

Integrasi Role-Based Access Control Menggunakan Filament Shield Pada Sistem Informasi Pengelolaan Tugas Akhir

Riswanda Al Farisi¹, Afta Ramadhan Zayn², Benni Agung Nugroho³, Agustono Heriadi⁴, Novita Dewi Susanti⁵

^{1,2,3,4,5}Manajemen Informatika, Politeknik Negeri Malang

Email: ¹riswanda.al@polinema.ac.id, ²afta.ramadhan@polinema.ac.id, ³benni.agung@polinema.ac.id,

⁴agustono.heriadi@polinema.ac.id, ⁵novita.dewi@polinema.ac.id

Email Penulis Korespondensi: riswanda.al@polinema.ac.id

Abstrak

Sistem informasi pengelolaan tugas akhir dikembangkan sebagai solusi digital untuk meningkatkan efisiensi proses administrasi akademik serta mempermudah koordinasi antara admin, dosen, dan mahasiswa. Kompleksitas sistem ini menuntut adanya mekanisme pengaturan hak akses yang jelas agar setiap pengguna hanya dapat menjalankan fungsi sesuai perannya. Untuk menjawab kebutuhan tersebut, penelitian ini mengimplementasikan kontrol akses berbasis peran, yaitu *Role-Based Access Control* (RBAC) menggunakan package Filament Shield pada *framework* Laravel. Penggunaan Filament Shield memberikan fleksibilitas tinggi dalam pengelolaan peran dan izin, serta mampu menyesuaikan struktur akses yang kompleks sesuai kebutuhan sistem. Proses pengujian dilakukan dengan metode *Black Box Testing* menggunakan pendekatan *Functional Testing* guna memastikan seluruh fungsi sistem berjalan sesuai dengan hak akses masing-masing pengguna. Hasil pengujian menunjukkan bahwa sistem telah berfungsi sesuai dengan rancangan dan mampu memberikan output yang sesuai dengan skenario pengujian. RBAC yang terintegrasi melalui Filament Shield terbukti efektif dalam membatasi akses pengguna, menjaga keamanan data, dan meningkatkan efisiensi pengelolaan sistem. Secara keseluruhan, sistem informasi tugas akhir ini dapat diimplementasikan dengan baik untuk mendukung kegiatan akademik dan meningkatkan kualitas manajemen administrasi di lingkungan program studi.

Kata Kunci: Black Box Testing, Filament Shield, Laravel, Sistem Informasi, Tugas Akhir, Role-Based Access Control (RBAC)

Abstract

The final project management information system was developed as a digital solution to enhance administrative efficiency and streamline coordination among administrators, lecturers, and students. The system's complexity requires a well-defined access control mechanism to ensure that each user operates only within their assigned role. To address this need, the study implements a role-based access control (RBAC) mechanism using the Filament Shield package in the Laravel framework. Filament Shield provides high flexibility in managing roles and permissions while supporting complex access structures based on system requirements. The testing process was conducted using the Black Box Testing method with a Functional Testing approach to ensure that all system functions operate according to each user's access rights. The test results show that the system performs as designed and produces outputs consistent with the defined scenarios. The integration of RBAC through Filament Shield effectively restricts user access, enhances data security, and improves the efficiency of user management. Overall, this final project management information system can be well implemented to support academic activities and improve administrative management quality within the study program.

Keywords: Black Box Testing, Filament Shield, Laravel, Information System, Final Project, Role-Based Access Control (RBAC)

1. PENDAHULUAN

Perkembangan teknologi informasi telah memberikan dampak signifikan terhadap berbagai aspek kehidupan masyarakat, terutama dalam bidang pendidikan tinggi. Digitalisasi layanan akademik di perguruan tinggi menjadi langkah strategis dalam meningkatkan efisiensi, transparansi, serta akurasi proses administrasi akademik [1]. Salah satu implementasinya adalah pengembangan sistem informasi akademik yang berfungsi untuk mendukung berbagai kegiatan operasional, mulai dari pengelolaan data mahasiswa, kegiatan perkuliahan, hingga proses tugas akhir. Sistem informasi akademik tidak hanya membantu pihak administrasi, tetapi juga memberikan kemudahan bagi dosen dan mahasiswa dalam mengakses informasi secara cepat dan terintegrasi [2].

Penelitian mengenai pengembangan sistem informasi berbasis web telah dilakukan untuk meningkatkan efektivitas pengelolaan data dan layanan akademik maupun bisnis digital. Penelitian berjudul Sistem Informasi Akademik menggunakan PHP dan MySQL pada Sekolah Tinggi Manajemen Informatika Komputer (STMIK) Indonesia Banda Aceh menunjukkan bahwa penerapan sistem informasi akademik berbasis web mampu membantu proses pengolahan data akademik menjadi lebih terstruktur, cepat, dan efisien [3]. Sementara itu, penelitian pengembangan prototype *e-commerce* bouquet bunga berbasis laravel dengan integrasi rajaongkir dan midtrans menekankan pemanfaatan *framework* Laravel dalam membangun sistem yang modern, terintegrasi, dan mudah dikembangkan [4].

Penelitian sebelumnya yang dilakukan oleh Riswanda Al Farisi dkk telah berhasil mengembangkan sistem informasi akademik pengelolaan tugas akhir menggunakan *framework* Laravel. Sistem tersebut mencakup fitur pengajuan judul, pendaftaran ujian tugas akhir, serta penilaian hasil ujian. Hasil implementasi menunjukkan peningkatan efisiensi dalam

manajemen data dan alur proses tugas akhir. Namun demikian, dari sisi keamanan dan pengelolaan hak akses pengguna, sistem tersebut masih memiliki keterbatasan karena mekanisme pengaturan akses masih dilakukan secara manual tanpa kontrol keamanan yang terstandarisasi. Hal ini berpotensi menimbulkan risiko, seperti tumpang tindih hak akses antar pengguna, serta kemungkinan akses terhadap data yang tidak sesuai dengan peran pengguna. Selain itu, apabila terdapat pengembangan fitur baru pada sistem di masa mendatang, hal tersebut berpotensi menyulitkan proses pengaturan kontrol hak akses pengguna [5].

Sebagai upaya untuk mengatasi permasalahan tersebut, penelitian ini mengusulkan penerapan *Role-Based Access Control* (RBAC) menggunakan *Filament Shield*, yaitu ekstensi dari *Filament PHP* yang menyediakan pengelolaan *role* dan *permission* secara otomatis serta terintegrasi. RBAC berfungsi sebagai mekanisme kontrol akses yang mengatur hak pengguna berdasarkan peran yang dimiliki, sehingga hanya pengguna dengan otorisasi tertentu yang dapat mengakses fitur maupun data tertentu [6] [7]. Penelitian sebelumnya mengenai implementasi sistem keamanan basis data berbasis *Role-Based Access Control* (RBAC) pada Aplikasi *Enterprise Resource Planning* menjelaskan bahwa metode RBAC mampu meningkatkan keamanan sistem melalui pembatasan hak akses pengguna sesuai tugas dan tanggung jawabnya dalam organisasi. Melalui pendekatan tersebut, setiap pengguna hanya diberikan akses terhadap data dan fitur yang relevan dengan perannya, sehingga dapat meminimalkan risiko penyalahgunaan akses serta meningkatkan keamanan dan integritas data pada sistem *Enterprise Resource Planning* (ERP). Hasil penelitian menunjukkan bahwa implementasi RBAC membantu pengelolaan hak akses menjadi lebih terstruktur, efisien, dan mudah dikontrol, khususnya pada sistem yang memiliki banyak pengguna dengan tingkat akses yang berbeda-beda [8].

Oleh karena itu, penerapan RBAC menggunakan *Filament Shield* pada sistem informasi pengelolaan tugas akhir diharapkan dapat meningkatkan keamanan sistem sekaligus mendukung efisiensi penggunaan aplikasi sesuai dengan otorisasi yang dibutuhkan [9]. Penerapan RBAC pada sistem informasi akademik menggunakan *Filament Shield* dilakukan agar lebih mudah dalam proses integrasinya. *Filament Shield* merupakan *plugin* yang menyediakan sistem manajemen akses berbasis peran yang mudah dan intuitif untuk admin panel *Filament*. Secara teknis, *package* ini adalah lapisan antarmuka pengguna (UI) dan alat otomatisasi di atas *package Spatie/Laravel-Permission*. *Package* ini menyediakan tabel basis data (*roles* dan *permissions*) dan *trait* yang diperlukan (*HasRoles*) untuk menerapkan RBAC di model *user*. *Filament Shield* adalah alat yang memperluas dan mengotomatisasi penggunaan *package Spatie* ini dalam konteks antarmuka *Filament*.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengintegrasikan RBAC menggunakan *Filament Shield* pada sistem informasi akademik pengelolaan tugas akhir, serta menguji efektivitas penerapannya dalam membatasi hak akses pengguna sesuai peran masing-masing. Pengujian dilakukan dengan menggunakan metode *black-box testing* untuk memastikan bahwa sistem dapat mengontrol dan membatasi akses berdasarkan peran pengguna secara tepat. Penelitian ini diharapkan mampu memberikan kontribusi terhadap pengembangan sistem informasi akademik yang lebih aman, efisien, dan terstruktur di lingkungan perguruan tinggi.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Metodologi penelitian merupakan serangkaian langkah sistematis yang dirancang untuk membantu peneliti memperoleh data yang akurat dan relevan, sehingga proses analisis dapat dilakukan secara menyeluruh. Pendekatan ini bertujuan memecahkan masalah penelitian secara terarah dan menghasilkan solusi yang dapat dipertanggungjawabkan. Pada penelitian ini, metodologi digunakan untuk mendukung proses perancangan dan implementasi sistem informasi akademik pengelolaan tugas akhir berbasis *Role-Based Access Control* (RBAC), terutama untuk meningkatkan keamanan akses, efektivitas kerja admin, serta fleksibilitas manajemen pengguna. Tahapan penelitian disusun secara terstruktur, mulai dari pengumpulan data hingga penerapan RBAC pada sistem. Adapun langkah-langkah yang ditempuh adalah sebagai berikut:

a. Teknik Pengumpulan Data

Pengumpulan data dilakukan untuk mendapatkan gambaran yang tepat mengenai kondisi lapangan, kebutuhan sistem, serta bagaimana peran masing-masing pengguna dalam menjalankan proses pengelolaan tugas akhir. Metode yang digunakan adalah:

1. Observasi

Observasi dilakukan dengan cara mengamati secara langsung sistem pengelolaan tugas akhir yang sudah berjalan pada program studi. Melalui pengamatan ini, peneliti memperoleh informasi mengenai:

- Alur kerja sistem saat ini
- Peran dan aktivitas tiap kategori pengguna (mahasiswa, dosen pembimbing, dosen penguji, dan admin)
- Permasalahan yang muncul dalam pengelolaan data dan hak akses
- Mekanisme input, proses, hingga validasi informasi

Hasil observasi menjadi dasar untuk memahami batasan dan kebutuhan sistem baru, khususnya terkait pembagian hak akses dan penerapan RBAC.

2. Wawancara

Wawancara dilakukan untuk menggali informasi lebih mendalam dari pihak-pihak yang terlibat langsung dalam pengelolaan tugas akhir, seperti ketua program studi, admin pengelola tugas akhir, staf akademik yang berhubungan dengan pendataan mahasiswa. Wawancara ini bertujuan untuk mengetahui kendala yang sering muncul dalam proses pengelolaan tugas akhir, mengidentifikasi kebutuhan sistem dari sisi pengguna, memahami kebijakan akademik terkait regulasi pengelolaan tugas akhir, menentukan tingkat akses apa saja yang diperlukan oleh masing-masing peran

b. Studi Literatur

Studi literatur dilakukan untuk memperoleh landasan teori dan referensi ilmiah terkait, konsep dan implementasi *Role-Based Access Control* (RBAC), keamanan sistem informasi berbasis manajemen peran pengguna, praktik terbaik dalam pengembangan sistem akademik, penelitian terdahulu yang relevan mengenai kontrol akses dan manajemen pengguna. Sumber literatur berasal dari jurnal ilmiah, buku, prosiding, serta dokumentasi resmi terkait RBAC. Studi ini menjadi dasar teoritis yang memperkuat rancangan arsitektur sistem, membantu memilih pendekatan implementasi yang sesuai, serta menghindari kesalahan dalam desain hak akses.

c. Penerapan *Role Based Access Control* (RBAC)

Tahap selanjutnya adalah implementasi RBAC pada sistem informasi akademik pengelolaan tugas akhir. RBAC digunakan sebagai metode untuk mengatur hak akses berdasarkan peran yang dimiliki pengguna dalam sistem. Penerapan ini meliputi beberapa langkah meliputi identifikasi peran pengguna, penetapan hak akses setiap peran, implementasi pada sistem, evaluasi dan pengujian sistem.

2.2 Metode Pengembangan Sistem

Metode pengembangan sistem pada penelitian ini menggunakan kerangka kerja Software Development Life Cycle (SDLC) dengan pendekatan *Waterfall*. Pendekatan *Waterfall* dipilih karena menyediakan alur pengerjaan yang runtut dan terdokumentasi dengan baik, sehingga setiap tahap memiliki keluaran yang jelas untuk menjadi dasar tahapan berikutnya [9]. Model ini dinilai sesuai dengan karakteristik penelitian yang berfokus pada penambahan komponen keamanan baru ke dalam sistem yang sudah berjalan, khususnya integrasi mekanisme RBAC melalui Filament Shield pada sistem informasi pengelolaan tugas akhir berbasis Laravel dan Filament. Secara keseluruhan, metode ini terdiri dari beberapa tahapan utama, yaitu analisis kebutuhan, perancangan sistem, implementasi, pengujian, penerapan, dan pemeliharaan.

1. Analisis Kebutuhan

Pada tahap analisis kebutuhan, peneliti melakukan proses identifikasi terhadap struktur pengguna, peran, serta pola hak akses yang telah diterapkan pada sistem sebelumnya. Tujuannya adalah untuk menentukan kebutuhan spesifik yang harus dipenuhi agar integrasi RBAC dapat berjalan secara optimal.

2. Perancangan Sistem

Tahap perancangan sistem dilakukan dengan membangun model otorisasi yang menggambarkan hubungan antara pengguna, peran, dan hak akses menggunakan *use case diagram* serta dokumentasi pendukung lainnya. Selain itu, dilakukan pula perancangan struktur basis data tambahan yang diperlukan untuk menyimpan informasi terkait role, permission, serta relasi di antara keduanya.

3. Implementasi

Tahap implementasi melibatkan proses integrasi package Filament Shield ke dalam proyek Laravel Filament yang sudah ada. Aktivitas utama meliputi instalasi package, pembuatan dan konfigurasi role serta permission, hingga penerapan middleware untuk mengatur kontrol akses berdasarkan peran masing-masing pengguna. Pada tahap ini, peneliti juga melakukan penyesuaian kode pada beberapa modul agar sesuai dengan aturan RBAC yang baru diterapkan.

4. Pengujian

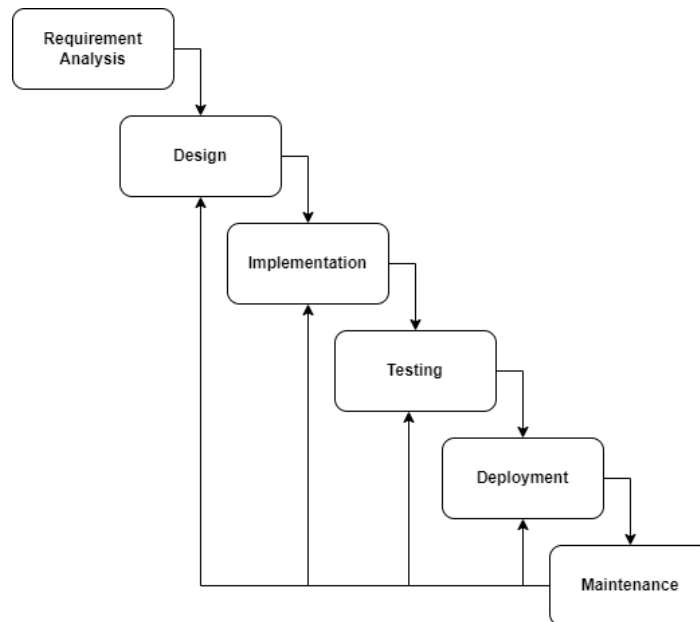
Pengujian dilakukan menggunakan metode *black-box testing* untuk memastikan bahwa setiap fitur kontrol akses berfungsi sesuai dengan aturan yang telah ditetapkan berdasarkan peran pengguna. Pengujian dilakukan pada berbagai skenario penggunaan untuk memastikan bahwa tidak ada pengguna yang dapat mengakses fitur di luar kewenangannya. Selain itu, tahap ini juga bertujuan untuk menemukan potensi bug atau celah keamanan yang mungkin muncul setelah penerapan RBAC. Hasil pengujian menjadi acuan untuk menentukan apakah sistem siap diterapkan atau membutuhkan perbaikan lanjutan.

5. Penerapan

Tahap penerapan (*deployment*) Filament Shield RBAC dilakukan dengan mengaktifkan seluruh konfigurasi kontrol akses berbasis peran ke dalam lingkungan produksi sistem informasi tugas akhir. Pada tahap ini, seluruh komponen RBAC diintegrasikan secara penuh dengan modul yang sudah ada, sehingga sistem dapat beroperasi dengan kontrol akses yang baru.

6. Pemeliharaan

Tahap pemeliharaan (*maintenance*) merupakan proses perbaikan dan penyempurnaan lanjutan setelah sistem diimplementasikan guna memastikan kinerja tetap stabil dan optimal. Pada tahap ini, dilakukan pembaruan terhadap role, permission, maupun konfigurasi keamanan berdasarkan kebutuhan baru yang muncul seiring berjalannya waktu. Selain itu, proses evaluasi terhadap efektivitas penerapan RBAC juga dilakukan untuk menilai dampaknya pada peningkatan keamanan informasi, efisiensi pengelolaan pengguna, dan kemudahan adaptasi terhadap pengembangan fitur tambahan di masa mendatang. Tahap pemeliharaan memastikan sistem tetap relevan dan aman digunakan dalam jangka panjang.



Gambar 1. Metode *Waterfall*

3. HASIL DAN PEMBAHASAN

3.1 UML (*Unified Modeling Language*)

Unified Modeling Language (UML) merupakan sebuah bahasa pemodelan standar yang banyak digunakan di berbagai industri untuk membantu proses analisis, perancangan, dan pendefinisian kebutuhan perangkat lunak. UML berfungsi sebagai media representasi visual yang memudahkan pengembang memahami struktur, perilaku, dan interaksi antar komponen dalam sebuah sistem. Selain itu, UML memiliki kemampuan untuk menggambarkan arsitektur perangkat lunak secara terstruktur, terutama dalam pengembangan aplikasi berbasis objek, sehingga dokumentasi sistem dapat disusun dengan lebih rapi dan mudah dipelihara [10].

Pada penelitian ini, UML digunakan sebagai landasan dalam merancang gambaran sistem secara formal agar setiap fungsi dan alur kerja dapat terdefinisi dengan jelas. Diagram UML membantu memastikan bahwa proses pengembangan mengikuti kebutuhan pengguna secara konsisten dan meminimalkan kesalahan desain. Oleh karena itu, penerapan diagram UML — khususnya *use case diagram* — menjadi langkah penting untuk menggambarkan hubungan dan aktivitas antara sistem informasi pengelolaan tugas akhir dengan berbagai jenis pengguna yang terlibat.

3.1.1 Use Case Diagram

Use Case Diagram merupakan salah satu komponen inti dalam UML yang digunakan untuk menggambarkan bagaimana sebuah sistem berinteraksi dengan aktor yang terlibat. Diagram ini bertujuan menjelaskan fungsi-fungsi apa saja yang disediakan sistem serta bagaimana pengguna menjalankan setiap fungsi tersebut. Dalam konteks penelitian ini, sistem informasi pengelolaan tugas akhir hanya dijelaskan secara singkat sebagai platform akademik yang memfasilitasi pengajuan, penjadwalan, dan penilaian tugas akhir [11].

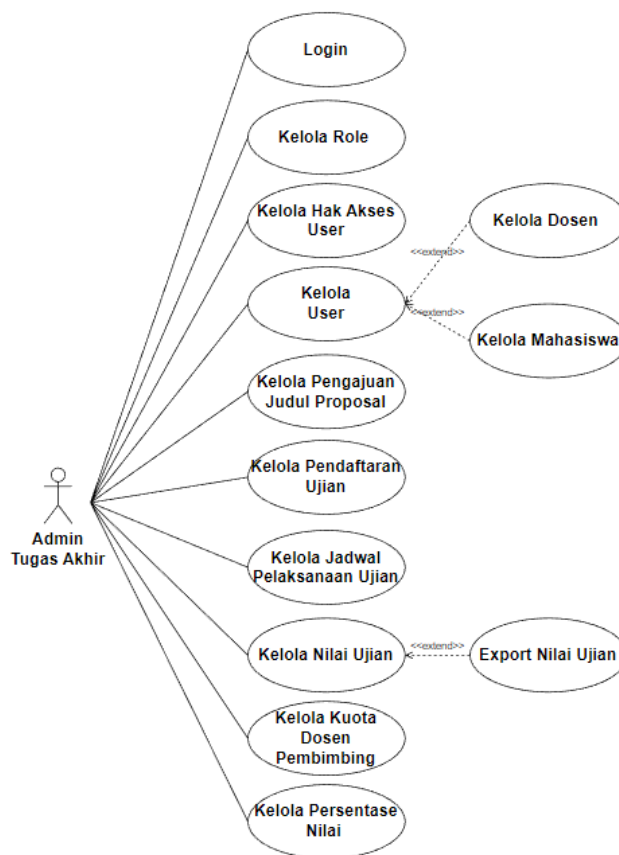
Fokus utama dari use case diagram adalah mendefinisikan hubungan antar aktor dan use case, sehingga struktur kebutuhan fungsional dapat terlihat secara sistematis. Setiap aktor yang terlibat memiliki peran dan batasan akses yang berbeda, dan diagram ini membantu mengidentifikasi alur interaksi tersebut sejak tahap awal pengembangan. *Use case diagram* berfungsi sebagai dokumen dasar yang memandu pengembang dalam menentukan logika proses yang tepat sesuai kebutuhan pengguna. Berdasarkan sistem informasi akademik pengelolaan tugas akhir yang telah dikembangkan sebelumnya, *use case diagram* pengelolaan hak akses disusun berdasarkan tiga kategori peran utama, yaitu Admin Tugas

Akhir, Dosen, dan Mahasiswa. Pembagian ini dilakukan untuk memastikan bahwa setiap peran memiliki akses dan aktivitas yang sesuai dengan tanggung jawabnya.

1. Use Case Admin Tugas Akhir

Admin Tugas Akhir merupakan aktor dengan cakupan akses terluas sehingga memiliki jumlah *use case* paling banyak. Beberapa *use case* utama meliputi:

- a. Mengelola *role* dan hak akses pengguna, termasuk menambah, memperbarui, dan menghapus *role* sesuai kebutuhan sistem.
- b. Mengimpor data pengguna melalui file Excel, di mana admin memproses data dosen dan mahasiswa yang telah ditentukan *role*-nya sebelum masuk ke sistem.
- c. Mengelola proses akademik, seperti memverifikasi pengajuan judul, mengatur pendaftaran ujian, membuat jadwal ujian, serta melakukan pengolahan nilai.
- d. Mengatur kuota dosen pembimbing dan menentukan persentase penilaian, yang penting untuk mengontrol distribusi bimbingan dan standar evaluasi.



Gambar 2. Use Case Role Admin Tugas Akhir

2. Use Case Dosen

Use case pada aktor dosen difokuskan pada aktivitas akademik yang berkaitan dengan bimbingan dan penilaian.

Use case utamanya meliputi:

- a. Memberikan persetujuan atau validasi terhadap pengajuan judul yang diajukan mahasiswa bimbingannya.
- b. Mengisi nilai ujian tugas akhir sesuai perannya sebagai pembimbing atau penguji.
- c. Mengakses daftar mahasiswa bimbingan, baik untuk keperluan monitoring maupun pengelolaan proses akademik lainnya.

Setiap *use case* ini dirancang untuk memastikan bahwa dosen hanya berinteraksi dengan data mahasiswa yang menjadi tanggung jawabnya. Pemodelan ini memperjelas batasan akses dosen agar tidak terjadi pencampuran data antar mahasiswa atau antar dosen yang tidak terkait. *Use case diagram* dosen dapat dilihat seperti pada Gambar 3.



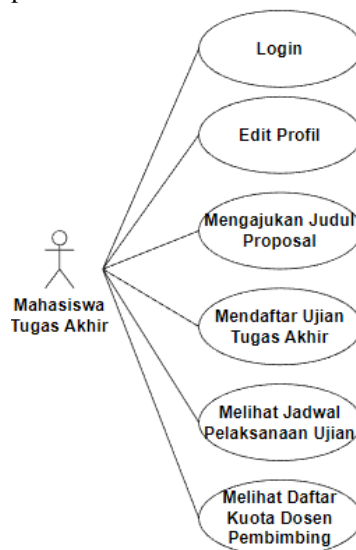
Gambar 3. Use Case Role Dosen

3. Use Case Mahasiswa Tugas Akhir

Aktor mahasiswa memiliki use case yang berfokus pada proses administrasi akademik dan penyelesaian tugas akhir. Use case bagi mahasiswa meliputi:

- a. Mengajukan judul proposal melalui sistem dengan mengikuti format dan ketentuan yang telah ditetapkan.
- b. Melakukan pendaftaran ujian tugas akhir, di mana sistem memvalidasi kelengkapan syarat untuk melanjutkan proses.
- c. Melihat informasi jadwal, status nilai, dan perkembangan bimbingan, sehingga mahasiswa dapat memantau proses akademiknya secara mandiri.

Use case mahasiswa dirancang sebagai serangkaian proses linier yang membantu memastikan bahwa alur penyelesaian tugas akhir berjalan teratur. Dengan pemodelan tersebut, mahasiswa memahami langkah-langkah yang harus dilalui dan sistem dapat memberikan navigasi yang lebih jelas sesuai alur akademik. Use case diagram mahasiswa tugas akhir dapat dilihat pada Gambar 4.



Gambar 4. Use Case Role Tugas akhir

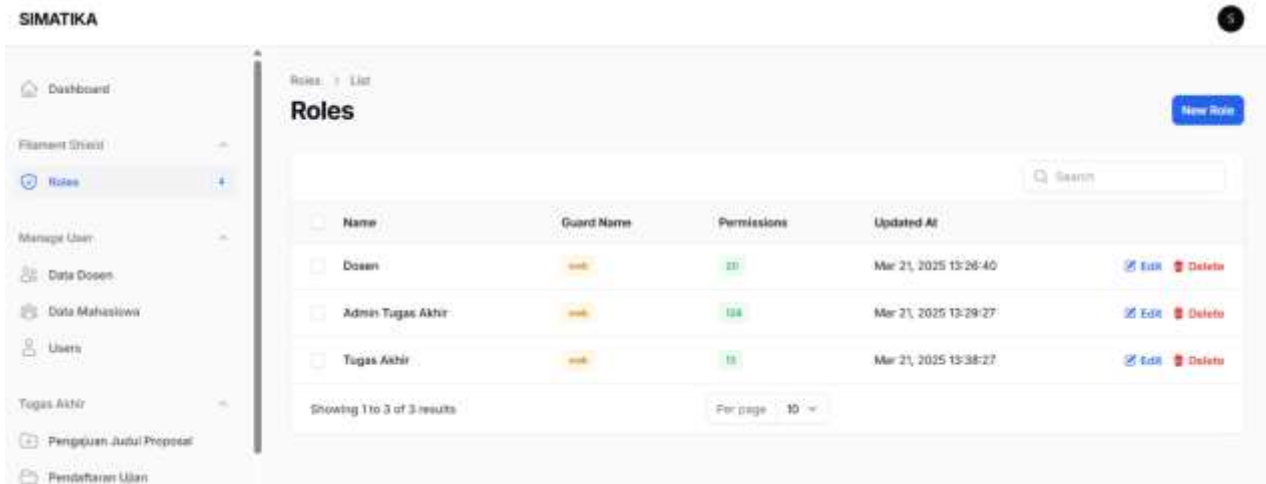
3.2 Implementasi

Implementasi sistem merupakan tahapan krusial di mana desain sistem yang telah dirampungkan diwujudkan. Proses ini melibatkan perubahan rancangan menjadi kode program untuk menghasilkan sistem yang berfungsi dan sesuai

dengan spesifikasi yang direncanakan sebelumnya. Pada penelitian ini tahap implementasi dilakukan dengan melakukan integrasi RBAC menggunakan Filament Shield pada sistem informasi akademik pengelolaan tugas akhir.

b. *Dashboard Roles* Filament Shield

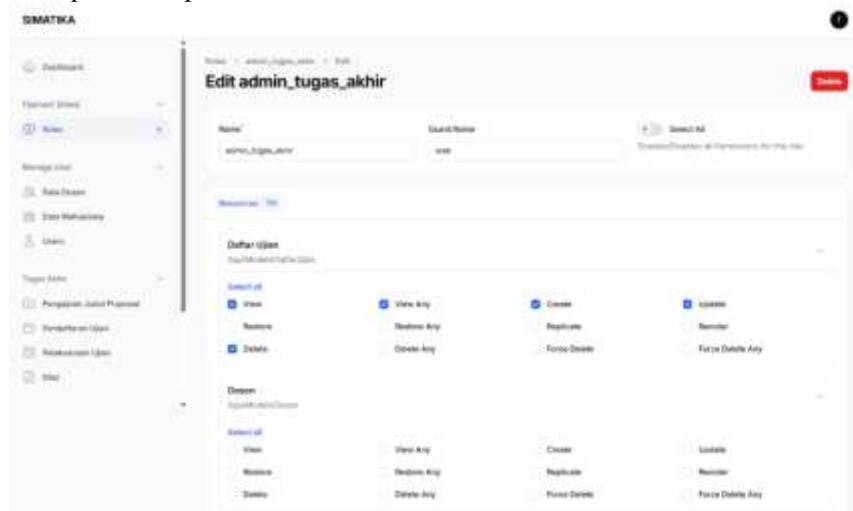
Pada menu ini dapat dilihat daftar *role* yang telah dibuat pada sistem informasi akademik pengelolaan tugas akhir terdiri yang dari beberapa role. Admin dapat menambah, mengubah atau menghapus *role* sesuai kebutuhan. Apabila nanti ada pengembangan sistem lebih lanjut, admin dapat menambahkan role baru sesuai kebutuhan. Pada Gambar 5 dapat dilihat terdapat tiga *role* utama pada sistem ini yaitu: Admin Tugas Akhir, Dosen, dan Mahasiswa Tugas Akhir.



Gambar 5. Daftar Role

c. Hak Akses

Ketika menambahkan sebuah *role* baru, admin bertugas untuk menentukan hak akses pada *role* yang akan dibuat. Hak akses yang disediakan berdasarkan modul-modul yang telah dikembangkan pada sistem sebelumnya, seperti User, Dosen, Mahasiswa, Daftar Ujian, Jadwal Ujian, Judul Proposal, Kuota Dosen, Nilai, Pengaturan Nilai dan Role. Hak akses yang bisa digunakan pada modul-modul tersebut mencakup beberapa hal seperti view, view any, create, update, delete dan lain-lain. RBAC menggunakan Filament Shield sangat memudahkan pengguna dan fleksibel. Hal itu karena role baru bisa ditambahkan, dan menyesuaikan modul yang tersedia atau modul baru. Apabila ada peningkatan penggunaan aplikasi yang lebih kompleks, admin dapat menambahkan *role*, dan memberikan hak akses pada modul-modul baru juga. Contoh pengaturan hak akses sebuah role admin pada modul-modul yang tersedia dapat dilihat pada Gambar 6.

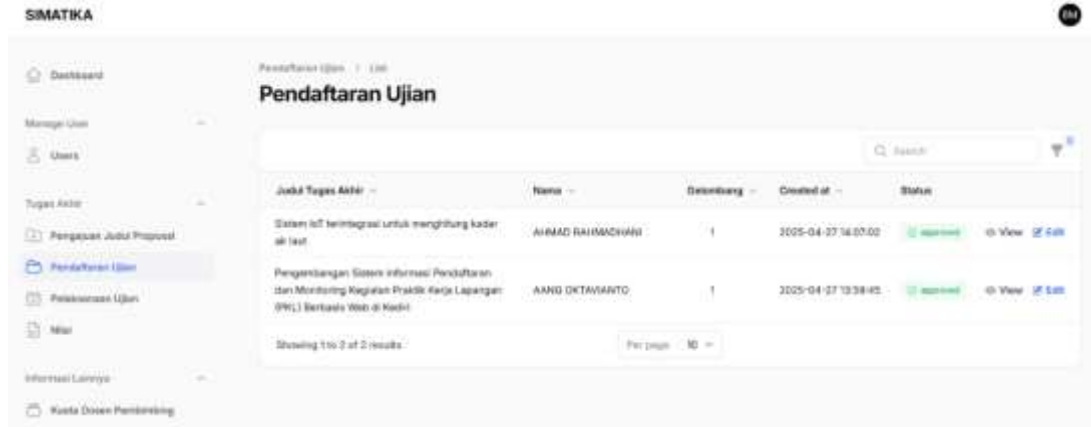


Gambar 6. Pengaturan Hak Akses

d. Dosen

Tampilan sistem untuk dosen menunjukkan bahwa dosen dapat mengakses beberapa menu, yaitu menu users untuk mengubah profil pribadinya. Pada menu pengajuan judul tugas akhir, dosen memiliki peran untuk memvalidasi ajuan

judul proposal yang telah dilakukan mahasiswa, namun hanya dosen pembimbing yang dipilih mahasiswa yang mampu validasi ajuan proposalnya dan dosen berhak menerima atau menolak. Apabila dosen menerima ajuan proposal tersebut, maka akan otomatis mengurangi jumlah kuota bimbingannya. Kemudian dosen juga berhak untuk memvalidasi pendaftaran ujian dari mahasiswa bimbingannya sendiri. Setelah dilakukan validasi, maka admin akan menentukan kapan jadwal pelaksanaan ujian tugas akhir mahasiswa tersebut. Dosen pembimbing, dosen penguji 1 dan dosen penguji 2 memiliki wewenang untuk memberikan nilai pada mahasiswa yang di uji.

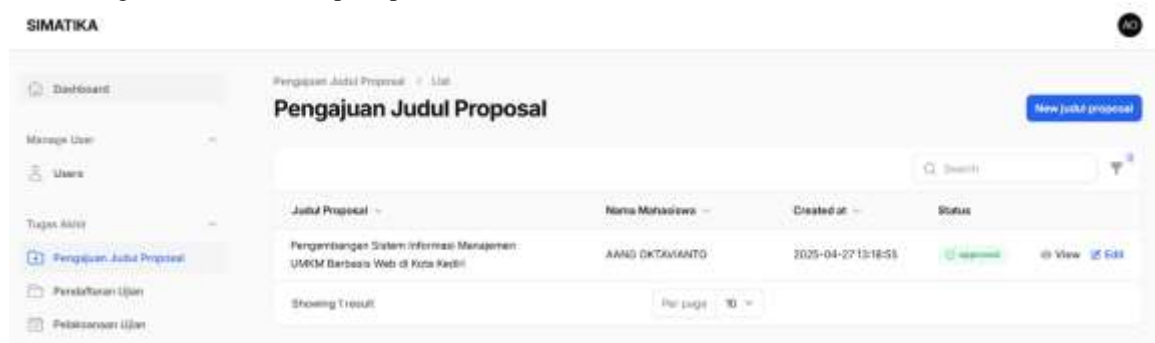


Gambar 5. Tampilan Menu Role Dosen

e. Mahasiswa

Tampilan sistem untuk mahasiswa tugas akhir menunjukkan bahwa mahasiswa tugas akhir memiliki akses ke beberapa menu utama, yaitu mahasiswa dapat mengubah profil pribadinya. Mahasiswa tugas akhir juga memiliki akses untuk mengajukan judul proposal miliknya sendiri dan mengajukan calon dosen pembimbing. Selain itu mahasiswa juga dapat melihat judul proposal mahasiswa yang lain, sehingga mahasiswa juga dapat mendapatkan referensi judul proposal dari beberapa judul yang sudah terdaftar pada sistem. Tujuannya adalah agar tidak ada duplikasi judul antar mahasiswa dan menambah wawasan mahasiswa terkait judul apa yang bisa diambil untuk menyelesaikan tugas akhir.

Mahasiswa tugas akhir yang sudah mendapatkan persetujuan judul proposal dari calon dosen pembimbing, maka mahasiswa tersebut berarti mendapatkan persetujuan menjadi bimbingan dosen tersebut. Mahasiswa juga bisa mengajukan pendaftaran ujian apabila sudah mendapatkan persetujuan dari dosen pembimbing. Apabila pendaftaran ujian mahasiswa tersebut sudah dilakukan validasi oleh dosen pembimbing, maka akan otomatis masuk ke pelaksanaan ujian tugas akhir yang mana jadwalnya ditentukan oleh admin tugas akhir. Pada menu pelaksanaan ujian, mahasiswa dapat cek statusnya setelah pelaksanaan ujian, apakah masih proses menunggu waktu ujian, apakah belum revisi artinya sudah ujian tapi belum mengumpulkan revisi atau statusnya selesai yang menunjukkan bahwa mahasiswa tersebut sudah mengumpulkan revisi yang mana validasi status dilakukan oleh dosen pembimbing mahasiswa tugas akhir tersebut seperti pada Gambar 6.



Gambar 6. Tampilan Menu Role Mahasiswa

3.3 Pengujian

Pengujian dilakukan menggunakan metode *Black Box Testing* dengan pendekatan *functional testing* untuk memastikan setiap fungsi pada sistem berjalan sesuai dengan kebutuhan yang telah ditentukan[12]. Pengujian *black box* merupakan teknik validasi perangkat lunak yang berfokus pada pemeriksaan fungsionalitas aplikasi tanpa perlu memahami struktur internal atau mekanisme kerjanya[13]. Berdasarkan hasil pengujian menggunakan *black box testing* diperoleh hasil yang menunjukkan bahwa semua skenario uji berdasarkan peran dan hak akses yang dilakukan oleh

pengguna berjalan sesuai hasil yang diharapkan. Baik admin tugas akhir, dosen atau mahasiswa memiliki hak akses masing-masing yang mana ada beberapa fitur yang tidak diizinkan oleh sistem sesuai dengan aturan yang telah ditentukan.

Tabel 1. *Black box Testing*

No	Fitur / Modul	Admin Tugas Akhir	Dosen	Mahasiswa Tugas Akhir
1	Login	Login ke sistem sebagai admin	Login ke sistem sebagai dosen	Login ke sistem sebagai mahasiswa
2	Edit Profil	Mengubah data profil pribadi	Mengubah data profil pribadi	Mengubah data profil pribadi
3	Kelola Role	Menambah, mengubah, dan menghapus role pengguna	Tidak Diizinkan	Tidak Diizinkan
4	Kelola Hak Akses User	Mengatur hak akses tiap user (admin, dosen, mahasiswa)	Tidak Diizinkan	Tidak Diizinkan
5	Kelola User	Menambah, mengedit, dan menghapus data user	Tidak Diizinkan	Tidak Diizinkan
6	Kelola Dosen	Mengelola data dosen	Tidak Diizinkan	Tidak Diizinkan
7	Kelola Mahasiswa	Mengelola data mahasiswa	Tidak Diizinkan	Tidak Diizinkan
8	Kelola Pengajuan Judul Proposal	Menambah, mengedit, dan menghapus pengajuan judul proposal	Memvalidasi pengajuan judul dari mahasiswa bimbingan	Mengajukan judul proposal
9	Kelola Pendaftaran Ujian	Menambah, mengedit, dan menghapus pendaftaran ujian	Memvalidasi pendaftaran ujian mahasiswa bimbingan/penguji	Mendaftar ujian tugas akhir
10	Kelola Jadwal Pelaksanaan Ujian	Menambah, mengubah, menghapus dan mengatur jadwal ujian	Melihat dan validasi jadwal ujian yang diampu	Melihat jadwal pelaksanaan ujian
11	Kelola Nilai Ujian	Melihat dan merekap nilai ujian mahasiswa	Menginput nilai ujian sesuai peran (pembimbing/penguji)	Tidak Diizinkan
12	Export Nilai Ujian	Mengekspor nilai ujian ke format Excel atau PDF	Tidak Diizinkan	Tidak Diizinkan
13	Kelola Kuota Dosen Pembimbing	Menentukan dan mengelola kuota dosen pembimbing	Melihat daftar kuota dosen pembimbing	Melihat daftar kuota pembimbing sebelum memilih dosen
14	Kelola Persentase Nilai	Menentukan bobot persentase nilai pembimbing dan penguji	Tidak Diizinkan	Tidak Diizinkan

4. KESIMPULAN

Berdasarkan hasil perancangan dan pengujian menggunakan metode *Black Box Testing* dengan pendekatan *functional testing*, sistem informasi pengelolaan tugas akhir yang dibangun telah berfungsi sesuai dengan kebutuhan pengguna. Setiap fitur utama, seperti pengelolaan judul proposal, pendaftaran ujian, penjadwalan ujian, penginputan nilai, serta pengaturan kuota dosen pembimbing, berhasil diuji dengan hasil yang sesuai dengan skenario yang diharapkan. Hasil pengujian menunjukkan bahwa sistem mampu mengakomodasi seluruh peran pengguna yaitu admin tugas akhir, dosen, dan mahasiswa sesuai hak aksesnya masing-masing. Admin dapat mengelola data dan pengaturan sistem secara menyeluruh, dosen dapat melakukan validasi dan penilaian ujian mahasiswa, sementara mahasiswa dapat mengajukan judul, mendaftar ujian, dan melihat informasi jadwal serta kuota pembimbing.

Secara keseluruhan, sistem telah berjalan dengan stabil dan menampilkan keluaran yang sesuai pada setiap pengujian fungsi. Penggunaan metode *Black Box Testing* membantu memastikan bahwa setiap proses di sisi pengguna dapat beroperasi dengan baik, tanpa perlu meninjau kode program secara langsung. Hasil ini menunjukkan bahwa rancangan *use case* yang dibuat telah terimplementasi dengan baik pada aplikasi, sehingga sistem informasi tugas akhir dengan menerapkan RBAC menggunakan Filament Shield ini layak untuk digunakan dalam mendukung kegiatan akademik, khususnya dalam proses administrasi dan monitoring tugas akhir di lingkungan program studi.

UCAPAN TERIMAKASIH

Penulis mengucapkan rasa terima kasih yang mendalam atas pertolongan dan kasih sayang Tuhan Yang Maha Esa, sehingga proses penelitian dan penyusunan laporan ini dapat diselesaikan hingga tahap akhir. Penghargaan yang tulus juga penulis sampaikan kepada semua pihak yang telah memberikan bimbingan, dukungan, serta bantuan dalam berbagai bentuk. Secara khusus, penulis berterima kasih kepada para dosen, keluarga tercinta, dan rekan-rekan yang selalu hadir memberikan dorongan, arahan, serta semangat sepanjang perjalanan penyusunan penelitian ini.

DAFTAR PUSTAKA

- [1] M. Afiani and A. Muzija, "Perancangan Aplikasi Perpustakaan Berbasis Web Menggunakan Metode Design Thinking (Studi Kasus: SMPN 2 Kesesi)," *Djtechno : Jurnal Teknologi Informasi*, vol. 6, no. 2, 2025, doi: 10.46576/djtechno.
- [2] N. Zakiyah, S. Firdaus, M. Arief Nugroho, Y. Biantoro, and D. Prasetya, "SIPAS: Sistem Informasi Pengolahan Survei Untuk Optimalisasi Manajemen Data Survei Hidrografi," *Jurnal Sistem Informasi TGD*, vol. 4, no. 5, pp. 1151–1161, 2025, [Online]. Available: <https://ojs.trigunadharma.ac.id/index.php/jsi>
- [3] A. Latifurrahman, Imilda, and A. Salam, "Sistem Informasi Akademik menggunakan PHP dan MySQL pada Sekolah Tinggi Manajemen Informatika Komputer (STMIK) Indonesia Banda Aceh," *Jurnal Sistem Komputer (SISKOM)*, vol. 3, no. 2, pp. 74–83, Aug. 2023, doi: 10.35870/siskom.v3i2.796.
- [4] A. Handayani and S. Muzid, "Pengembangan Prototype E-commerce Bouquet Bunga Berbasis Laravel dengan Integrasi Rajaongkir dan Midtrans," *JURNAL SISTEM INFORMASI TGD*, vol. 4, no. 5, pp. 1162–1173, 2025, [Online]. Available: <https://ojs.trigunadharma.ac.id/index.php/jsi>
- [5] R. Al Farisi, A. Ramadhan Zayn, B. Agung Nugroho, A. Heriadi, M. anajemen Informatika, and P. M. Negeri alang, "Implementasi Sistem Informasi Akademik Pengelolaan Tugas Akhir Berbasis Laravel dan Filament," *Jurnal Sistem Informasi TGD*, vol. 4, pp. 486–496, 2025, [Online]. Available: <https://ojs.trigunadharma.ac.id/index.php/jsi>
- [6] Y. Yuricha and I. K. Phan, "Penerapan Role Based Access Control dalam Sistem Supply Chain Management Berbasis Cloud," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 3, no. 2, pp. 339–348, Nov. 2023, doi: 10.57152/malcom.v3i2.1259.
- [7] A. Muliani Harahap, "Implementasi Metode Role-Based Access Control Pada Aplikasi E-Raport di MIN 15 Langkat Berbasis Android," *Jurnal Teknik Informatika Unika ST. Thomas*, vol. 09, no. 01, p. 6615683, Jun. 2024.
- [8] M Sahyudi and Erliyan Redy Susanto, "Analisis Implementasi Sistem Keamanan Basis Data Berbasis Role-Based Access Control (RBAC) pada Aplikasi Enterprise Resource Planning," *SATESI: Jurnal Sains Teknologi dan Sistem Informasi*, vol. 5, no. 1, pp. 105–116, Apr. 2025, doi: 10.54259/satesi.v5i1.3997.
- [9] Rubiyanto, Selo, and Widyawan, "Implementasi Role-Based Access Control (RBAC) Pada Pemanfaatan Data Kependudukan Ditingkat Kabupaten," in *Seminar Nasional Sains dan Teknologi*, Nov. 2017.
- [10] A. Husna Ahadi and T. Djauhari, "Pengembangan Sistem Informasi Manajemen Hotel Multi-Cabang Berbasis Web Dengan Framework Laravel," *Djtechno : Jurnal Teknologi Informasi*, vol. 6, no. 2, 2025, doi: 10.46576/djtechno.
- [11] F. Yudianto, M. Ibadurahman, and L. Yuliana, "Rancang Bangun Sistem Pengajuan Kegiatan Ormawa Dengan Menggunakan Metode Prototyping," *Djtechno : Jurnal Teknologi Informasi*, vol. 6, no. 2, 2025, doi: 10.46576/djtechno.
- [12] A. Pradana Putra, F. Andriyanto, T. Dewi Muji Harti, and W. Puspitasari, "Pengujian Aplikasi Point Of Sale Berbasis Web Menggunakan Black Box Testing," *Jurnal Bina Komputer*, vol. 2, Feb. 2020.
- [13] Muhammad Helmi Satria Fedianto, Firza Prima Aditiawan, and Muhammad Muharrom Al Haromainy, "Pengujian Sistem Jaringan Dokumentasi Dan Informasi Menggunakan Black Box Testing Dan White Box Testing," *Jurnal Publikasi Sistem Informasi dan Manajemen Bisnis (JUPSIM)*, vol. 3, no. 1, pp. 213–221, Dec. 2024, doi: 10.55606/jupsim.v3i1.2447.