**LAPORAN**

**“APLIKASI UNIDAYAN”**

****

**Oleh :**

**OKI JAYA GUNA : 22650161**

**GAVID RIZAL : 22650172**

**MUHAMMAD ASFAR DANI : 22650168**

**WA ODE BALQIS SELVIA : 22650170**

**MARINI YUNING TYASMAR : 22650171**

**IIS PRATAMI : 22650173**

**ARFAN HASMU : 22650175**

**KASRIN JAYA : 22650176**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS DAYANU IKHSANUDDIN**

**PASARWAJO**

**2023**

1. **ABSTRAK**

Dalam era digital saat ini, pengelolaan jadwal pelajaran di institusi pendidikan menuntut solusi yang lebih efisien dan otomatis. Oleh karena itu, kami mengembangkan sebuah aplikasi manajemen jadwal pelajaran yang bertujuan untuk memodernisasi dan mempermudah proses pengelolaan daftar dosen, jadwal pelajaran, serta data mahasiswa. Aplikasi ini dirancang menggunakan framework Django untuk backend dan HTML, CSS, serta JavaScript untuk frontend, menawarkan antarmuka yang user-friendly dan responsif.

Laporan ini mendokumentasikan seluruh proses pengembangan aplikasi, mulai dari perancangan awal hingga pengujian akhir. Melalui penggunaan diagram use case, class, dan aktivitas, kami berhasil mendefinisikan kebutuhan fungsional aplikasi dan alur kerja sistem. Implementasi aplikasi diuji melalui serangkaian pengujian unit, integrasi, dan penerimaan pengguna, yang semuanya memastikan bahwa aplikasi berjalan sesuai dengan kebutuhan yang diharapkan.

Dengan aplikasi ini, diharapkan institusi pendidikan dapat meningkatkan efisiensi dalam pengelolaan jadwal pelajaran, serta memberikan akses yang mudah dan cepat bagi dosen dan mahasiswa untuk mendapatkan informasi jadwal pelajaran terkini. Selain itu, aplikasi juga menawarkan kemungkinan pengembangan lebih lanjut, termasuk integrasi dengan sistem lain dan penambahan fitur-fitur baru untuk mendukung kebutuhan pengguna.

1. **PENDAHULUAN**
2. **Latar Belakang**

Pengelolaan jadwal pelajaran merupakan salah satu aspek penting dalam institusi pendidikan. Proses ini mencakup penjadwalan mata pelajaran, pengelolaan daftar dosen, serta pengaturan data mahasiswa. Dengan semakin berkembangnya teknologi informasi, penggunaan sistem manual dirasa kurang efektif dan efisien. Oleh karena itu, diperlukan sebuah aplikasi yang dapat membantu dalam manajemen jadwal pelajaran secara otomatis dan terstruktur.

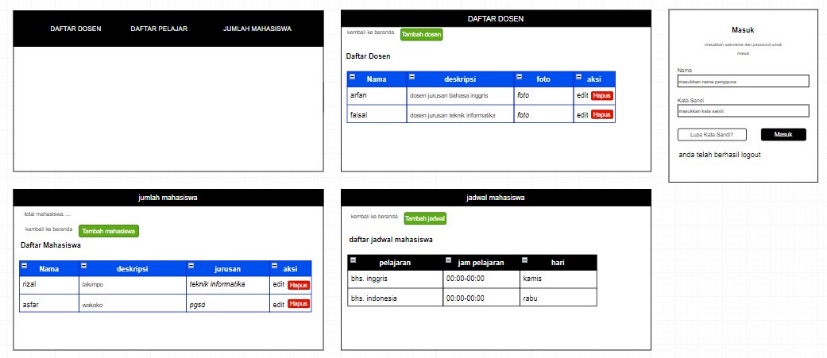
1. **Tujuan**

Laporan ini bertujuan untuk mendokumentasikan proses pengembangan aplikasi manajemen jadwal pelajaran. Aplikasi ini dirancang untuk memudahkan admin dalam mengelola daftar dosen, jadwal pelajaran, dan data mahasiswa, serta memberikan kemudahan bagi pengguna untuk melihat jadwal pelajaran yang telah ditentukan. Proses pengembangan melibatkan beberapa tahapan penting, yaitu perancangan sistem, implementasi, dan pengujian, yang kesemuanya akan dijelaskan secara rinci dalam laporan ini.

1. **Manfaat**

Dengan adanya aplikasi ini, diharapkan dapat meningkatkan efisiensi dan efektivitas dalam pengelolaan jadwal pelajaran di institusi pendidikan, serta memberikan pengalaman yang lebih baik bagi pengguna. Laporan ini juga akan membahas berbagai diagram yang digunakan dalam perancangan, detail implementasi, serta hasil pengujian untuk memastikan aplikasi berfungsi sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan.

1. **ANALISIS KEBUTUHAN**
2. **WIREFRAME**



Wireframe adalah representasi visual dari struktur dan tata letak aplikasi. Dalam proyek ini, wireframe digunakan untuk merancang antarmuka pengguna (UI) dari aplikasi manajemen jadwal pelajaran. Wireframe membantu memvisualisasikan elemen-elemen penting dari halaman-halaman aplikasi sebelum tahap implementasi dimulai. Berikut adalah penjelasan detail tentang wireframe untuk aplikasi ini:

1. **Halaman Login**

Deskripsi: Halaman login digunakan oleh pengguna untuk mengakses sistem dengan memasukkan kredensial mereka.

1. Elemen Utama:

* Field Username: Tempat untuk memasukkan username.
* Field Password: Tempat untuk memasukkan password.
* Button Login: Tombol untuk mengirimkan kredensial dan masuk ke sistem.
* Link Lupa Password: Opsi untuk pengguna yang lupa password.

1. Alur Kerja:

* Pengguna memasukkan username dan password, lalu menekan tombol login untuk masuk ke dashboard.

1. **Dashboard Admin**

Deskripsi: Halaman utama setelah login untuk admin, yang berfungsi sebagai pusat navigasi untuk mengelola data.

1. Elemen Utama:

* Menu Navigasi: Tautan ke halaman-halaman lain seperti kelola dosen, kelola jadwal, dan kelola mahasiswa.
* Ringkasan Data: Informasi ringkas mengenai jumlah dosen, jumlah mata pelajaran, dan jumlah mahasiswa.
* Tabel Data: Menampilkan data-data terbaru atau penting yang perlu diperhatikan.

1. Alur Kerja:

* Admin dapat mengakses berbagai halaman pengelolaan dari menu navigasi dan melihat ringkasan data penting di dashboard.

1. **Halaman Kelola Daftar Dosen**

Deskripsi: Halaman ini memungkinkan admin untuk menambah, mengubah, dan menghapus data dosen.

1. Elemen Utama:

* Form Tambah/Ubah Dosen: Formulir untuk memasukkan atau mengubah informasi dosen.
* Tabel Dosen: Menampilkan daftar dosen yang sudah ada, dengan opsi untuk mengedit atau menghapus masing-masing dosen.
* Button Simpan: Tombol untuk menyimpan data dosen yang baru atau yang sudah diubah.

1. Alur Kerja:

* Admin dapat menambah dosen baru dengan mengisi formulir dan menekan tombol simpan, atau mengedit/hapus data dosen yang sudah ada dari tabel.

1. **Halaman Kelola Jadwal Pelajaran**

Deskripsi: Halaman ini digunakan oleh admin untuk mengelola jadwal pelajaran.

1. Elemen Utama:

* Form Tambah/Ubah Jadwal: Formulir untuk memasukkan atau mengubah informasi jadwal pelajaran.
* Tabel Jadwal: Menampilkan daftar jadwal pelajaran yang sudah ada, dengan opsi untuk mengedit atau menghapus masing-masing jadwal.
* Button Simpan: Tombol untuk menyimpan data jadwal yang baru atau yang sudah diubah.

1. Alur Kerja:

* Admin dapat menambah jadwal pelajaran baru dengan mengisi formulir dan menekan tombol simpan, atau mengedit/hapus data jadwal yang sudah ada dari tabel.

1. **Halaman Lihat Jadwal Pelajaran**

Deskripsi: Halaman ini memungkinkan pengguna untuk melihat jadwal pelajaran yang tersedia.

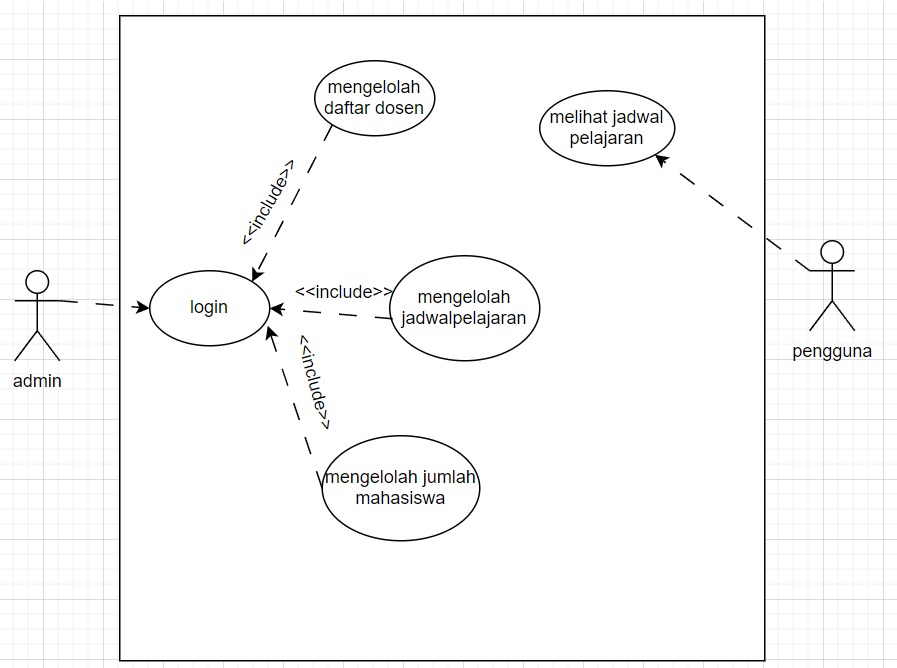
1. Elemen Utama:

* Filter Pencarian: Opsi untuk memfilter jadwal berdasarkan mata pelajaran, dosen, atau hari.
* Tabel Jadwal: Menampilkan jadwal pelajaran sesuai dengan filter yang dipilih.

1. Alur Kerja:

* Pengguna dapat memfilter jadwal pelajaran berdasarkan kriteria yang diinginkan dan melihat hasilnya dalam tabel yang disediakan.

1. **USE CASE**

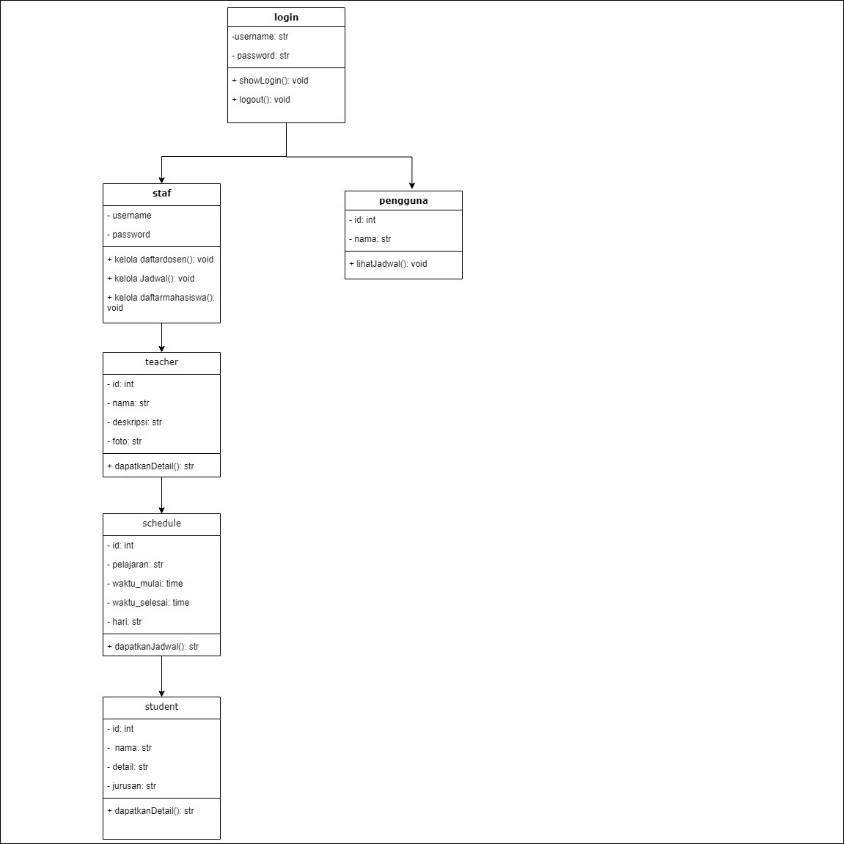


1. Deskripsi Aktor Dan Peran

Use case diagram di atas menggambarkan interaksi antara aktor (admin dan pengguna) dengan sistem aplikasi manajemen jadwal pelajaran. Diagram ini menunjukkan bagaimana masing-masing aktor berinteraksi dengan berbagai fungsi yang ada dalam sistem.

Aktor:

1. Admin: Bertanggung jawab untuk mengelola data yang ada dalam sistem.
2. Pengguna: Dapat melihat jadwal pelajaran yang telah diatur oleh admin.
3. Use Case Diagram Dan Deskripsi Tiap Use Case
   1. Login
      1. Deskripsi: Proses autentikasi untuk memastikan bahwa hanya pengguna yang terotorisasi yang dapat mengakses sistem.
      2. Aktor yang terlibat: Admin, Pengguna
      3. Include: Semua use case lainnya membutuhkan proses login terlebih dahulu.
   2. Mengelolah Daftar Dosen
4. Deskripsi: Admin dapat menambah, mengubah, atau menghapus data dosen yang akan mengajar.
5. Aktor yang terlibat: Admin
6. Include: Login
   1. Mengelolah Jadwal Pelajaran
      1. Deskripsi: Admin dapat mengatur jadwal pelajaran yang meliputi penjadwalan waktu dan tempat pelajaran serta dosen yang mengajar.
      2. Aktor yang terlibat: Admin
      3. Include: Login
   2. Mengelolah Jumlah Mahasiswa
      1. Deskripsi: Admin dapat mengatur jumlah mahasiswa yang terdaftar dalam sistem.
      2. Aktor yang terlibat: Admin
      3. Include: Login
   3. Melihat Jadwal Pelajaran
      1. Deskripsi: Pengguna dapat melihat jadwal pelajaran yang telah diatur oleh admin.
      2. Aktor yang terlibat: Pengguna
      3. Include: Login
7. **PERANCANGAN SISTEM**
8. **CLASS DIAGRAM**



1. Deskripsi Umum

Class diagram di atas menggambarkan struktur dari sistem aplikasi manajemen jadwal pelajaran. Diagram ini menunjukkan berbagai kelas yang ada dalam sistem beserta atribut dan metode yang dimiliki masing-masing kelas.

1. Kelas Dan Deskripsinya
2. Login

* Atribut:
  + username: str
  + password: str
* Metode:
  + showLogin(): void
  + logout(): void
* Deskripsi: Kelas ini bertanggung jawab untuk proses autentikasi pengguna.

1. Staf

* Atribut:
  + username: str
  + password: str
* Metode:
  + kelolaDaftardosen(): void
  + kelolaJadwal(): void
  + kelolaJumlahmahasiswa(): void
* Deskripsi: Kelas ini merepresentasikan admin yang memiliki kemampuan untuk mengelola data dosen, jadwal pelajaran, dan jumlah mahasiswa.

1. Pengguna

* Atribut:
  + id: int
  + nama: str
* Metode:
  + lihatJadwal(): void
* Deskripsi: Kelas ini merepresentasikan pengguna yang dapat melihat jadwal pelajaran.

1. Teacher (Dosen)

* Atribut:
  + id: int
  + nama: str
  + deskripsi: str
  + foto: str
* Metode:
  + dapatkanDetail(): str
* Deskripsi: Kelas ini merepresentasikan data dosen yang mengajar. Setiap dosen memiliki atribut ID, nama, deskripsi, dan foto.

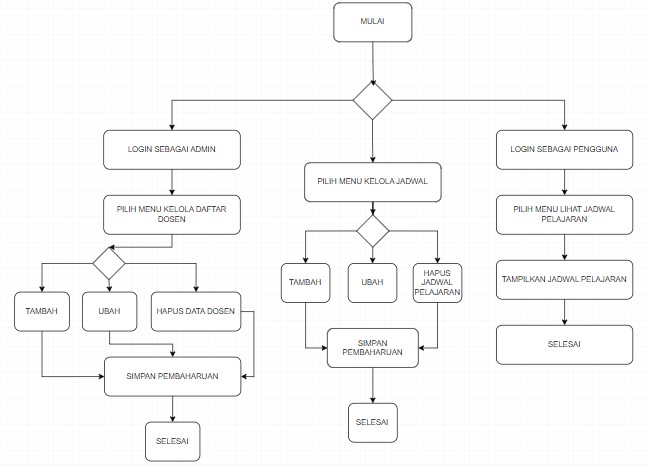
1. Schedule (Jadwal Pelajaran)

* Atribut:
  + id: int
  + pelajaran: str
  + waktu\_mulai: time
  + waktu\_selesai: time
  + hari: str
* Metode:
  + dapatkanJadwal(): str
* Deskripsi: Kelas ini merepresentasikan jadwal pelajaran yang mencakup ID, pelajaran, waktu mulai, waktu selesai, dan hari.

1. Student (Mahasiswa)

* Atribut:
  + id: int
  + nama: str
  + detail: str
  + jurusan: str
* Metode:
  + dapatkanDetail(): str
* Deskripsi: Kelas ini merepresentasikan data mahasiswa yang mencakup ID, nama, detail, dan jurusan.

1. **DIAGRAM AKTIVITAS**



1. Deskripsi Umum

Aktivitas diagram di bawah ini menggambarkan langkah-langkah yang diambil oleh pengguna atau admin dalam menjalankan berbagai fungsi di dalam sistem aplikasi manajemen jadwal pelajaran. Diagram ini menunjukkan alur kerja dari awal hingga akhir untuk setiap aktivitas yang terlibat.

1. Aktivitas
2. Login

* Mulai: Pengguna memulai proses dengan mengakses halaman login.
* Tampilkan Halaman Login: Sistem menampilkan halaman login untuk pengguna memasukkan kredensial mereka.
* Masukkan Kredensial: Pengguna memasukkan username dan password mereka.
* Validasi Kredensial: Sistem memeriksa apakah kredensial yang dimasukkan valid.
  + Kredensial Valid: Jika kredensial valid, pengguna diberikan akses ke halaman laporan.
  + Kredensial Tidak Valid: Jika kredensial tidak valid, sistem menampilkan pesan kesalahan.
* Selesai: Proses login selesai.

1. Mengelola Daftar Dosen

* Mulai: Proses dimulai dengan admin mengakses sistem.
* Login sebagai Admin: Admin melakukan login ke dalam sistem.
* Pilih Menu Kelola Daftar Dosen: Admin memilih opsi untuk mengelola daftar dosen.
* Tambah/Ubah/Hapus Data Dosen: Admin dapat menambah, mengubah, atau menghapus data dosen.
* Simpan Perubahan: Sistem menyimpan perubahan yang dilakukan oleh admin.
* Selesai: Proses pengelolaan daftar dosen selesai.

1. Mengelola Jadwal Pelajaran

* Mulai: Proses dimulai dengan admin mengakses sistem.
* Login sebagai Admin: Admin melakukan login ke dalam sistem.
* Pilih Menu Kelola Jadwal: Admin memilih opsi untuk mengelola jadwal pelajaran.
* Tambah/Ubah/Hapus Jadwal Pelajaran: Admin dapat menambah, mengubah, atau menghapus jadwal pelajaran.
* Simpan Perubahan: Sistem menyimpan perubahan yang dilakukan oleh admin.
* Selesai: Proses pengelolaan jadwal pelajaran selesai.

1. Melihat Jadwal Pelajaran

* Mulai: Proses dimulai dengan pengguna mengakses sistem.
* Login sebagai Pengguna: Pengguna melakukan login ke dalam sistem.
* Pilih Menu Lihat Jadwal Pelajaran: Pengguna memilih opsi untuk melihat jadwal pelajaran.
* Tampilkan Jadwal Pelajaran: Sistem menampilkan jadwal pelajaran kepada pengguna.
* Selesai: Proses melihat jadwal pelajaran selesai.

1. **IMPLEMENTASI**

Implementasi aplikasi ini dilakukan menggunakan bahasa pemrograman Python dengan framework Django untuk pengembangan web. Aplikasi ini dirancang untuk membantu pengelolaan jadwal pelajaran, daftar dosen, dan data mahasiswa. Berikut ini adalah penjelasan tentang struktur proyek dan beberapa potongan kode penting.

1. **PENGUJIAN**

Pengujian adalah tahap penting dalam pengembangan aplikasi untuk memastikan bahwa semua fungsi bekerja sebagaimana mestinya dan bahwa aplikasi memenuhi kebutuhan pengguna. Berikut ini adalah penjelasan tentang metode pengujian yang digunakan, serta beberapa skenario pengujian untuk aplikasi manajemen jadwal pelajaran.

### Metode Pengujian

1. **Pengujian Unit (Unit Testing)** Pengujian unit dilakukan untuk memastikan bahwa setiap unit atau komponen individual dari aplikasi berfungsi dengan benar. Pengujian ini biasanya dilakukan oleh pengembang selama fase pengembangan.
2. **Pengujian Integrasi (Integration Testing)** Pengujian integrasi dilakukan untuk memastikan bahwa berbagai komponen atau unit aplikasi bekerja dengan baik ketika digabungkan. Pengujian ini membantu mengidentifikasi masalah yang mungkin terjadi ketika berbagai bagian aplikasi berinteraksi satu sama lain.
3. **Pengujian Fungsional (Functional Testing)** Pengujian fungsional dilakukan untuk memastikan bahwa aplikasi bekerja sesuai dengan kebutuhan atau spesifikasi yang telah ditetapkan. Pengujian ini berfokus pada pengujian fitur dan fungsionalitas aplikasi.
4. **Pengujian Sistem (System Testing)** Pengujian sistem dilakukan untuk memastikan bahwa seluruh sistem aplikasi berfungsi dengan baik. Pengujian ini mencakup pengujian fungsionalitas, performa, keamanan, dan aspek lainnya dari aplikasi.
5. **Pengujian Penerimaan Pengguna (User Acceptance Testing)** Pengujian penerimaan pengguna dilakukan untuk memastikan bahwa aplikasi memenuhi kebutuhan dan ekspektasi pengguna akhir. Pengujian ini biasanya melibatkan pengguna akhir yang mencoba aplikasi dalam lingkungan yang mirip dengan kondisi operasional yang sebenarnya.

### Skenario Pengujian

Berikut adalah beberapa skenario pengujian untuk aplikasi manajemen jadwal pelajaran:

1. **Skenario Pengujian Login**

**Deskripsi**: Memastikan bahwa pengguna dapat login dengan kredensial yang benar dan tidak bisa login dengan kredensial yang salah.

**Langkah Pengujian**:

* 1. Akses halaman login.
  2. Masukkan username dan password yang valid.
  3. Klik tombol login.
  4. Verifikasi bahwa pengguna diarahkan ke halaman dashboard.
  5. Masukkan username dan password yang tidak valid.
  6. Klik tombol login.
  7. Verifikasi bahwa pesan kesalahan ditampilkan.

1. **Skenario Pengujian Mengelola Daftar Dosen**

**Deskripsi**: Memastikan bahwa admin dapat menambah, mengubah, dan menghapus data dosen.

**Langkah Pengujian**:

* 1. Login sebagai admin.
  2. Akses halaman kelola daftar dosen.
  3. Tambah data dosen baru.
  4. Verifikasi bahwa data dosen baru muncul di daftar.
  5. Ubah data dosen yang ada.
  6. Verifikasi bahwa perubahan disimpan dan ditampilkan dengan benar.
  7. Hapus data dosen.
  8. Verifikasi bahwa data dosen dihapus dari daftar.

1. **Skenario Pengujian Mengelola Jadwal Pelajaran**

**Deskripsi**: Memastikan bahwa admin dapat menambah, mengubah, dan menghapus jadwal pelajaran.

**Langkah Pengujian**:

* 1. Login sebagai admin.
  2. Akses halaman kelola jadwal pelajaran.
  3. Tambah jadwal pelajaran baru.
  4. Verifikasi bahwa jadwal pelajaran baru muncul di daftar.
  5. Ubah jadwal pelajaran yang ada.
  6. Verifikasi bahwa perubahan disimpan dan ditampilkan dengan benar.
  7. Hapus jadwal pelajaran.
  8. Verifikasi bahwa jadwal pelajaran dihapus dari daftar.

1. **Skenario Pengujian Melihat Jadwal Pelajaran**

**Deskripsi**: Memastikan bahwa pengguna dapat melihat jadwal pelajaran yang tersedia.

**Langkah Pengujian**:

* 1. Login sebagai pengguna.
  2. Akses halaman lihat jadwal pelajaran.
  3. Verifikasi bahwa semua jadwal pelajaran yang tersedia ditampilkan dengan benar.

1. **KESIMPULAN**

Aplikasi manajemen jadwal pelajaran yang telah dikembangkan berhasil memenuhi kebutuhan utama untuk mengelola daftar dosen, jadwal pelajaran, dan data mahasiswa. Melalui serangkaian tahapan perancangan yang melibatkan Use Case Diagram, Class Diagram, dan Activity Diagram, aplikasi ini dapat memberikan panduan yang jelas tentang alur kerja dan hubungan antar komponen dalam sistem. Implementasi menggunakan framework Django memungkinkan pengembangan yang cepat dan terstruktur, sementara pengujian unit, integrasi, dan fungsional memastikan setiap komponen dan fungsi bekerja sesuai dengan spesifikasi.

Pengujian penerimaan pengguna menunjukkan bahwa aplikasi ini dapat digunakan oleh pengguna akhir dengan sedikit atau tanpa masalah. Admin dapat dengan mudah mengelola daftar dosen, jadwal pelajaran, dan data mahasiswa melalui antarmuka yang disediakan, dan pengguna dapat melihat jadwal pelajaran dengan tampilan yang jelas dan informatif. Secara keseluruhan, aplikasi ini memberikan solusi efektif untuk manajemen jadwal pelajaran di institusi pendidikan dan siap digunakan dalam lingkungan operasional nyata.