



Seminar Nasional Ilmu Teknik dan Aplikasi Industri (SINTA)

Homepage: sinta.eng.unila.ac.id



Penerapan *admin panel package* untuk optimasi produktivitas pengembangan *backend* sistem layanan akademik

WE Sulistiono*, Y Mulyani, M Mardiana, NPT Prezilia, P Indriyanti, ZA Basyary

Universitas Lampung; Jl. Prof. Dr. Ir. Sumantri Brojonegoro No.1 35145

INFORMASI ARTIKEL

ABSTRAK

Riwayat artikel:

Diterima 12/01/2026

Direvisi 02/02/2026

Kata kunci:

Sistem Layanan Akademik;
Produktivitas;
Admin Panel Package;
Laravel;
Backend;

Correspondent Email:

eko@eng.unila.ac.id

Sistem layanan akademik menjadi salah satu contoh digitalisasi dalam dunia pendidikan yang bersifat kompleks karena banyaknya modul serta role pengguna dalam alur bisnis sistem. Tingginya tingkat kompleksitas dari sistem layanan akademik dapat mempengaruhi waktu pengembangan *backend* sistem. Banyaknya formulir dan tabel yang digunakan pada sistem dapat mempengaruhi waktu pengembangan pada fitur CRUD (*create, read, update, delete*), routing sistem serta validasi form. Penerapan *admin panel package* yang diintegrasikan dengan *framework* seperti Laravel dapat membantu menghemat waktu pengembangan karena adanya fitur atau komponen yang siap digunakan sehingga pengembang tidak perlu menuliskan kode dari awal. Penelitian ini bertujuan untuk mengevaluasi penerapan *admin panel package* dalam pengembangan *backend* layanan akademik, dengan fokus pada peningkatan produktivitas pengembang. Adanya pengaruh terhadap waktu pengembangan serta konsistensi penulisan kode dalam pengembangan sistem dapat berpengaruh pada tingkat produktivitas tim pengembang. Semakin tinggi produktivitas, maka semakin cepat, hemat dan terstruktur dalam proses pengembangan sistemnya. Metode yang digunakan adalah studi kasus dengan pendekatan Rapid Application Development (RAD). Hasil pengukuran tingkat produktivitas tim pengembang akan memberikan informasi terkait seberapa efektif penggunaan *admin panel* seperti Filament dalam mempercepat waktu pengembangan sistem. Informasi ini penting untuk menentukan strategi pengembangan yang optimal untuk mendukung kinerja tim pengembang.

1. Pendahuluan

Dalam era transformasi digital, perguruan tinggi dituntut untuk menyediakan layanan akademik yang terintegrasi, efisien, dan mudah diakses oleh seluruh civitas akademika. Sistem informasi layanan akademik pada program studi berperan penting dalam mendukung berbagai aktivitas seperti layanan pendaftaran seminar, verifikasi berkas seminar serta pengaturan jadwal seminar yang dapat diakses oleh berbagai jenis pengguna seperti mahasiswa, dosen dan admin.. Untuk itu, pengembangan sistem yang andal dan dapat dikembangkan secara berkelanjutan menjadi kebutuhan utama dalam lingkungan pendidikan tinggi.

Pengembangan *backend* sebagai fondasi dari sistem informasi memiliki tantangan tersendiri. Selain harus memenuhi kebutuhan fungsional yang kompleks, proses pengembangan *backend* juga harus mampu memastikan keamanan, kecepatan akses, serta kemudahan dalam pemeliharaan. Dalam praktiknya, membangun *backend* secara manual dengan Laravel konvensional memerlukan waktu yang panjang, penulisan kode yang berulang, dan tingkat konsistensi yang tinggi. Banyaknya formulir dan tabel yang digunakan pada sistem dapat mempengaruhi waktu pengembangan pada fitur CRUD (*create, read, update, delete*), routing sistem serta validasi form. Selain itu, tim pengembang juga perlu membuat kode untuk tampilan sistem

termasuk tabel dan formulir yang digunakan sehingga waktu yang dibutuhkan untuk mengembangkan sistem menjadi lebih lama. Kurangnya standarisasi dalam penulisan kode juga dapat menyebabkan ketidakkonsistenan dalam struktur kode yang berdampak pada sulitnya proses pemeliharaan (Palomba dkk., 2018).

Penerapan *admin panel package* dalam pengembangan *backend* sistem layanan akademik dapat menjadi solusi terhadap permasalahan tersebut. *Admin panel package* yang diintegrasikan dengan *framework* seperti Laravel dapat membantu menghemat waktu pengembangan karena adanya fitur atau komponen seperti manajemen CRUD, hak akses pengguna, formulir dan validasi serta notifikasi sistem yang sudah siap digunakan sehingga pengembang tidak perlu menuliskan kode dari awal (Wijanarko, 2025). Contoh *admin panel package* yang populer digunakan oleh para pengembang khususnya pada *framework* Laravel yaitu Filament, LaraAdmin serta Voyager. Selain itu, adanya komponen yang dapat digunakan kembali (*reusable*) dalam *admin panel package* dapat menjaga konsistensi penulisan kode sehingga kode menjadi lebih bersih, mengurangi jumlah kesalahan (*bug*), serta mempercepat proses *debugging* dan pemeliharaan sistem. Adanya pengaruh terhadap waktu pengembangan serta konsistensi penulisan kode dalam pengembangan sistem dapat berpengaruh pada tingkat produktivitas tim pengembang (Cheng dkk., 2022).

Meskipun penggunaannya semakin populer, namun belum banyak kajian akademik yang membahas penerapan *admin panel package* dalam konteks sistem informasi layanan akademik, khususnya dari perspektif produktivitas pengembangan. Padahal, hal tersebut dapat menjadi bahan evaluasi untuk strategi teknis yang dapat mendukung percepatan pengembangan sistem informasi. Berdasarkan kondisi tersebut, penelitian ini dilakukan untuk menerapkan dan mengevaluasi pemanfaatan *admin panel package* dalam membangun *backend* layanan akademik. Dengan melakukan studi kasus dan analisis produktivitas pengembangan, diharapkan penelitian ini dapat memberikan kontribusi dalam bentuk rekomendasi teknis bagi institusi maupun pengembang sistem informasi di lingkungan pendidikan tinggi.

2. Tinjauan Pustaka

2.1. Admin Panel Package

Admin panel package merupakan salah satu package yang dapat digunakan pada *framework* pemrograman seperti Laravel, Django, Ruby, Node.js dan lain sebagainya. *Package* ini menyediakan berbagai macam komponen dan fitur siap pakai yang dirancang untuk membangun antarmuka admin yang modern, interaktif

dan mudah dikelola. Pemanfaatan *admin panel package* tidak hanya pada pengembangan UI sistem pada sisi pengguna (*frontend*) tetapi juga pada sisi server (*backend*).

2.2. Laravel Filament

Filament menjadi salah satu *admin panel package* yang dapat digunakan pada *framework* Laravel. Filament menawarkan berbagai fitur dan komponen yang dapat dipakai oleh pengembang seperti panel builder, form builder, table builder, notifications dan sebagainya. Penggunaan komponen yang ada pada Filament seperti form builder dapat meningkatkan produktivitas pengembang dengan mengurangi waktu yang dibutuhkan untuk menuliskan kode secara manual (Wijanarko, 2025).

2.3. Konsep Produktivitas Pengembangan Perangkat Lunak

Produktivitas dapat diartikan sebagai rasio antara jumlah output yang dihasilkan dengan jumlah input atau usaha yang digunakan. Produktivitas dapat diukur melalui rumus berikut (Banker dan Kauffman, 1991).

$$Productivity = \frac{Size\ of\ application\ developed}{Labor\ consumed\ during\ development} \quad (1)$$

Size of application developed merujuk pada ukuran aplikasi atau sistem yang dikembangkan. Sedangkan *labor consumed during development* merujuk pada jumlah waktu yang dibutuhkan dalam pengembangan suatu sistem.

Simple Model of Productivity dapat diartikan sebagai rasio output yang dihasilkan terhadap sumber daya yang digunakan (Card, 2006). Adapun teknik yang akan digunakan untuk mengukur produktivitas dalam penelitian ini yaitu *Functional Productivity*. *Functional Productivity* merupakan pengukuran terhadap banyaknya *functional points* yang dapat diselesaikan dalam waktu tertentu. Pengukuran *Functional Productivity* dapat diukur melalui rumus berikut.

$$Functional\ productivity = \frac{Number\ of\ Function\ Points}{Man\ hours\ or\ days\ or\ months} \quad (2)$$

Function Points dapat diartikan sebagai unit pengukuran perangkat lunak melalui identifikasi fungsionalitas sistem berdasarkan karakteristik yang ada. Adapun perhitungan function points dapat dituliskan dengan rumus sebagai berikut.

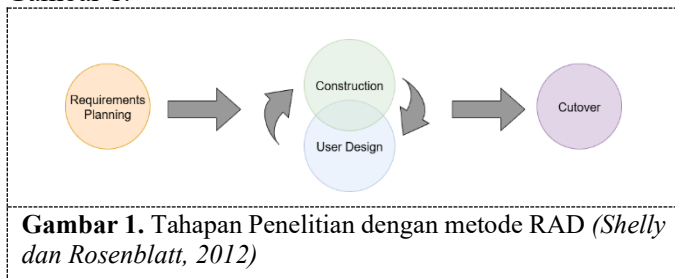
$$Function\ Points = CFP \times (0.65 + (0.01 \times RCAF)) \quad (2)$$

CFP atau Crude Function Points merupakan perhitungan terhadap 5 komponen pada perangkat lunak dengan memberikan bobot berdasarkan level

kompleksitasnya yang terbagi menjadi 3 kategori yakni mudah, sedang dan kompleks. Adapun 5 komponen tersebut, yaitu External Input (EI), External Output (EO), External Inquiry (EI), Internal Logical File (ILF) dan External Interface File (EIF). Kemudian, RCAF atau Relative Complexity Adjustment Factor merupakan pengukuran kompleksitas sistem berdasarkan 14 karakteristik mulai dari “tidak memiliki pengaruh” hingga “memiliki pengaruh besar” pada sistem dengan skala 0-5..

3. Metodologi

Penelitian ini menerapkan pendekatan *Rapid Application Development (RAD)*, yaitu metode pengembangan perangkat lunak yang menekankan kecepatan iterasi, pembuatan prototipe, serta keterlibatan pengguna secara aktif. Pendekatan ini sesuai digunakan untuk mengevaluasi efektivitas penggunaan *admin panel package* dalam mempercepat proses pengembangan *backend* layanan akademik. Adapun tahapan penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian dengan metode RAD (Shelly dan Rosenblatt, 2012)

3.1. Requirements planning

Tahapan ini mengkaji kendala-kendala yang umum terjadi dalam pengembangan *backend* sistem akademik secara konvensional dan kebutuhan fungsional utama dari sistem layanan akademik. Pada tahap ini juga dilakukan perencanaan dan studi literatur untuk meninjau pendekatan teknologi yang relevan yang digunakan dalam penelitian ini yaitu Filament serta merancang indikator untuk mengukur produktivitas, meliputi durasi pengembangan serta jumlah *function points* dari fitur yang diselesaikan.

3.2. User design

Pada tahap ini, dilakukan perancangan prototipe sistem dengan merancang modul-modul utama dalam bentuk resource, relasi data, dan hak akses pengguna.

3.3. Construction

Pada tahap ini, dilakukan implementasi dimana *backend* sistem dikembangkan menggunakan pendekatan berbasis komponen yang telah dirancang sebelumnya. Proses ini dilakukan secara modular dan

iteratif agar memudahkan proses validasi dan penyempurnaan pada tahap berikutnya.

3.4. Cutover

Pada tahap *cutover* dilakukan pengujian terhadap hasil pengembangan. Seluruh aktivitas pengembangan didokumentasikan, termasuk waktu pengerjaan, kendala teknis, serta pengujian terhadap fungsionalitas sistem agar sesuai dengan kebutuhan pengguna.

3.5. Evaluasi

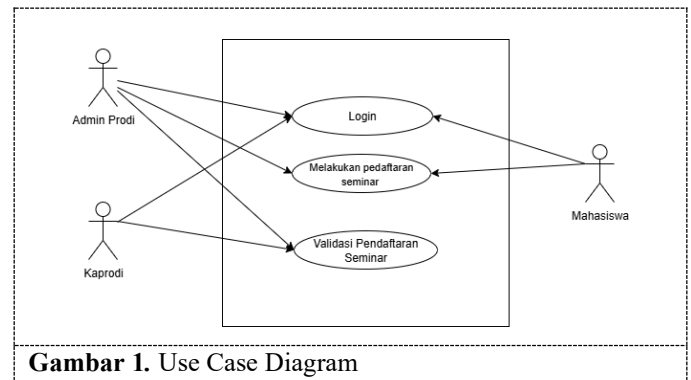
Pada tahap ini, dilakukan pengembangan ulang modul yang sama dengan pendekatan Laravel konvensional tanpa bantuan admin panel. Seluruh data dari kedua pendekatan dikumpulkan secara sistematis untuk keperluan perbandingan. Fokus evaluasi meliputi efisiensi pengembangan, kualitas struktur kode, dan kemudahan pemeliharaan.

4. Hasil dan pembahasan

4.1. User Design

1) Use Case Diagram

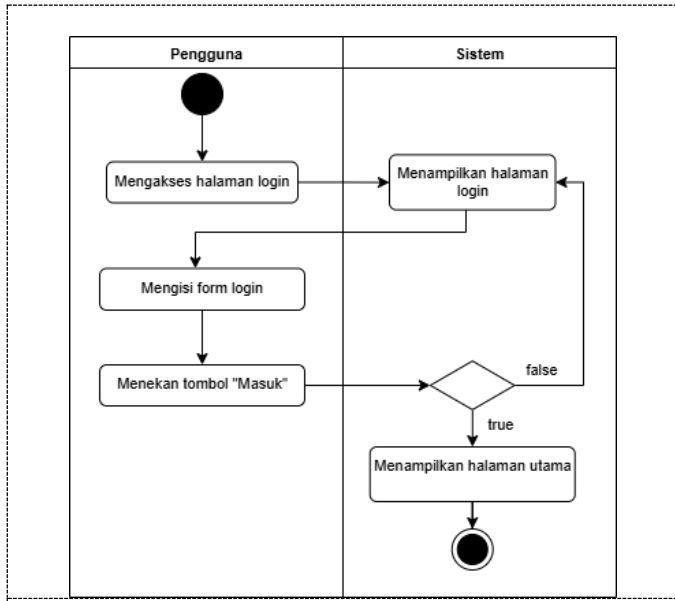
Gambar 2 memperlihatkan fungsionalitas yang terdapat dapat sistem layanan mahasiswa, yang terdiri atas tiga *use case*



Gambar 1. Use Case Diagram

2) Activity Diagram

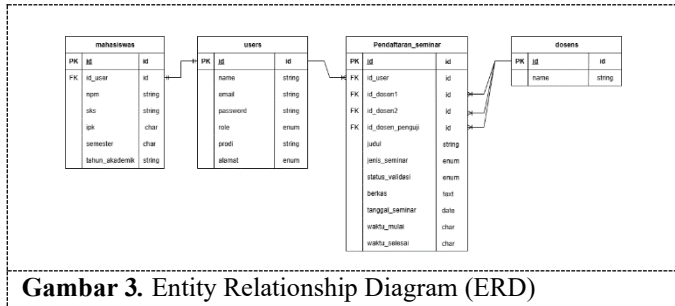
Masing-masing *use case* pada Gambar 2 dimodelkan perilakunya menggunakan *Activity Diagram* pada Gambar 3 dan 4.



Gambar 2. Activity Diagram Login

3) Entity Relationship Diagram

Data yang disimpan di dalam basis data dimodelkan dengan *entity relationship diagram*, yang dapat dilihat pada Gambar 5.

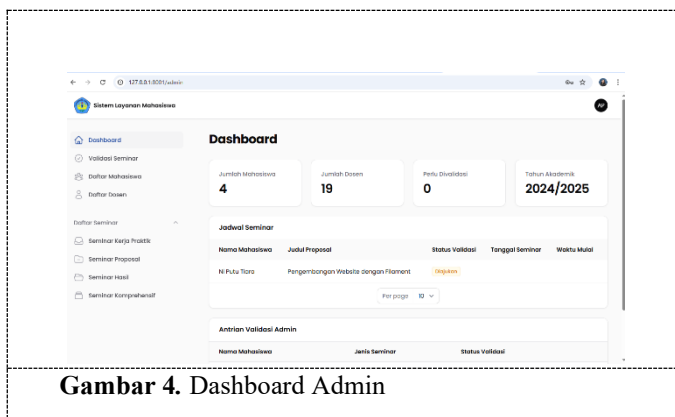


Gambar 3. Entity Relationship Diagram (ERD)

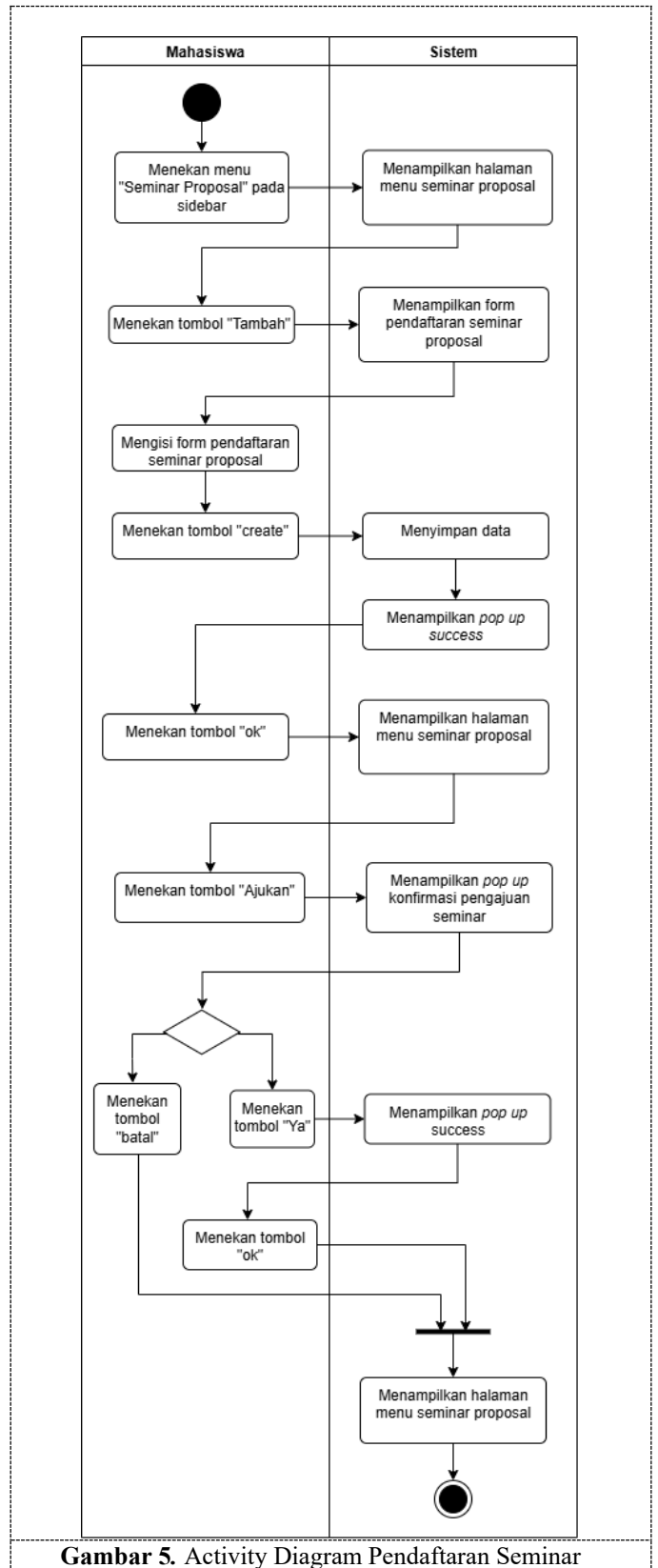
4.2. Construction

1) Role Admin

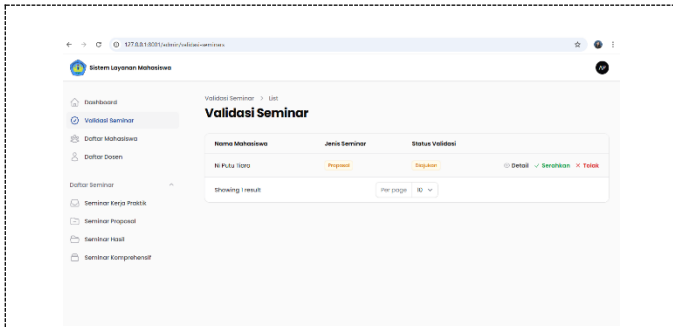
Gambar 6 menampilkan UI dari dashboard untuk Admin. Sementara Gambar 7 menampilkan UI untuk validasi seminar.



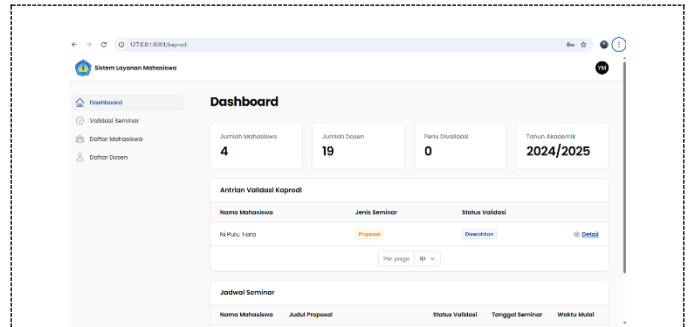
Gambar 4. Dashboard Admin



Gambar 5. Activity Diagram Pendaftaran Seminar



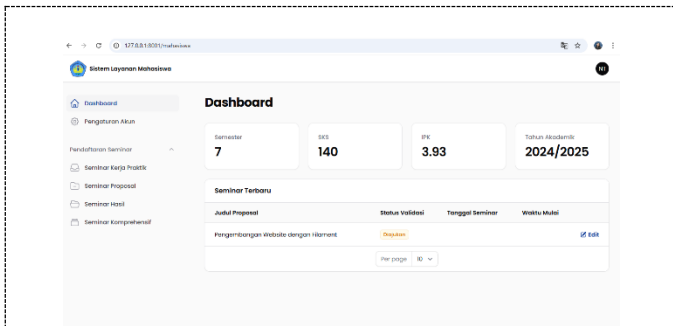
Gambar 6. Validasi Seminar



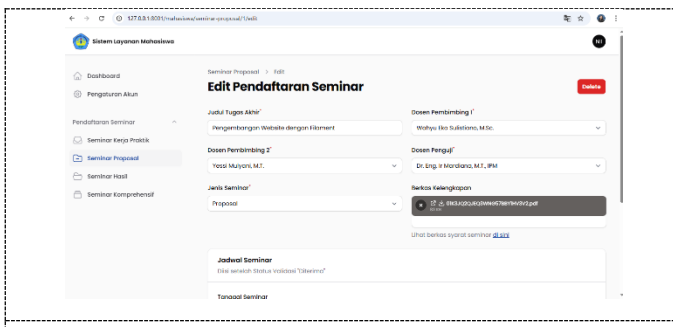
Gambar 9. Halaman Dashboard Kaprodi

2) Role Mahasiswa

Pada role mahasiswa, terdapat tampilan dashboard seperti pada Gambar 8, dan tampilan Edit Pendaftaran Seminar, seperti pada Gambar 9.



Gambar 7. Halaman Dashboard Mahasiswa



Gambar 8. Halaman Edit Pendaftaran Seminar

3) Role Kaprodi

Kaprodi memiliki halaman dashboar seperti pada Gambar 10.

4.3. Cutover

Pada tahap ini, dilakukan pengujian untuk memastikan 276 ystem berjalan sesuai dengan kebutuhan pengguna. Pengujian dilakukan dengan menggunakan metode *blackbox testing* untuk menguji fungsionalitas 276system.

Tabel 1 Ringkasan Pengujian Fitur pada Sistem

No	Tabel Fitur	Test Case	Pass	Fail	% Berhasil	
1.	Login ke dalam 276 ystem (admin)		2	2	0	100%
2.	Melakukan validasi seminar (admin)		2	2	0	100%
3.	Melakukan pendaftaran seminar (admin)		2	2	0	100%
4.	Login ke dalam 276 ystem (kaprodi).		2	2	0	100%
5.	Melakukan validasi seminar (kaprodi).		2	2	0	100%

6.	Login ke dalam sistem (mahasiswa).	2	2	0	100%
7.	Melakukan pendaftaran seminar (mahasiswa).	2	2	0	100%

Seperti terlihat pada Tabel 1., pengujian terhadap 7 fitur yang dimiliki oleh role Admin, Kaprodi dan Mahasiswa menghasilkan total 14 *test case*. Seluruh *test case* berhasil dijalankan tanpa adanya kegagalan, sehingga menunjukkan bahwa semua fitur telah berfungsi sesuai dengan yang diharapkan dengan tingkat keberhasilan 100%.

4.4. Evaluasi

A. Pengukuran Produktivitas

Terdapat beberapa langkah yang perlu dilakukan, yaitu:

1. Menghitung jumlah function points pada fitur Login dan Pendaftaran Seminar.

Function Points dapat diukur setelah menentukan nilai CFP dan RCAF. Berdasarkan perhitungan yang sudah dilakukan, didapatkan hasil pada Tabel 2.

Tabel 2. Jumlah Function Points pada Tiap Fitur

No	Fitur	Jumlah Function Points
1.	Login	16.15
2.	Pendaftaran Seminar	44.65

2. Mencatat waktu yang dibutuhkan dalam pengembangan fitur.
3. Menghitung functional productivity pada tiap fitur.

B. Perbandingan Hasil Implementasi

Berdasarkan pengembangan fitur Login dengan menggunakan admin panel package dan tanpa admin panel package Filament, didapatkan hasil pada Tabel 3., sementara itu untuk fitur Pendaftaran Seminar bisa dilihat di Tabel 4.

Tabel 3 Hasil Perbandingan Fitur Login

Aspek	Dengan Filament	Tanpa Filament
Waktu Pengembangan	4 jam 34 menit	7 jam 43 menit
Tampilan UI	Blade disediakan bawaan Filament dan responsif.	Blade dituliskan secara manual

		oleh pengembang.
Autentikasi	Menambahkan fungsi <code>canAccessPanel()</code> pada file <code>models > User.php</code> untuk hak akses panel dan menambahkan id pada tiap panel provider.	Membuat file <code>controller</code> dan <code>middleware</code> untuk konfigurasi <code>direct</code> halaman setelah login dan hak akses panel.
Routing	Tersedia pada bawaan filament.	Perlu didefinisikan pada file <code>web > route.php</code>
Komponen (Formulir)	Sudah tersedia form builder bawaan filament.	Kode form dituliskan secara manual.
Widgets Dashboard	Terdapat fitur <code>generate widgets</code> dan perlu menambahkan data yang akan ditampilkan.	Pembuatan <code>widgets</code> dilakukan secara manual termasuk kode <code>blade</code> dan data yang akan ditampilkan.

Tabel 4. Hasil Perbandingan pada Fitur Pendaftaran Seminar

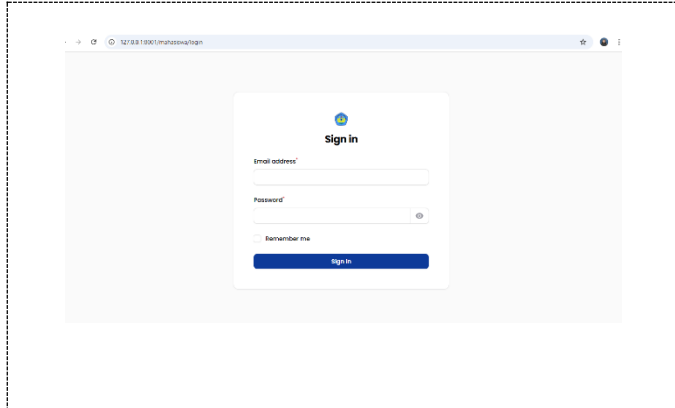
Aspek	Dengan Filament	Tanpa Filament
Waktu Pengembangan	4 jam 15 menit	8 jam 51 menit
Routing	Tersedia pada bawaan filament.	Perlu didefinisikan pada file <code>webphp</code>
CRUD (create, read, update, delete)	Fitur CRUD tersedia dalam satu folder <code>resource</code> . Pengembangan hanya perlu menambahkan <code>custom action</code> , <code>query data</code> dan konfigurasi halaman.	Membuat file <code>controller.php</code> untuk membuat fungsi yang berisikan <code>action</code> serta <code>query</code> yang digunakan, membuat file <code>blade.php</code> untuk membuat <code>view</code> halaman pada fitur Pendaftaran seminar, membuat file <code>routes.php</code> untuk konfigurasi rute pada tiap halaman fitur Pendaftaran seminar.

Komponen (Formulir, breadcrumbs, modal, table)	Tersedia form builder, builder, breadcrumbs, dan modal bawaan filament.	Kode formulir, breadcrumbs, modal dan tabel dituliskan secara manual.
--	---	---

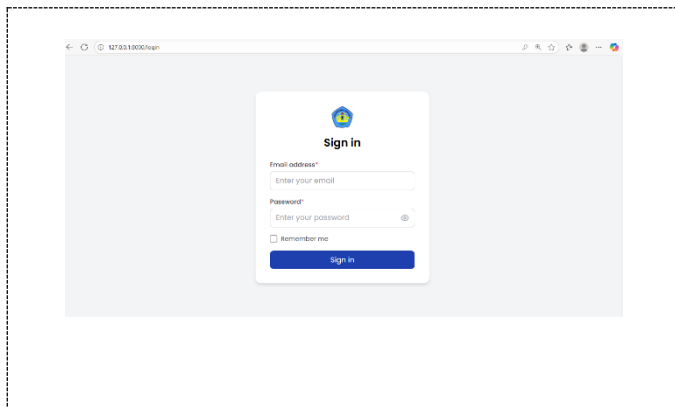
Berikut ini perbandingan untuk tampilan pengguna yang dikembangkan dengan Filament dan tanpa Filament

1) Fitur Login

Perbandingan tampilan bisa dilihat pada Gambar 11 dan Gambar 12.



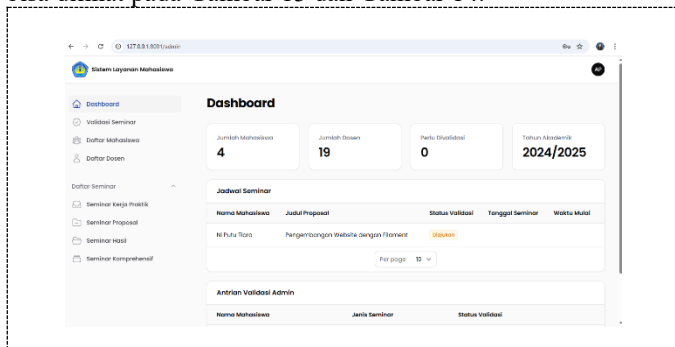
Gambar 10. Halaman Login dengan Filament



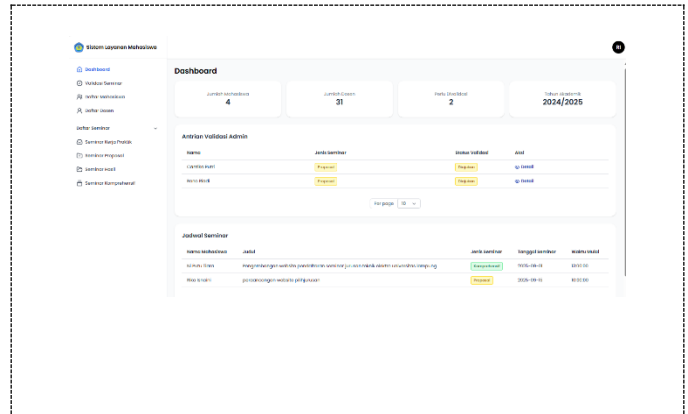
Gambar 11. Halaman Login dengan Tanpa Filament

2) Fitur Pendaftaran Seminar

Perbandingan tampilan fitur Pendaftaran Seminar bisa dilihat pada Gambar 13 dan Gambar 14.



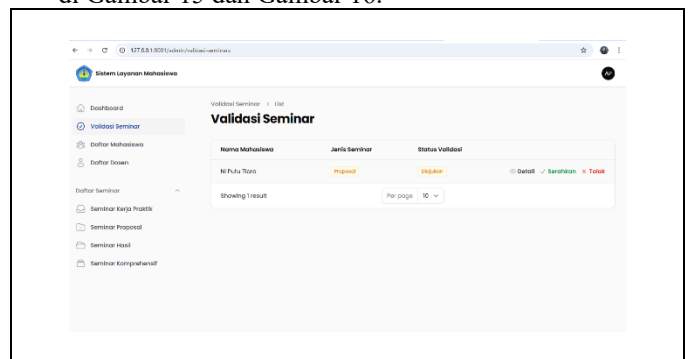
Gambar 12. Halaman Dashboard Admin dengan Filament



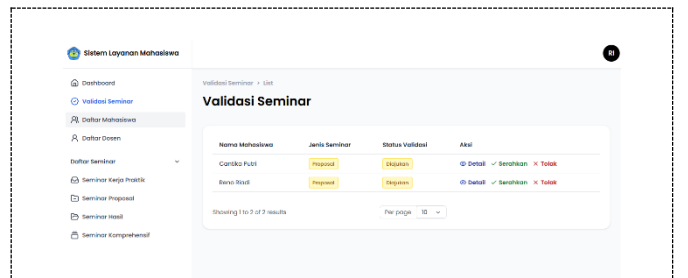
Gambar 13. Halaman Dashboard Admin Tanpa Filament

3) Fitur Validasi Seminar

Perbandingan tampilan fitur Validasi Seminar bisa dilihat di Gambar 15 dan Gambar 16.



Gambar 14. Halaman Validasi Seminar dengan Filament



Gambar 15. Halaman Validasi Seminar Tanpa Filament

5. Kesimpulan

Penelitian ini telah menerapkan Framework laravel yang terintegrasi dengan admin panel package dalam pengembangan fitur Login dan Pendaftaran Seminar pada Sistem Layanan Mahasiswa sesuai dengan kebutuhan bisnis yang telah diidentifikasi sebelumnya. Sistem berhasil dikembangkan dengan berbagai macam fitur yang dapat memudahkan proses administrasi di lingkungan Jurusan.

Pengujian fungsionalitas sistem telah dilakukan menggunakan blackbox testing. Didapatkan hasil persentase skor pengujian sebesar 100% yang berarti

sistem berhasil melewati seluruh *test case* dengan nilai batas yang telah diberikan sebelumnya serta menunjukkan *output* yang sesuai dengan kebutuhan sistem.

Berdasarkan pengukuran *functional productivity* yang telah dilakukan, penerapan *admin panel package* dalam pengembangan fitur *Login* memiliki nilai produktivitas yang lebih besar, yakni 3.5 dengan waktu pengembangan 4.56 jam dibandingkan tanpa menerapkan *admin panel package* dengan nilai 2.1 dan waktu pengembangannya 7.74 jam. Lalu, pada pengembangan fitur Pendaftaran Seminar, penerapan *admin panel package* juga menghasilkan nilai produktivitas yang lebih besar, yakni 10.5 dengan waktu pengembangan 4.25 jam dibandingkan tanpa *admin panel package* yang menghasilkan nilai 5.1 dan waktu pengembangannya 8.84 jam. Dengan begitu, dapat disimpulkan bahwa penerapan *admin panel package* yang terintegrasi pada *framework* Laravel mempercepat waktu pengembangan sehingga produktivitas tim pengembang meningkat.

Daftar Pustaka

- Banker, R. D., and Kauffman, R. J. (1991) "Reuse and Productivity in Integrated Computer-Aided Software Engineering: An Empirical Study," *MIS Q.*, vol. 15, no. 3, pp. 375–401.
- Card, D. N. (2006) "The Challenge of Productivity Measurement," *Proc. Pac. Northwest Softw. Qual. Conf.*
- Cheng, L. *et al.*, (2022) "What improves developer productivity at google? code quality," in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, Singapore Singapore: ACM, , pp. 1302–1313.
- Palomba, F., Bavota, G., Di Penta, M., Fasano, F., Oliveto, R., and De Lucia, A. (2018) "On the diffuseness and the impact on maintainability of code smells: a large scale empirical investigation," in *Proceedings of the 40th International Conference on Software Engineering*, in ICSE '18. New York, NY, USA: Association for Computing Machinery, p. 482.
- Shelly, G. B., Rosenblatt, H. J. (2012) *Systems Analysis and Design*, 9th ed, Course Technology Cengage Learning, Boston
- Wijanarko, S. (2025) "Implementasi Filament Form Builder Dalam Sistem Informasi Berbasis Laravel," *J. Tek. Inform. STMIK Antar Bangsa*, vol. 11, no. 1, pp. 14–17