahan saat
mengirim pengajuan.")
    else:
        form = ConsultationForm()
        context = {
            'konsultasi_list': konsultasi_list,
            'form': form,
        }
    return render(request, 'konsultasi/dashboard.html', context)
```

4.2.5 Implementasi Halaman Galeri

Halaman galeri menampilkan dokumentasi berupa foto dan video dari kegiatan SIPRIMA. Setiap item kegiatan ditampilkan dalam bentuk grid, di mana satu kegiatan bisa memuat beberapa foto dalam bentuk slide yang menggunakan *Bootstrap Carousel*.

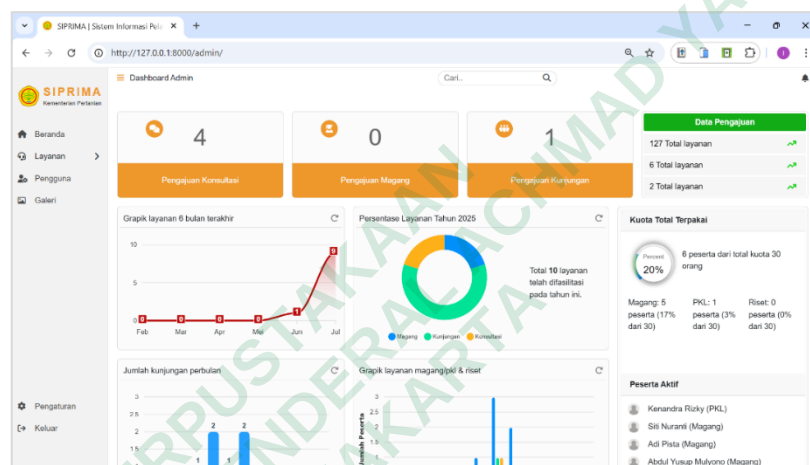


Gambar 4.13 Halaman galeri

4.2.6 Implementasi Halaman Layanan (Admin)

1. Dashboard Admin

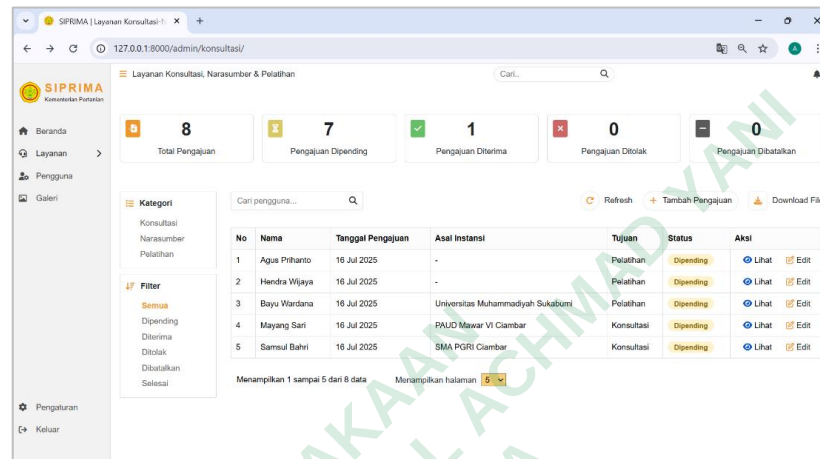
Halaman Dashboard Admin menyajikan ringkasan data pengajuan layanan dalam bentuk card dan grafik. Statistik layanan divisualisasikan menggunakan *Apex Charts*, sedangkan kalender jadwal kunjungan menggunakan FullCalendar. Terdapat juga informasi kuota, jumlah peserta aktif, serta pencarian cepat untuk memudahkan pemantauan layanan oleh admin.



Gambar 4.14 *Dashboard admin*

2. Halaman Pengajuan

Halaman ini digunakan admin untuk mengelola pengajuan layanan, termasuk melihat status (diproses, diterima, ditolak, dibatalkan), mencari nama, memfilter data, menambah pengajuan, melihat detail, dan mengunduh data dalam bentuk file.



Gambar 4.15 Kelola pengajuan

Berikut merupakan Kode yang digunakan untuk menampilkan data pengajuan layanan pada halaman admin.

```
@user_passes_test(lambda u: u.is_staff)
def index_consultation(request):
    konsultasi = ConsultationService.objects.all().order_by('-submission_date')
    total_konsultasi = konsultasi.count()
    konsultasi_dipending = konsultasi.filter(status='Dipending').count()
    konsultasi_diterima = konsultasi.filter(status='Diterima').count()
    konsultasi_ditolak = konsultasi.filter(status='Ditolak').count()
    konsultasi_dibatalkan = konsultasi.filter(status='Dibatalkan').count()
    konsultasi_selesai = konsultasi.filter(status='Selesai').count()
    context = {
        'total_konsultasi': total_konsultasi,
        'konsultasi_dipending': konsultasi_dipending,
        'konsultasi_diterima': konsultasi_diterima,
        'konsultasi_ditolak': konsultasi_ditolak,
        'konsultasi_dibatalkan': konsultasi_dibatalkan,
        'konsultasi_selesai': konsultasi_selesai,
        'konsultasi': konsultasi,
    }
```

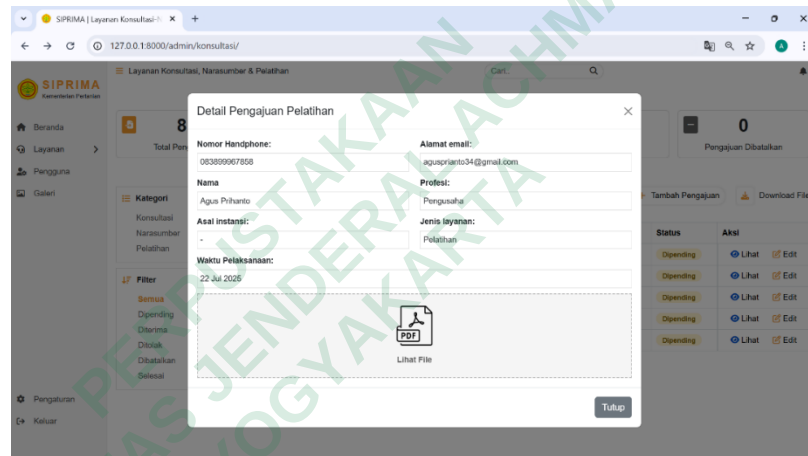
```

        'page_title': 'Layanan Konsultasi, Narasumber &
        Pelatihan'
    }
    return render(request, 'services/consultations/index.html',
    context)

```

3. Lihat data pengajuan

Tampilan ini menampilkan detail data pengajuan layanan dalam bentuk pop-up menggunakan *Bootstrap* modal, sehingga admin dapat melihat informasi lengkap seperti data diri, jenis layanan, waktu pelaksanaan, dan dokumen yang dilampirkan tanpa perlu berpindah halaman.



Gambar 4.16 Data pengajuan

4. Edit data pengajuan

Halaman ini memungkinkan admin untuk mengubah data pengajuan layanan, memilih jenis layanan yang sesuai, serta menetapkan status pengajuan seperti Disetujui, Ditolak, atau Menunggu. Admin memiliki kontrol penuh untuk memperbaiki atau memverifikasi informasi yang telah diajukan oleh pengguna.

The screenshot displays the 'Edit Pengajuan Konsultasi' interface. On the left, a sidebar lists navigation options: Beranda, Layanan, Pengguna, and Galeri. The main content area is split into two panels. The 'Validasi Data' panel features a 'Surat Pengajuan' section with a PDF icon and a 'Lihat File' link, a 'Status' dropdown menu set to 'Pending', and an orange 'Simpan' button. The 'Data Pengajuan' panel contains a form with the following fields: 'Nama Lengkap' (Agus Prihanto), 'Email' (agusprianto34@gmail.com), 'No HP' (083899967858), 'Profesi' (Pengusaha), 'Asal Instansi' (empty), 'Jenis Layanan' (empty), 'Pelatihan' (empty), and 'Waktu Pelaksanaan' (22/07/2025).

Gambar 4.17 Edit data pengajuan

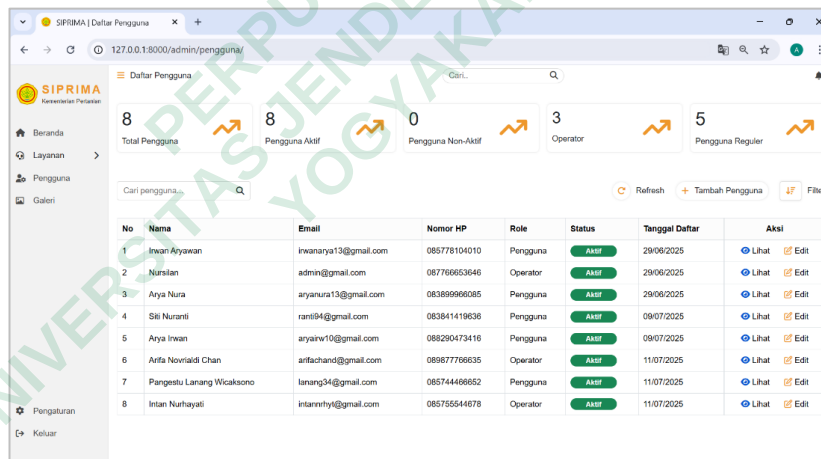
4.2.7 Implementasi Manajemen Akun Pengguna (Admin)

1. Halaman Utama Akun Pengguna

Halaman Akun Pengguna digunakan oleh admin untuk melihat dan mengelola seluruh data akun yang terdaftar dalam sistem. Pada halaman ini, admin dapat melihat jumlah total pengguna, jumlah pengguna aktif dan nonaktif, serta jumlah pengguna berdasarkan peran seperti operator dan pengguna regular.

Admin juga dapat mencari pengguna melalui kolom pencarian, memfilter data, serta memperbarui tampilan data dengan tombol *refresh*. Tersedia tabel yang menampilkan informasi inti dari setiap pengguna seperti nama, email, nomor HP, peran, status, dan tanggal pendaftaran.

Untuk setiap pengguna, admin dapat memilih tindakan seperti Lihat untuk melihat detail data, atau Edit untuk mengubah informasi pengguna. Selain itu, tersedia tombol Tambah Pengguna untuk membuat akun baru langsung dari halaman ini.



The screenshot shows the SIPRIMA Admin Dashboard. At the top, there are statistics for Total Pengguna (8), Pengguna Aktif (8), Pengguna Non-Aktif (0), Operator (3), and Pengguna Regular (5). Below these are search, refresh, and add user buttons. A table lists 8 users with columns for No, Nama, Email, Nomor HP, Role, Status, Tanggal Daftar, and Aksi. The table data is as follows:

No	Nama	Email	Nomor HP	Role	Status	Tanggal Daftar	Aksi
1	Irwani Ariyawan	irwanarya13@gmail.com	085778104010	Pengguna	Aktif	29/06/2025	Lihat Edit
2	Nursilan	admirn@gmail.com	08776653646	Operator	Aktif	29/06/2025	Lihat Edit
3	Arya Nura	aryanura13@gmail.com	08389966085	Pengguna	Aktif	29/06/2025	Lihat Edit
4	Siti Nuranti	ranti94@gmail.com	083841419636	Pengguna	Aktif	09/07/2025	Lihat Edit
5	Arya Irwan	aryairw10@gmail.com	089290473416	Pengguna	Aktif	09/07/2025	Lihat Edit
6	Arifa Novriadi Chan	arifachand@gmail.com	089877766635	Operator	Aktif	11/07/2025	Lihat Edit
7	Pangestu Lanang Wicaksono	lanang34@gmail.com	085744466652	Pengguna	Aktif	11/07/2025	Lihat Edit
8	Irtan Nurhayati	irtannrhy@gmail.com	085755644678	Operator	Aktif	11/07/2025	Lihat Edit

Gambar 4.18 Kelola pengguna

Berikut kode pemrograman untuk mengelola akun pengguna

```
@@user_passes_test(lambda u: u.is_staff)
def list_user(request):
    users = CustomUser.objects.all()
    total_users = users.count()
    active_users = users.filter(is_active=True).count()
    inactive_users = users.filter(is_active=False).count()
    operator_users = users.filter(is_staff=True).count()
    regular_users = users.filter(is_staff=False,
```

```

is_superuser=False).count()
    context = {
        'users': users,
        'total_users': total_users,
        'active_users': active_users,
        'inactive_users': inactive_users,
        'operator_users': operator_users,
        'regular_users': regular_users,
        'page_title': 'Daftar Pengguna'
    }
    return render(request, 'users/user.html', context)

```

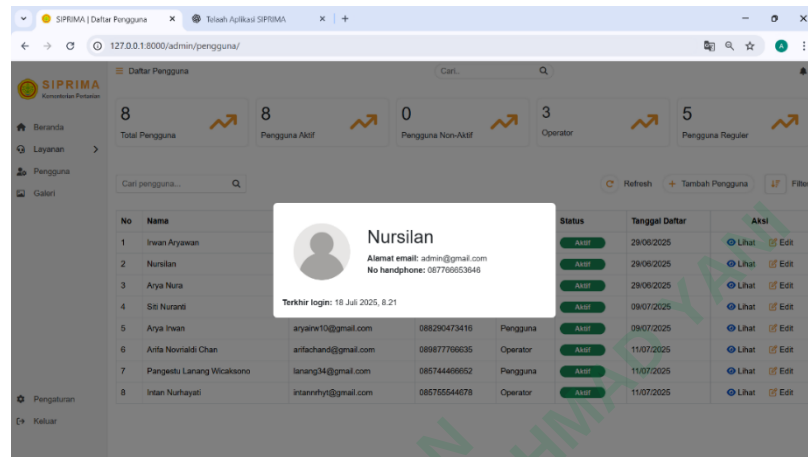
2. Tambah Pengguna Baru

Formulir ditampilkan dalam bentuk pop-up untuk memudahkan admin menambahkan pengguna tanpa harus berpindah halaman, form ini terdiri dari data diri dan peran pengguna

Gambar 4.19 Form tambah pengguna

3. Lihat Akun Pengguna

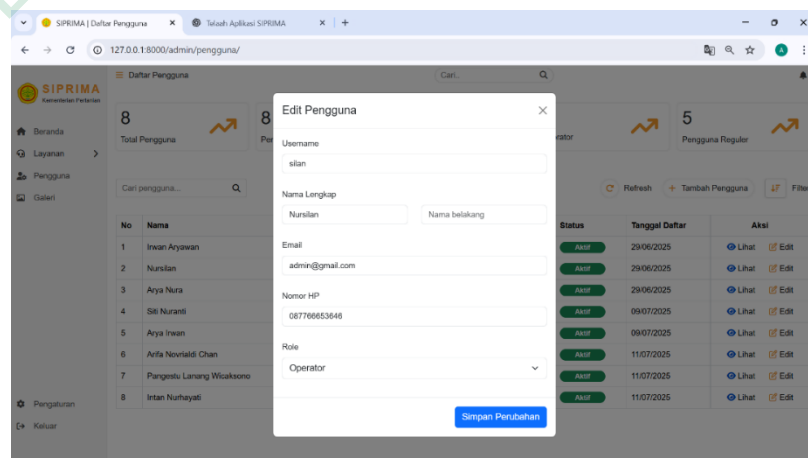
Informasi lengkap pengguna ditampilkan dalam pop-up agar admin dapat dengan cepat melihat data tanpa meninggalkan halaman utama



Gambar 4.20 Detail data pengguna

4. Edit Akun Pengguna

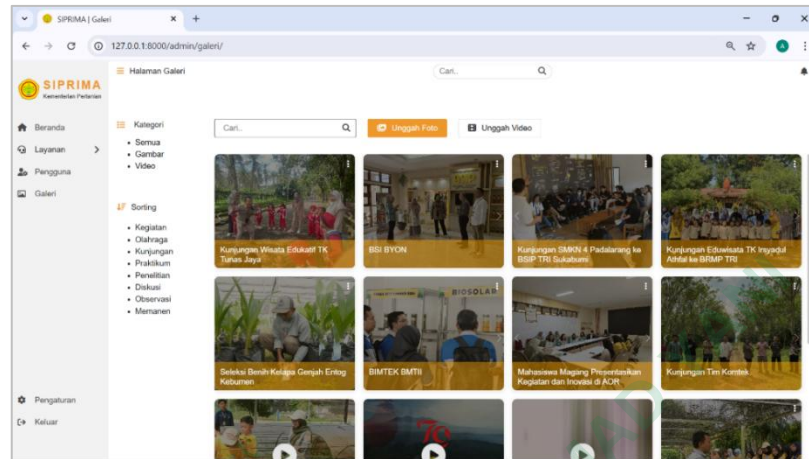
Admin dapat melakukan pengeditan data pengguna melalui tampilan pop-up yang interaktif. Dalam pop-up ini, admin bisa memperbarui data diri pengguna, email, nomor telepon, serta mengubah peran (role) pengguna seperti Admin, Operator, atau Pengguna. Selain itu, admin juga dapat mengatur status akun aktif atau tidak aktif sesuai kebutuhan



Gambar 4.21 Edit akun

4.2.8 Implementasi Kelola Galeri

1. Halaman Utama Galeri



Gambar 4.22 Galeri admin

Halaman utama galeri menampilkan daftar konten berupa gambar dan video dalam bentuk grid. Admin dapat memfilter berdasarkan kategori dan jenis kegiatan di panel kiri, serta melakukan pencarian melalui kolom pencarian. Tersedia tombol “Unggah Foto” dan “Unggah Video” untuk menambahkan konten baru. Setiap item galeri dilengkapi dengan pratinjau, judul, dan menu tindakan untuk melihat, mengedit, atau menghapus. Tampilan ini memudahkan admin dalam mengelola dokumentasi kegiatan secara menyeluruh.

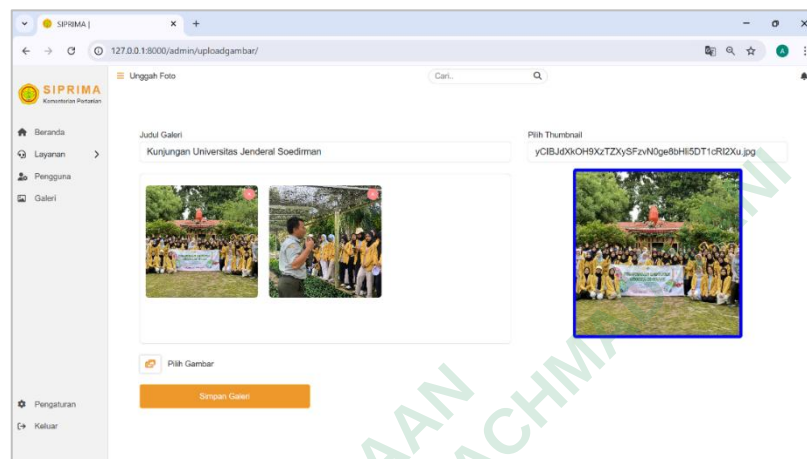
Berikut kode untuk tampilan utama admin

```
@user_passes_test(lambda u: u.is_staff)
def gallery(request):
    galleries = Gallery.objects.prefetch_related('images',
    'videos').all()
    context = {
        'galleries': galleries,
        'page_title': 'Halaman Galeri',
    }
    return render(request, 'gallery/gallery.html', context)
```

2. Tambah Galeri Gambar

Halaman tambah galeri memungkinkan admin mengunggah beberapa gambar sekaligus, menambahkan judul galeri, dan memilih salah satu gambar sebagai thumbnail utama. Gambar-gambar yang diunggah

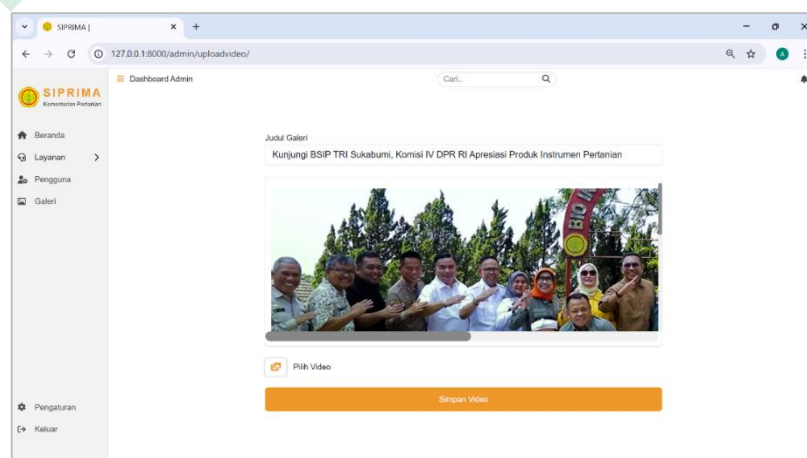
langsung ditampilkan dalam bentuk pratinjau grid. Gambar pertama secara otomatis dijadikan thumbnail, namun admin bisa mengubahnya. Setelah semua data lengkap, admin dapat menekan tombol "Simpan Galeri" untuk menyimpan galeri ke sistem.



Gambar 4.23 Tambah gambar

3. Tambah Galeri Video

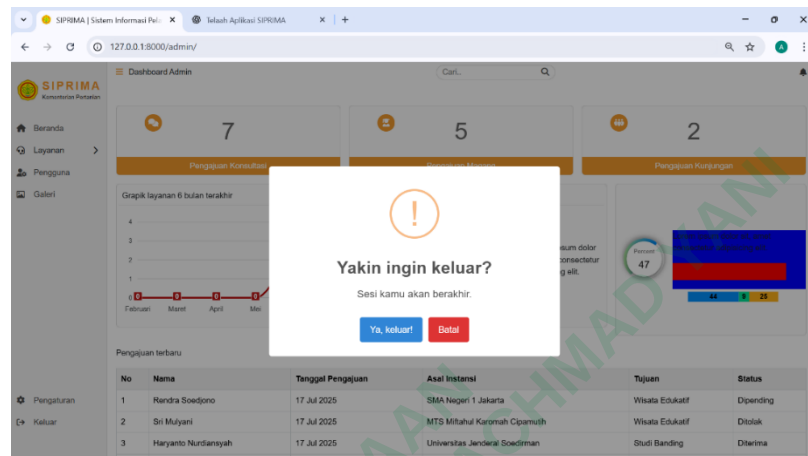
Halaman tambah galeri video memiliki tampilan dan fungsi yang mirip dengan halaman tambah galeri gambar. Admin dapat mengunggah video dan memberikan judul galeri. Setiap video yang dipilih langsung ditampilkan dalam bentuk pratinjau. Setelah proses unggah selesai dan data lengkap, admin dapat menyimpan galeri dengan menekan tombol "Simpan Galeri".



Gambar 4.24 Tambah video

4. Implementasi menu keluar

Menu Keluar berfungsi untuk mengakhiri session dan keluar dari akun. Sebelum benar-benar keluar, sistem akan menampilkan pop-up untuk memastikan kamu yakin ingin meninggalkan halaman ini.



Gambar 4.25 Pesan konfirmasi keluar

4.3 STRUKTUR DAN IMPLEMENTASI BASIS DATA

Implementasi basis data pada sistem SIPRIMA disusun berdasarkan desain yang telah dirancang sebelumnya pada bagian 3.3.2 Perancangan Basis Data. Perancangan database sistem dilakukan untuk merepresentasikan struktur data layanan secara menyeluruh dan relasional. Implementasi dilakukan menggunakan Django ORM (*Object-Relational Mapping*), yang memungkinkan pemetaan langsung antara model Python dan tabel dalam database MySQL secara otomatis dan efisien.

Struktur basis data yang dibangun merefleksikan entitas layanan yang ada, seperti pengguna, konsultasi, magang, kunjungan, dan galeri. Setiap entitas direpresentasikan sebagai kelas model yang didefinisikan dalam file `models.py`, dengan atribut dan relasi yang disesuaikan berdasarkan kebutuhan sistem.

```
class CustomUser (AbstractUser):
    email = models.EmailField(unique=True)
    phone_number = models.CharField(max_length=15, blank=True,
null=True)
    USERNAME_FIELD = 'email'
    REQUIRED_FIELDS = ['username']

    photo = models.ImageField(
        upload_to='user_photos/',
        blank=True,
        null=True,
    )

    @property
    def role(self):
        if self.is_superuser:
            return "Operator"
        elif self.is_staff:
            return "Admin"
        else:
            return "User"

    def __str__(self):
        return self.username

class ConsultationService(models.Model):
    user = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
    full_name = models.CharField(max_length=250)
    email = models.EmailField(max_length=254)
    phone_number = models.CharField(max_length=50)
    profession = models.CharField(max_length=250)
```

```

        institution = models.CharField(max_length=250)
        service_type = models.CharField(max_length=50,
choices=SERVICE_TYPE)
        implementation_date = models.DateField()
        document = models.FileField(upload_to='document/',
max_length=500)
        submission_date = models.DateTimeField(auto_now_add=True)
        status = models.CharField(max_length=50, choices=STATUS,
default='Dipending')

    def __str__(self):
        return self.full_name

class Gallery(models.Model):
    user = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
    title = models.CharField(max_length=255)
    thumbnail = models.ImageField(upload_to='gallery/thumbnail/',
null=True, blank=True)
    is_archived = models.BooleanField(default=False)
    created_at = models.DateTimeField(auto_now_add=True)
    def __str__(self):
        return self.title

class GalleryImage(models.Model):
    gallery = models.ForeignKey(Gallery,
on_delete=models.CASCADE, related_name='images')
    image = models.ImageField(upload_to='gallery/images/')
    uploaded_at = models.DateTimeField(auto_now_add=True)

class GalleryVideo(models.Model):
    gallery = models.ForeignKey(Gallery,
on_delete=models.CASCADE, related_name='videos')
    video = models.FileField(upload_to='gallery/videos/')
    uploaded_at = models.DateTimeField(auto_now_add=True)

```

4.4 PENGUJIAN SISTEM

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fungsi pada sistem informasi layanan SIPRIMA berjalan sesuai dengan yang dirancang. Pengujian ini bertujuan untuk mengidentifikasi kesalahan (error), mengevaluasi kinerja fitur, serta memastikan kualitas dan keandalan sistem sebelum digunakan oleh pengguna sebenarnya.

4.4.1 BlackBox Testing

Pengujian sistem dilakukan dengan metode *Black Box Testing* untuk memverifikasi bahwa setiap fitur pada sistem SIPRIMA berjalan sesuai dengan

kebutuhan fungsional. Pengujian ini berfokus pada masukan (input) dan keluaran (output) dari sistem, tanpa memperhatikan proses internal kode program.

Berikut adalah tabel pengujian *black box* untuk beberapa fitur utama sistem:

Tabel 4.1 Black box testing

Fitur yang di uji	Skenario pengujian	Data Uji	Hasil yang Diharapkan	Status
Login Pengguna	Pengguna login dengan data yang valid	Email: admin@mail.com Password: admin123	Berhasil masuk ke halaman utama pengguna	Sesuai
Login Pengguna	Login dengan email yang salah	Email: adam@mail.com Password: admin123	Muncul pesan kesalahan: Masukkan nama pengguna email dan sandi yang benar. Huruf besar/kecil pada bidang ini berpengaruh.	Sesuai
Register Pengguna Baru	Mendaftar akun baru dengan data lengkap	Data lengkap valid	Registrasi berhasil dan akun bisa login	Sesuai
Register Pengguna Baru	Mendaftar akun dengan email yang sudah terdaftar	Email: admin@gmail.com	Muncul pesan: "Pengguna dengan email telah ada"	Sesuai
Form Pengajuan Layanan	Mengirim form dengan seluruh data yang valid	Semua field diisi dan file diunggah	Form berhasil dikirim dan muncul notifikasi sukses	Sesuai
Form Pengajuan Layanan	Mengirim form tanpa mengisi salah satu input wajib	Nama lengkap kosong	Muncul pesan kesalahan pada field kosong	
Kalender Kunjungan	Menampilkan jadwal kunjungan	-		Sesuai

Fitur yang di uji	Skenario pengujian	Data Uji	Hasil yang Diharapkan	Status
	yang sudah disetujui		Jadwal tampil di kalender sesuai tanggalnya	
Admin Verifikasi Layanan	Admin mengubah status pengajuan menjadi “Diterima”	Status: Diterima	Status berubah dan tampil di dashboard	Sesuai
Tampilan Galeri Publik	Menampilkan dokumentasi terbaru secara slide	-	Galeri tampil dalam format carousel / grid	Sesuai

4.4.2 Uji Coba Langsung (User Acceptance Test/UAT)

Pengujian *User Acceptance Test* (UAT) dilakukan untuk mengevaluasi sejauh mana sistem yang telah dibangun dapat diterima dan digunakan oleh pengguna sesuai kebutuhan dan tujuan layanan. Uji coba ini melibatkan pengguna akhir, yaitu pemohon layanan dan admin/operator dari BPSI-TRI, untuk mengoperasikan langsung sistem SIPRIMA.

Uji coba dilakukan dengan memberikan sejumlah skenario penggunaan yang merepresentasikan aktivitas sebenarnya.

1. Pengujian Fungsi Administrator Sistem

Tabel 4.2 Pengujian sistem oleh operator

Skenario Pengujian	Deskripsi Tindakan	Hasil yang Diharapkan	Status
Login sebagai operator	Operator/admin masuk menggunakan akun admin yang valid	Berhasil masuk ke dashboard admin	Sesuai
Verifikasi Pengajuan	Admin mengakses daftar pengajuan dan mengubah status menjadi diterima/ditolak	Status pengajuan berubah dan notifikasi dikirim ke pemohon	Seesuai

Skenario Pengujian	Deskripsi Tindakan	Hasil yang Diharapkan	Status
Kelola Pengguna	Admin menambah/mengedit/menghapus data pengguna dari panel kelola	Perubahan data tersimpan dan ditampilkan dengan benar	Sesuai
Lihat Statistik	Admin melihat grafik jumlah pengajuan layanan bulanan	Grafik tampil sesuai data aktual	Sesuai

2. Pengujian Sistem Oleh Pemohon

Tabel 4.3 Pengujian sistem oleh pemohon

Skenario Pengujian	Deskripsi Tindakan	Hasil yang Diharapkan	Status
Login Pengguna	Pengguna memasukkan email dan kata sandi yang valid	Sistem menampilkan dashboard pengguna	Sesuai
Pengajuan Layanan	Pengguna mengisi dan mengirim formulir pengajuan	Notifikasi pengajuan berhasil dan data tersimpan	Sesuai
Melihat Status Pengajuan	Pengguna mengecek status dari dashboard	Status ditampilkan dengan benar	Sesuai
Mengunduh Surat Balasan	Pengguna mengklik tombol unduh pada surat balasan yang telah disetujui	File berhasil diunduh tanpa kendala	Sesuai

4.5 PEMBAHASAN

Bab ini membahas hasil implementasi dan pengujian sistem berdasarkan pertanyaan penelitian yang telah dirumuskan sebelumnya. Pembahasan difokuskan pada efektivitas sistem SIPRIMA sebagai layanan publik berbasis web, perancangan basis data untuk mendukung proses layanan, pendekatan pengembangan menggunakan metode prototyping, serta pengujian sistem untuk memastikan fungsionalitas dan penerimaan pengguna.

4.5.1 Efektivitas Antarmuka Sistem SIPRIMA dalam Mendukung Layanan Publik Berbasis Web

Sistem SIPRIMA dirancang sebagai aplikasi web yang dapat diakses secara daring oleh masyarakat umum dan admin layanan. Antarmuka pengguna dikembangkan dengan prinsip usability, responsif, dan mudah dipahami. Melalui tampilan yang terstruktur dan alur yang jelas, pengguna dapat dengan mudah:

1. Melakukan pendaftaran dan login (termasuk login menggunakan akun Google).
2. Mengisi dan mengirim pengajuan layanan seperti konsultasi, magang, kunjungan, atau perpustakaan.
3. Melihat status pengajuan secara real-time di dashboard pengguna.
4. Mengakses informasi tambahan seperti syarat layanan, galeri, dan jadwal kunjungan.

4.5.2 Perancangan Basis Data Relasional dalam Mendukung Layanan Terintegrasi

Struktur basis data SIPRIMA disusun secara relasional untuk memastikan integrasi data yang baik antara berbagai entitas layanan. Model ini mendukung pengelolaan data seperti:

1. Informasi akun pengguna dan profil.
2. Data pengajuan layanan beserta dokumen yang diunggah.
3. Riwayat status dan tindak lanjut oleh admin.
4. Galeri dokumentasi kegiatan.
5. Jadwal kunjungan yang ditampilkan melalui kalender interaktif.

4.5.3 Penerapan Metode Prototyping dalam Pengembangan Sistem

Metode prototyping digunakan dalam pengembangan sistem untuk memastikan kesesuaian antara kebutuhan pengguna dan fitur sistem. Prototipe awal dirancang menggunakan Figma sebagai alat bantu visual, lalu divalidasi melalui masukan dari pengguna instansi.

Melalui pendekatan iteratif ini, perubahan dapat dilakukan lebih cepat, seperti penyesuaian tampilan form pengajuan, pengelompokan menu layanan, dan konfirmasi visual setelah pengajuan dikirim. Keterlibatan pengguna secara langsung mempercepat proses validasi desain dan menjadikan sistem lebih sesuai dengan kebutuhan operasional BPSI-TRI.

4.5.4 Pengujian Sistem untuk Menjamin Fungsionalitas dan Penerimaan Pengguna

Pengujian dilakukan melalui metode Black Box Testing dan User Acceptance Test (UAT). Hasil pengujian menunjukkan bahwa sistem berjalan sesuai fungsi, di antaranya:

1. Formulir dapat disubmit dengan benar dan tersimpan di basis data.
2. Verifikasi dan perubahan status layanan oleh admin berfungsi normal.
3. Dashboard menampilkan statistik layanan, kalender kunjungan, dan pengajuan terbaru secara real-time.
4. UAT yang melibatkan pengguna dan admin menunjukkan bahwa sistem mudah digunakan, fitur dapat diakses dengan baik, dan sesuai kebutuhan operasional.

Pengujian ini menunjukkan bahwa sistem SIPRIMA tidak hanya berhasil dikembangkan secara teknis, tetapi juga diterima dengan baik oleh penggunanya di lingkungan BPSI-TRI