

---

## **PENGEMBANGAN SISTEM INFORMASI UNTUK ADMINISTRASI DATA DOSEN FMIPA UNSRAT BAGIAN AKADEMIK BERBASIS WEB MENGGUNAKAN FRAMEWORK LARAVEL DENGAN ARSITEKTUR MVC (MODEL, VIEW, CONTROLLER)**

Natalia Pelango<sup>1)</sup>, Winsy Ch. D. Weku<sup>2)</sup>, Christian A. J. Soewoeh<sup>3)</sup>, Eliasta Ketaren<sup>4)</sup>

Sistem Informasi

Universitas Sam Ratulangi

Jl. Kampus Unsrat Bahu, Kleak, Malalayang, Kota Manado, Sulawesi Utara

email: nataliapelango106@student.unsrat.ac.id<sup>1)</sup>, winsy\_weku@unsrat.ac.id<sup>2)</sup>,

christian.suwuh@unsrat.ac.id<sup>3)</sup>, eliaستaketaren@unsrat.ac.id<sup>4)</sup>

---

### **Abstrak**

Data administrasi dosen adalah informasi yang berkaitan dengan tenaga pengajar di suatu institusi pendidikan, terutama di perguruan tinggi atau universitas. Data ini sangat penting untuk manajemen dan operasional suatu institusi. Perancangan sistem informasi untuk administrasi data dosen merupakan langkah penting untuk mengelola dan mengoptimalkan data terkait dosen secara efisien. Pengelolaan data secara manual menghadapi berbagai kendala, termasuk ketidakefisienan administratif dan risiko kesalahan manusia. membutuhkan solusi untuk meningkatkan efisiensi administrasi dan pengarsipan data dosen. Penelitian ini bertujuan merancang sistem informasi berbasis web untuk administrasi data dosen menggunakan arsitektur MVC dengan framework Laravel. Metode penelitian yang digunakan adalah pengembangan sistem dengan pendekatan waterfall, meliputi tahapan requirement, system design, implementation, testing, deployment dan maintenance. Hasil penelitian menunjukkan bahwa sistem informasi yang dikembangkan dapat mempermudah penginputan, pengarsipan, dan perekapan data dosen serta meningkatkan efisiensi dan akurasi administrasi. Kesimpulan dari penelitian ini adalah bahwa sistem informasi berbasis web dengan arsitektur MVC pada Laravel efektif dalam mengelola data dosen, meningkatkan efisiensi administrasi dan mengurangi risiko kesalahan manusia dalam pengelolaan data.

**Kata Kunci:** Data Administrasi, Sistem Informasi, MVC, Laravel.

### **1. Pendahuluan**

Pada era digital, pengelolaan data dosen di perguruan tinggi menjadi krusial untuk mendukung operasional lembaga dan meningkatkan kualitas pembelajaran. Penggunaan teknologi internet memfasilitasi perolehan informasi kampus dan implementasi sistem informasi berbasis web di perguruan tinggi negeri. Fakultas MIPA Universitas Sam Ratulangi di Kota Manado, Sulawesi Utara, didirikan pada tahun 1998, memiliki visi menjadi pusat pembelajaran dan penelitian yang kompeten. Divisi bidang akademik dan kerjasama dalam fakultas ini berperan penting dalam mengelola aspek pendidikan, penelitian, dan kerjasama antar lembaga.

Administrasi mencakup berbagai kegiatan seperti pengambilan keputusan, pengembangan kebijakan, pengelolaan informasi, dan koordinasi antara departemen [1]. Pengelolaan data yang semakin besar secara manual dapat menimbulkan ketidakefisienan administratif. Tugas administratif dalam mengelola data dosen, seperti jadwal mengajar, rekam jejak pengajaran, penelitian, dan pengabdian masyarakat, jika dilakukan secara manual memakan waktu lama dan rentan terhadap kesalahan manusia. Oleh karena itu, perancangan sistem informasi data akademik dosen diperlukan untuk memudahkan pengambilan data dan mengurangi risiko kesalahan.

Perancangan sistem informasi berbasis web untuk administrasi data dosen merupakan langkah strategis dalam administrasi dan pengarsipan akademik. Sistem yang dirancang menggunakan arsitektur MVC (Model View Controller) dengan framework Laravel dapat meningkatkan efisiensi dan akurasi administrasi. MVC memisahkan tanggung jawab antara model, view, dan controller, sehingga memfasilitasi pengembangan yang terstruktur dan mudah dipelihara. Penelitian menunjukkan bahwa sistem informasi berbasis web dengan arsitektur MVC pada Laravel efektif dalam mengelola data dosen, meningkatkan efisiensi administrasi, dan mengurangi risiko kesalahan manusia dalam pengelolaan data.

### **2. Landasan Teori**

#### **Sistem Informasi**

Sistem informasi adalah serangkaian komponen yang saling berhubungan dalam mengumpulkan atau memperoleh, memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan kontrol pada sebuah organisasi [2].

**Website**

Website adalah kumpulan halaman web yang diatur secara logis untuk menyajikan informasi atau layanan tertentu kepada pengguna melalui internet [3]. Website adalah platform yang terdiri dari berbagai halaman web yang dihubungkan bersama dan diakses melalui internet menggunakan browser yang menyajikan informasi dan layanan kepada pengguna [4].

**Database**

Database adalah kumpulan data yang terorganisasi dan biasanya memiliki beberapa arti atau kegunaan yang relevan. Data ini biasanya diatur dalam tabel dan dapat diakses secara elektronik melalui sistem manajemen

basis data (*Database Management System, DBMS*) [5].

**MVC (Model View Controller)**

MVC adalah pola desain arsitektur perangkat lunak yang bertujuan untuk memisahkan aplikasi menjadi tiga komponen utama. Model bertanggung jawab untuk menangani data aplikasi dan logika bisnis. Model mengelola data, baik untuk pengambilan dari basis data maupun untuk manipulasi data. Model biasanya tidak memiliki pengetahuan tentang tampilan atau bagaimana data tersebut disajikan. View bertugas untuk menampilkan data kepada pengguna. View mengambil data dari Model dan menyajikannya dalam bentuk yang dapat dipahami oleh pengguna. View juga menangani presentasi dan tata letak dari data. Controller bertindak sebagai penghubung antara Model dan View. Controller menerima input dari pengguna melalui View kemudian memproses dan memutuskan bagaimana berinteraksi dengan Model. Setelah memproses data, Controller menentukan View mana yang harus digunakan untuk menampilkan hasil kepada pengguna [6].

**Laravel**

Laravel adalah framework PHP yang didesain untuk memudahkan pengembangan aplikasi web dengan menyediakan berbagai fitur yang powerful dan ekstensif. Laravel menggunakan konsep-konsep modern dalam pengembangan web dan menyediakan banyak alat bantu untuk mempercepat proses pengembangan. Laravel menawarkan banyak fitur bawaan yang memungkinkan pengembang untuk fokus pada logika bisnis aplikasi daripada menghabiskan waktu untuk menulis kode repetitif. Beberapa fitur Laravel yaitu *Routing, Blade Templating, Eloquent ORM, Middleware, Authentication, Artisan CLI, Database Migrations, Eloquent Relationship, Validation* dan *Security* untuk mengelola interaksi dengan database serta sistem autentikasi yang mudah diimplementasikan [7].

**Black Box Testing**

*Black Box Testing* adalah metode pendekatan dalam pengujian perangkat lunak di mana fungsi-fungsi sistem diuji tanpa memperhatikan struktur internal dari kode program atau logika implementasi yang digunakan. Metode ini melibatkan pengujian input dan output dari suatu sistem tanpa pengetahuan tentang bagaimana proses internal bekerja. Metode Black Box memperlakukan sistem sebagai "kotak hitam" dimana hanya input dan output yang penting untuk diuji. Metode pengujian dilakukan tanpa memerlukan pengetahuan mendalam tentang implementasi internal dari sistem yang diuji [8].

**White Box Testing**

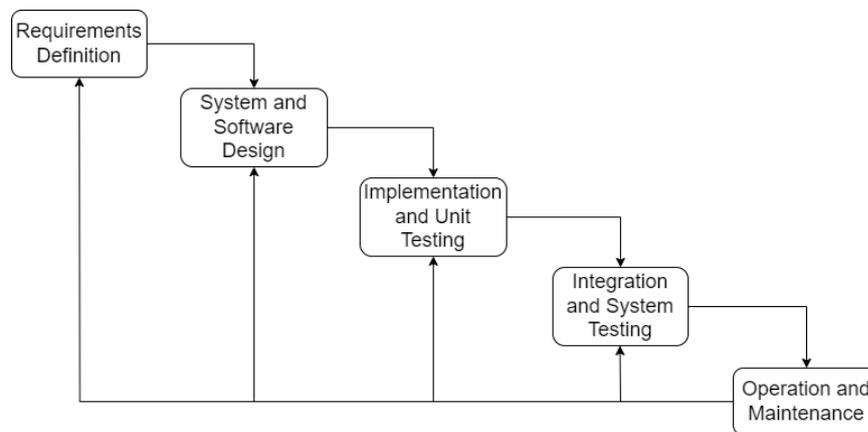
*White Box Testing* merupakan metode pengujian perangkat lunak yang menggunakan pengetahuan tentang struktur internal dan implementasi perangkat lunak. Pengujian ini dilakukan dengan memeriksa logika kontrol, aliran data, kondisi, loop, dan berbagai jalur kode lainnya untuk memastikan bahwa setiap bagian kode bekerja sesuai dengan yang diharapkan. Metode ini berfokus untuk memastikan bahwa semua aspek dari kode sumber diuji secara menyeluruh, sehingga memungkinkan pengembang untuk menemukan dan memperbaiki kesalahan lebih awal dalam proses pengembangan [9].

**3. Metode Penelitian****Waktu dan Tempat Penelitian**

Waktu penelitian dimulai pada bulan Februari - Juni 2024 dengan tempat penelitian di Fakultas Matematika dan Ilmu Pengetahuan Alam bagian Akademik. Teknik pengumpulan data yang digunakan yaitu, observasi, wawancara dan studi pustaka.

**Metode Penelitian**

Metode Waterfall adalah salah satu model proses dalam rekayasa perangkat lunak yang menggambarkan pendekatan linier untuk pengembangan perangkat lunak.



**Gambar 1.** Tahapan Metode Waterfall

SDLC (*Software Development Life Cycle*) atau Siklus Hidup Pengembangan Perangkat Lunak adalah konsep yang lebih umum yang merujuk pada keseluruhan proses pengembangan perangkat lunak dari awal hingga akhir. SDLC mencakup semua yang diperlukan untuk merencanakan, merancang, mengembangkan, menguji dan memelihara perangkat lunak [9].

Penjelasan tahapan metode Waterfall adalah sebagai berikut, yaitu :

*Analisis Kebutuhan (Requirements Analysis)*

Tahap ini melibatkan identifikasi kebutuhan pengguna dan sistem yang akan dikembangkan. Hasil dari tahap ini adalah dokumen spesifikasi kebutuhan.

*Perancangan Sistem (System and Software Design)*

Pada tahap ini, sistem dan perangkat lunak direncanakan secara keseluruhan. Tahap ini mencakup perancangan arsitektur sistem, desain detail modul, dan antarmuka pengguna.

*Implementasi dan Pengujian Unit (Implementation and Unit Testing)*

Tahap ini melibatkan implementasi kode program berdasarkan desain yang telah dibuat. Selain itu, dilakukan juga pengujian unit untuk memastikan setiap unit atau modul berfungsi dengan baik.

*Integrasi dan Pengujian Sistem (Integration and System Testing)*

Modul yang telah diuji kemudian diintegrasikan untuk membentuk sistem yang lengkap. Sistem ini kemudian diuji untuk memastikan bahwa semua komponen bekerja dengan baik bersama-sama.

*Operasi dan Pemeliharaan (Operation and Maintenance)*

Setelah sistem diluncurkan, tahap ini mencakup penggunaan sistem dalam operasi sehari-hari serta pemeliharaan sistem untuk memperbaiki bug atau menambah fitur baru [10]

#### **4. Hasil Penelitian**

##### **Analisis Kebutuhan**

Pada halaman menu admin, tersedia fitur-fitur seperti menu login, dashboard, daftar dosen, daftar user, profil, dan logout. Menu login memungkinkan admin untuk mengisi username dan password yang telah ada dan divalidasi oleh sistem agar dapat masuk ke sistem. Dashboard memungkinkan admin untuk melihat jumlah dosen dan jumlah user yang aktif. Daftar dosen memungkinkan admin untuk melihat dan mencetak daftar dosen serta detail data administrasi yang sudah terisi. Daftar user memungkinkan admin untuk melihat user yang aktif dan membuat akun baru dengan menentukan role user sebagai batas akses dari masing-masing akun tersebut. Menu profil memungkinkan admin untuk melihat profil akun dan mengubah password akun, dan menu logout memungkinkan admin untuk keluar dari akun setelah digunakan.

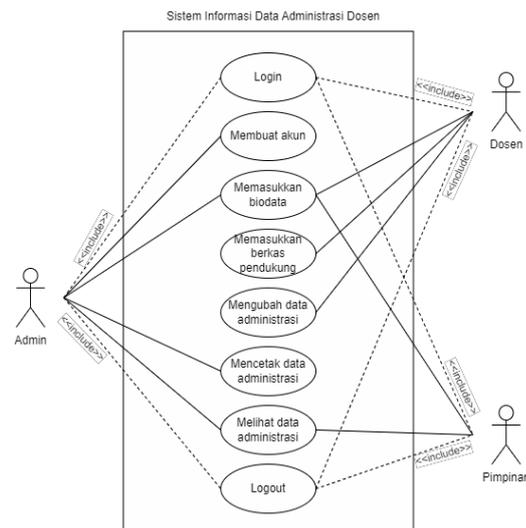
Halaman menu dosen mencakup fitur-fitur seperti menu login, dashboard, biodata, berkas pendukung, profil, dan logout. Menu login memungkinkan dosen untuk mengisi username dan password yang telah ada dan divalidasi oleh sistem agar dapat masuk ke sistem. Dashboard menampilkan profil dan biodata dosen serta presentase data administrasi dan berkas pendukung yang dimasukkan. Menu biodata memungkinkan dosen untuk mengisi atau memperbarui biodata mereka. Menu berkas pendukung memungkinkan dosen untuk memasukkan berkas-berkas pendukung dalam bentuk file PDF (Portable Document Format) untuk melengkapi data administrasi. Menu profil

memungkinkan dosen untuk melihat profil akun dan mengubah password akun, dan menu logout memungkinkan dosen untuk keluar dari akun setelah selesai digunakan.

Halaman menu pimpinan mencakup fitur-fitur seperti dashboard, daftar dosen, profil, dan logout. Dashboard memungkinkan pimpinan untuk menampilkan jumlah dosen dan jumlah pengguna. Daftar dosen memungkinkan dekan dan wakil dekan untuk melihat daftar dosen serta detail data administrasi dosen. Menu profil memungkinkan pimpinan untuk melihat profil akun dan mengubah password akun, dan menu logout memungkinkan dekan dan wakil dekan untuk keluar dari akun setelah selesai digunakan.

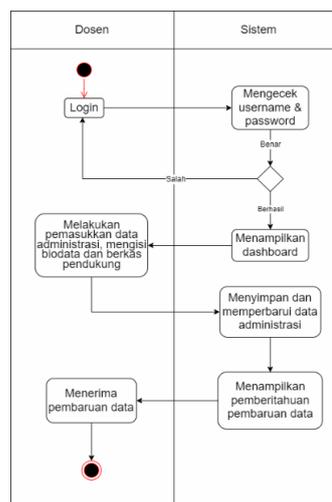
### Perancangan Sistem

Use case diagram mengilustrasikan relasi dan fungsi sistem dari sudut pandang pengguna untuk menjelaskan berbagai jenis interaksi antara pengguna (aktor) dan sistem yang digunakan. Pada aktor Dosen memiliki tiga use case yaitu memasukkan biodata, memasukkan berkas pendukung dan dapat mengubah data administrasi. Aktor Admin memiliki empat use case yaitu membuat akun, memasukkan biodata, mencetak data dan melihat data administrasi. Aktor pimpinan memiliki dua use case yaitu memasukkan biodata dan melihat data administrasi. Ketiga aktor tersebut sudah termasuk dengan login dan logout pada sistem. use case diagram yang dijelaskan dapat dilihat pada Gambar 2.



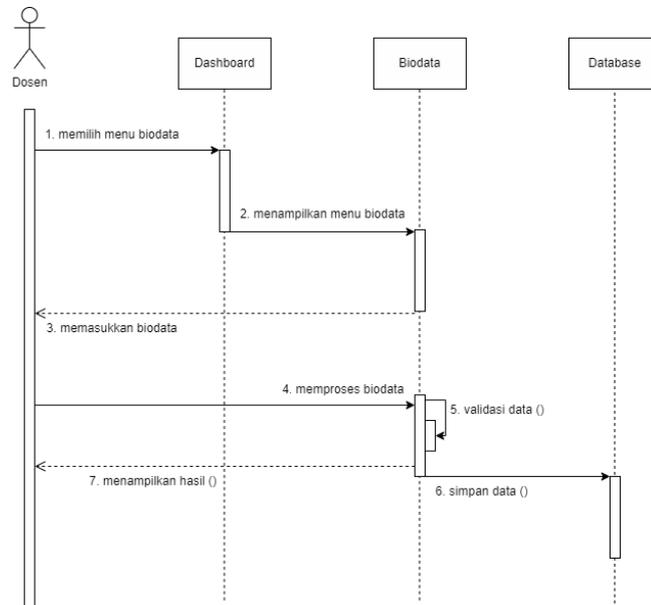
**Gambar 2.** Use Case Diagram Sistem Informasi Data Dosen

Pada Gambar 3, Activity Diagram pada dosen menggambarkan aktivitas yang dilakukan dosen di dalam sistem mulai dari awal aktivitas sampai akhir. Setelah pembuatan akun dilakukan oleh admin, dosen dapat membuka halaman website sistem. Sebelum masuk ke dalam sistem, dosen melakukan login terlebih dahulu. Jika login gagal maka sistem akan kembali ke halaman login, jika berhasil maka sistem akan langsung menampilkan halaman dashboard. Dosen dapat memasukkan data administrasi seperti biodata dan berkas pendukung yang diperlukan, jika telah selesai melakukan proses administrasi maka sistem akan menyimpan data untuk digunakan oleh admin dalam pengelolaan data administrasi.



**Gambar 3.** Activity Diagram Dosen

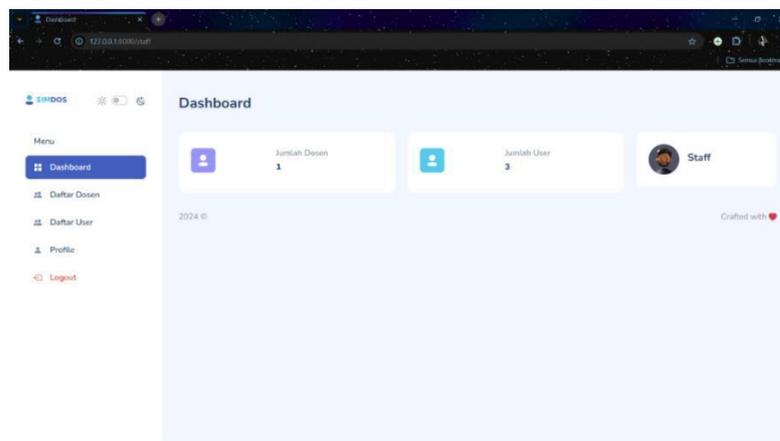
Pada Gambar 4 menunjukkan Sequence Diagram yang menggambarkan alur interaksi antara aktor (dosen), dashboard dan database dalam proses pengelolaan administrasi. Proses dimulai ketika aktor mengakses dashboard kemudian memilih opsi untuk melihat atau mengelola biodata. Dashboard mengirimkan permintaan tersebut ke modul biodata. Modul biodata, pada gilirannya, mengirimkan query ke database untuk mengambil data biodata yang diperlukan. Database merespons dengan mengirimkan data biodata kembali ke modul biodata. Setelah menerima data dari database, modul biodata memproses data tersebut dan mengirimkan hasilnya kembali ke dashboard. Akhirnya, dashboard menampilkan data biodata yang telah diproses kepada aktor. Sequence diagram ini menunjukkan alur interaksi yang terstruktur dan jelas mulai dari akses awal pengguna, pengambilan data dari Database, hingga penampilan data kepada pengguna melalui Dashboard.



**Gambar 4.** Sequence Diagram Dosen

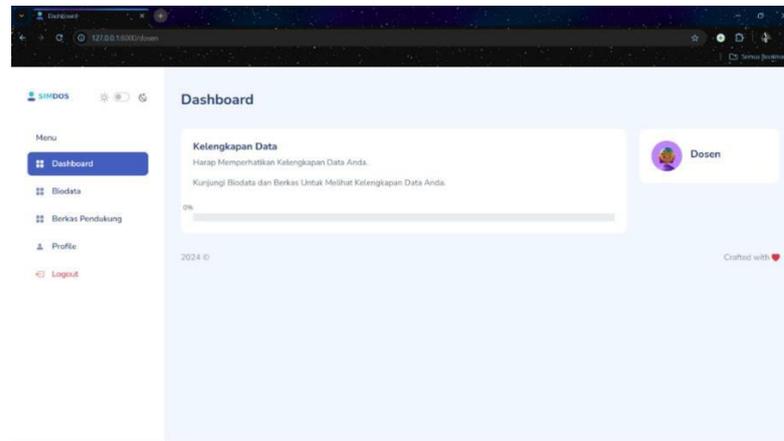
### Implementasi Sistem

Menu Admin adalah menu yang hanya dapat diakses oleh Staff Akademik. Menu ini terdiri dari halaman login, halaman menu dashboard, daftar dosen, daftar user, profil, dan menu logout. Pada menu ini terdapat tombol menu Logout yang berfungsi untuk keluar dari Menu Admin menuju Halaman Login.



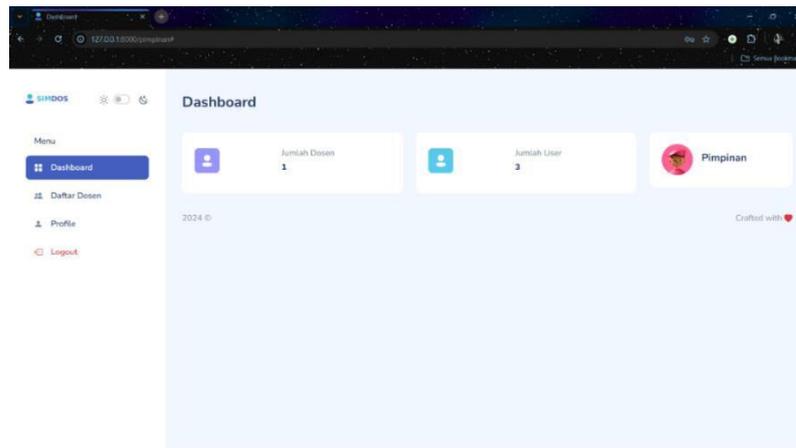
**Gambar 5.** Tampilan Menu Admin

Menu Dosen adalah menu yang dapat diakses oleh Dosen. Menu ini terdiri dari halaman login, halaman menu dashboard, biodata, berkas pendukung, profil, dan menu logout. Pada menu ini terdapat tombol menu Logout yang berfungsi untuk keluar dari menu dosen menuju ke Halaman Login.



**Gambar 6.** Tampilan Menu Dosen

Menu Pimpinan adalah menu yang hanya dapat diakses oleh Dekan dan Wakil Dekan. Menu ini terdiri dari halaman login, halaman menu dashboard, daftar dosen, profil, dan menu logout. Pada menu ini terdapat tombol menu logout yang berfungsi untuk keluar dari Menu Pimpinan menuju ke Halaman Login.



**Gambar 7.** Tampilan Menu Pimpinan

### Pengujian Sistem

Pengujian sistem menggunakan metode Blackbox Testing dan Whitebox Testing dimana masing-masing metode memiliki pendekatan, tujuan dan teknik yang berbeda.

Pengujian yang pertama menggunakan metode Black Box Testing. Metode ini menitikberatkan pada spesifikasi fungsional perangkat lunak di mana tester menetapkan serangkaian kondisi input dan menguji spesifikasi fungsional program tersebut. Dalam proses Black Box testing, program diuji dengan memasukkan data pada setiap form. Pengujian ini penting untuk memastikan bahwa program berjalan sesuai dengan kebutuhan yang ditetapkan.

**Tabel 1.** Black Box Testing Menu Dosen

No.	Komponen Pengujian	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Halaman Login	Menampilkan halaman login yang berisi form username dan password	Sistem menampilkan halaman login yang berisi form username dan password	Sesuai	Berhasil
2.	Tombol Sign In	Mengklik tombol sign in	Sistem menampilkan halaman menu utama	Sesuai	Berhasil
3.	Menu Biodata	Menampilkan halaman biodata yang berisi data administrasi	Sistem menampilkan halaman menu biodata	Sesuai	Berhasil

		Mengklik tombol tambah data	Sistem menampilkan form tambah data dan menyimpan data yang dimasukkan	Sesuai	Berhasil
4.	Tombol Tambah Data				
5.	Menu Berkas Pendukung	Menampilkan halaman berkas pendukung yang berisi form untuk memasukkan berkas pendukung	Sistem menampilkan halaman derkas pendukung dalam bentuk tabel yang berisi file data berkas pendukung	Sesuai	Berhasil
6.	Tombol Aksi	Mengklik tombol aksi	Sistem menampilkan form tambah berkas file dan menyimpan data	Sesuai	Berhasil
7.	Menu Profile	Menampilkan profile pengguna dan form mengubah password	Sistem menampilkan profile pengguna dan form mengubah password	Sesuai	Berhasil
8.	Tombol Update Data	Mengklik tombol update data	Sistem mengupdate nama yang diubah	Sesuai	Berhasil
9.	Tombol Update Password	Mengklik tombol update password	Sistem mengubah password dan menampilkan pemberitahuan password telah diubah	Sesuai	Berhasil
10.	Menu Logout	Mengklik logout	menu Sistem keluar dari akun pengguna dan kembali ke halaman login	Sesuai	Berhasil

Dalam pengujian sistem ini, digunakan metode White Box Testing dimana metode pengujian perangkat lunak dengan menggunakan Unit Testing. White box testing adalah metode pengujian perangkat lunak yang fokus pada pengujian struktur internal atau arsitektur aplikasi. Pengujian ini melibatkan pengecekan kode sumber untuk memastikan semua jalur eksekusi, kondisi, loop, dan pernyataan telah diuji dengan benar.

**Tabel 2.** Pengujian White Box

No	Test Case	Rute	Deskripsi Kasus	Uji	Langkah Uji	Hasil Pengujian
1.	LoginTest.php	/login	Route index Login		1. Akses route /login   2. Periksa status response	Status response 200
		/login	Pengguna bisa login dengan kredensial yang benar		1. Masuk dengan email dan password yang benar   2. Kirim POST request ke /login dengan kredensial yang bena	User terautentikasi   Redirect ke /dashboard

2.	StaffTest.php	/staff	Route index Staff	1. Akses route /staff   2. Periksa status response	Status response 200
		/staff/daftar-dosen	Staff dapat melihat daftar dosen	1. Lakukan login sebagai staff   2. Akses route /staff/daftar-dosen   3. Periksa status response dan jumlah data	Data dosen dalam format JSON
3.	DosenTest.php	/dosen	Route index Dosen	1. Akses route /dosen   2. Periksa status response	Status response 200
		/dosen/biodata	Dosen bisa memasukkan data administrasi	1. Buat user dosen   2. Lakukan login sebagai dosen   3. Kirim POST request ke /dosen/administrasi dengan data administrasi	Data administrasi tersimpan di database
4.	Pimpinan Test.php	/pimpinan	Route index Pimpinan	1. Akses route /pimpinan   2. Periksa status response	Status response 200
		/dosen/{id}	Pimpinan dapat melihat detail data dosen	1. Lakukan login sebagai pimpinan   2. Kirim GET request ke /dosen/{id} untuk melihat detail dosen	Status response 200     Data JSON harus sesuai dengan detail dosen

## 5. Kesimpulan

Berdasarkan penelitian yang dilakukan bahwa pengelolaan data dosen di perguruan tinggi sangat penting untuk mendukung operasional lembaga dan meningkatkan kualitas pembelajaran. Kompleksitas administrasi manual mengarah pada ketidakefisienan dan potensi kesalahan. Sistem menggunakan arsitektur MVC (Model View Controller) dengan penerapan framework Laravel untuk memastikan pengelolaan data, presentasi, dan logika pengontrolan yang terstruktur dan mudah dipelihara. Hal ini diharapkan dapat meningkatkan keakuratan data, mempercepat pengolahan informasi, dan menghindari kesalahan dalam administrasi data dosen. Pengujian whitebox memastikan bahwa semua komponen kode telah diuji secara menyeluruh, termasuk logika internal dan struktur kontrol program, sehingga setiap bagian berfungsi dengan benar dan tidak ada kesalahan dalam implementasi. Sementara itu, pengujian blackbox memastikan bahwa sistem memenuhi kebutuhan fungsional yang telah ditentukan tanpa memperhatikan struktur internalnya, dengan menguji antarmuka dan respons terhadap input pengguna. Hasil pengujian menunjukkan bahwa sistem informasi berbasis web ini berfungsi sesuai dengan yang diharapkan, memenuhi semua kebutuhan pengguna dan dapat diimplementasikan di lingkungan nyata.

## 6. Daftar Pustaka

- [1] G. H. Frederickson, K. B. Smith, C. Larimer, and M. J. Licari, *The Public Administration Theory Primer*, 3rd ed. Routledge, 2016.

- [2] K. C. Laudon and J. P. Laudon, *Management Information Systems: Managing the Digital Firm*. Pearson, 2012.
- [3] E. Turban, E. McLean, and J. Wetherbe, *Information Technology for Management: Transforming Organizations in the Digital Economy*. Wiley, 2008.
- [4] R. Stair and G. Reynolds, *Principles of Information Systems*. Cengage Learning, 2010.
- [5] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed. Pearson, 2015.
- [6] C. Larman, *Applying UML and patterns: An introduction to object-oriented analysis and design and iterative development*, 3rd ed. Prentice Hall, 2004.
- [7] M. Stauffer, *Laravel: Up & Running*. O'Reilly Media, 2016.
- [8] S. Desikan, G. Ramesh, and R. Nagrajan, *Software Testing: Principles and Practices*. McGraw-Hill, 2006.
- [9] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 8th ed. McGraw-Hill Education, 2014.
- [10] I. Sommerville, *Software Engineering*, 10th ed. Pearson, 2016