

# Sistem Informasi Pengajuan Proposal Tugas Akhir Berbasis Web dengan RBAC Multi-Tenant dan Deteksi Kemiripan Judul Menggunakan Metode Agile Scrum

Duandi<sup>1\*</sup>, Ilwan Syafrinal<sup>2</sup>

<sup>1,2</sup> Program Studi Teknik Perangkat Lunak, Fakultas Komputer, Universitas Universal

\*Corresponding author E-mail: [duandiduan22@gmail.com](mailto:duandiduan22@gmail.com)

## Article Info

### Article history:

Received 02-07-2026

Revised 18-07-2026

Accepted 27-07-2026

### Keyword:

Proposal Submission, Agile Scrum, Role-Based Access Control, Independent Assessment, Digital Rubric

## ABSTRACT

*The submission and evaluation of final project proposals constitute a vital administrative stage in higher education. At Universitas Universal, this process previously relied on semi-manual methods via Google Drive and spreadsheets, which led to data privacy vulnerabilities, examiner evaluation bias, and inefficient data recapitulation. To address these challenges, this study developed a centralized, web-based proposal submission and monitoring information system. Built using Laravel and MySQL databases, the software development process was managed through the Agile Scrum framework across four Sprints. The system incorporates Role-Based Access Control (RBAC) to enforce multi-tenant document security, an asynchronous similarity check using a recursive common substring match to prevent title duplication, and an independent grading module using digital rubrics to ensure objectivity. Software quality evaluations demonstrated high reliability: functional black-box testing achieved a 100% success rate, PHPUnit testing (183 test cases, 611 assertions) passed without errors, and white-box testing showed an overall statement code coverage of 76.9% (exceeding 87% in critical business logic and database model layers). The system successfully digitizes, secures, and standardizes the academic proposal workflow.*



Copyright © 2026. This is an open access article under the [CC BY](https://creativecommons.org/licenses/by/4.0/) license.

## I. PENDAHULUAN

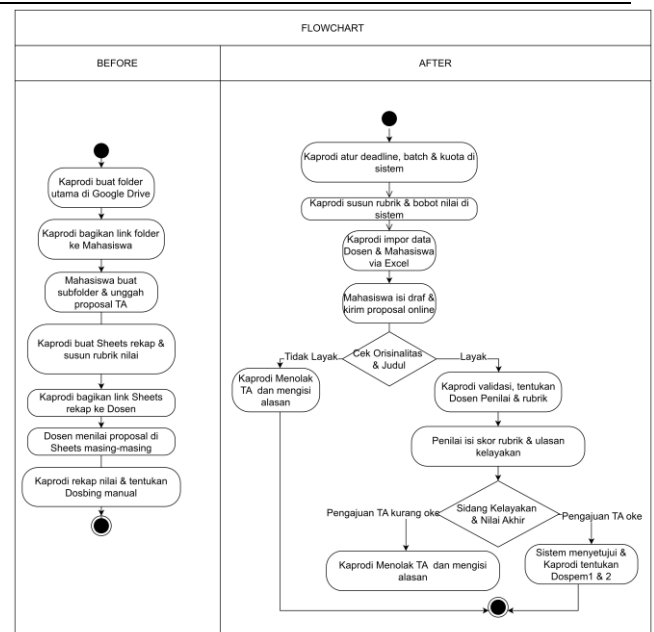
Memasuki era Revolusi Industri 4.0 dan Society 5.0, perguruan tinggi dihadapkan pada tuntutan akselerasi transformasi digital. Integrasi teknologi informasi ke dalam tata kelola perguruan tinggi (*Smart Campus*) diposisikan sebagai langkah strategis untuk meningkatkan efisiensi layanan administrasi akademik, transparansi proses, serta efisiensi beban kerja dosen dan mahasiswa [1], [2], [3]. Salah satu proses administratif yang paling krusial bagi kelangsungan studi mahasiswa adalah pengajuan dan penilaian proposal tugas akhir [4], [5]. Proses ini melibatkan alur koordinasi berlapis yang kompleks di mana mahasiswa mendaftarkan draf penelitiannya, dosen pembimbing mengarahkan, dosen penguji melakukan evaluasi secara formal, dan koordinator program studi (Kaprodin) mengesahkan kelayakan kelulusan topik tugas akhir tersebut

[6], [7]. Namun, pada Program Studi Teknik Perangkat Lunak Universitas Universal, pelaksanaan alur kerja ini masih bersifat semi-manual dengan menggunakan folder *Google Drive* bersama dan *spreadsheet* [8]. Ketergantungan terhadap infrastruktur *ad-hoc* tersebut melahirkan empat kategori masalah utama yang berdampak pada kualitas layanan akademis.

Permasalahan pertama berkaitan dengan tidak tersedianya alur kerja (*workflow*) terstruktur dan mekanisme pemantauan status proposal. Mahasiswa tidak dapat melacak secara *real-time* apakah dokumen mereka sedang direview, direvisi, atau disetujui, sehingga setiap tahapan pengajuan, pemeriksaan, hingga persetujuan proposal harus dipantau melalui pengecekan manual dan komunikasi terpisah [8]. Masalah kedua yang sangat kritis adalah celah keamanan privasi data dokumen. Penggunaan folder *Google Drive* bersama tanpa mekanisme pembatasan kontrol akses yang memadai membuat dokumen proposal yang diunggah mahasiswa dapat

diakses, disalin, atau bahkan dihapus secara tidak sengaja oleh mahasiswa lain. Kondisi ini menciptakan risiko tinggi terhadap pelanggaran privasi dan potensi plagiarisme internal sebelum proposal sempat diujikan secara resmi [9], [10]. Permasalahan ketiga adalah kerentanan terjadinya bias objektivitas dalam evaluasi. Karena *spreadsheet* penilaian bersifat terbuka bagi tim penguji, seorang dosen penilai dapat melihat skor yang diinput oleh dosen lain terlebih dahulu, yang secara tidak sadar memengaruhi keputusan penilaian mereka (*conformity bias*) dan mencederai prinsip penilaian yang independen [11]. Permasalahan keempat adalah ketiadaan sistem pencatatan riwayat revisi dan rekapitulasi data komentar yang terpusat. Proses pembuatan laporan penilaian dilakukan secara manual dengan mengumpulkan komentar dan nilai dari masing-masing dosen yang tersebar di *spreadsheet*, meningkatkan risiko kesalahan input (*human error*), hilangnya data riwayat penilaian, serta menyulitkan proses pelaporan rekapitulasi untuk kebutuhan akreditasi prodi [12].

Sebagai solusi komprehensif atas keterbatasan tersebut, penelitian ini merancang dan membangun sebuah Sistem Informasi Pengajuan Proposal Tugas Akhir berbasis web yang terintegrasi. Sistem ini dibangun menggunakan arsitektur *Model-View-Controller* (MVC) pada *framework* Laravel 12 dan basis data MySQL 8.4 [13] yang dikembangkan melalui kerangka kerja Agile Scrum yang iteratif dan berorientasi pada kebutuhan pengguna [6], [14]. Sistem yang diusulkan menyelesaikan masalah dengan menerapkan: (1) *Role-Based Access Control* (RBAC) untuk mengamankan hak akses file PDF proposal secara *multi-tenant* [9]; (2) Modul penilaian independen dengan rubrik digital yang menyembunyikan nilai dosen penguji lain hingga keputusan final diterbitkan [11]; dan (3) Integrasi API internal untuk deteksi kemiripan judul guna mencegah duplikasi topik penelitian terdahulu secara dini [15]. Pengembangan *platform* terpusat ini memberikan kontribusi ilmiah berupa penggabungan mekanisme deteksi kemiripan asinkron dan penilaian independen terisolasi untuk meningkatkan transparansi, objektivitas, dan keamanan data pada alur kerja akademik perguruan tinggi.



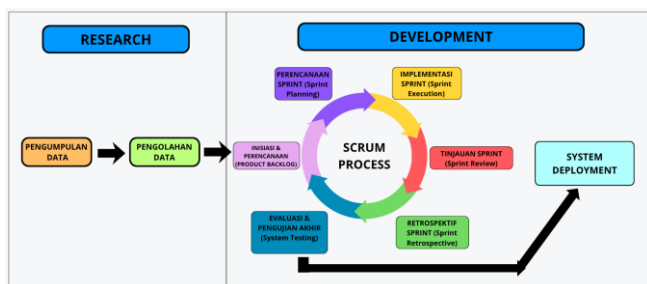
Gambar 1. Perbandingan alur berjalan yang semi-manual dengan alur sistem diusulkan

## II. METODE

Penelitian ini menggunakan metodologi Penelitian dan Pengembangan (*Research and Development* / R&D) dengan tujuan utama merancang, menguji, dan memvalidasi produk sistem informasi akademik digital [11]. Proses rekayasa perangkat lunak dikelola menggunakan kerangka kerja Agile Scrum yang fleksibel terhadap perubahan kebutuhan pengguna melalui siklus iteratif berupa Sprint [6], [16]. Karena penelitian ini diselesaikan oleh seorang peneliti tunggal, dilakukan adaptasi peran Scrum di mana peneliti merangkap peran sebagai *Product Owner* (pemilik visi produk) sekaligus *Development Team* (pengembang sistem), sedangkan Dosen Pembimbing, Dosen Penilai, dan Kaprodi bertindak sebagai pemangku kepentingan (*stakeholder*) yang memberikan umpan balik (*feedback*) pada fase *Sprint Review* di setiap akhir siklus.

### A. Agile Scrum Sebagai Metode Pengembangan

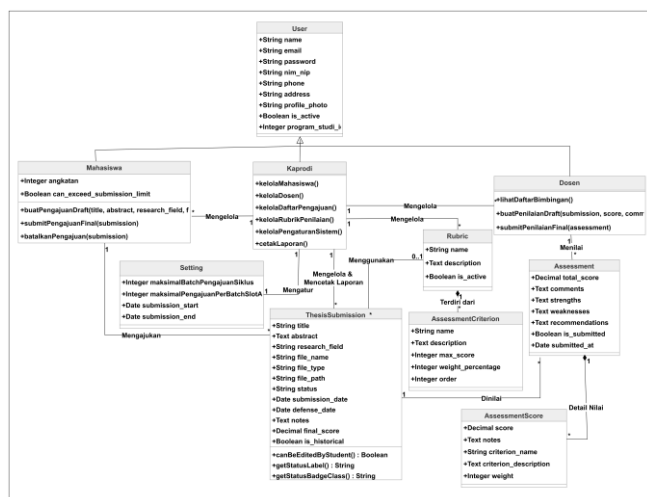
Metode Agile Scrum dipilih karena memiliki keunggulan adaptabilitas tinggi terhadap dinamika kebutuhan pengguna perguruan tinggi [17]. Alur pengembangan dikelola melalui beberapa artefak dan proses: *Product Backlog* (kumpulan kebutuhan fungsional), *Sprint Planning* (perencanaan siklus), *Sprint Execution* (eksekusi pengkodean dan unit testing), *Daily Scrum* (refleksi harian mandiri berupa *self-inspection*), *Sprint Review* (evaluasi produk dengan *stakeholder*), dan *Sprint Retrospective* (evaluasi proses internal). Pengembangan diselesaikan dalam empat siklus Sprint terstruktur dengan durasi masing-masing Sprint berkisar antara 2 minggu, memastikan penambahan fitur berjalan secara bertahap.



Gambar 2. Siklus metodologi Agile Scrum yang terdiri atas empat tahapan Sprint

### B. Arsitektur Sistem dan Relasi Pengguna

Sistem dibangun dengan pola arsitektur *Model-View-Controller* (MVC) menggunakan *framework* Laravel 12 dan RDBMS MySQL 8.4 [13]. Model data memetakan relasi entitas basis data secara terstruktur: model `User` berasosiasi dengan role Spatie Permission, dan memiliki relasi *one-to-many* terhadap `ThesisSubmission` (pengajuan proposal). Setiap pengajuan proposal memiliki relasi *one-to-many* terhadap `SubmissionFile` (berkas biner PDF), `ThesisStatus` (riwayat pelacakan status), dan `Assessment` (penilaian dosen) yang dipetakan menggunakan diagram kelas *Unified Modeling Language* (UML) [18]. Otorisasi keamanan RBAC memetakan peran pengguna secara relasional antara Mahasiswa, Dosen, Kaprodi, dan Admin untuk menegakkan batas akses file PDF secara *multi-tenant* [10].



Gambar 3. Desain ERD relasi tabel pengguna, proposal, status, dan penilaian

### C. Metode Deteksi Kemiripan Judul

Fitur deteksi kemiripan judul dirancang untuk mendeteksi potensi duplikasi judul penelitian terdahulu secara dini pada formulir pengajuan. Sistem mengadopsi algoritma pencocokan string berbasis *Recursive Common Substring* yang diimplementasikan menggunakan fungsi bawaan PHP `similar\_text()` [19]. Sebelum proses komparasi dilakukan,

sistem melakukan normalisasi data dengan mengubah seluruh karakter judul menjadi huruf kecil (*lowercase*). Persentase tingkat kemiripan judul yang diinput dengan judul-judul aktif di database dihitung secara *server-side* menggunakan persamaan (1):

$$Similarity = \frac{2 \times |C|}{|L1| + |L2|} \times 100\% \quad (1)$$

Keterangan variabel pada persamaan (1) didefinisikan sebagai berikut: *Similarity* merepresentasikan nilai persentase tingkat kemiripan antara dua string judul proposal;  $|C|$  merepresentasikan total jumlah karakter identik cocok (*matching characters*) terpanjang yang ditemukan;  $|L1|$  merepresentasikan panjang karakter judul pertama (string input mahasiswa); dan  $|L2|$  merepresentasikan panjang karakter judul kedua (string pembandingan di database).

## III. HASIL DAN PEMBAHASAN

Bagian ini menjabarkan hasil rancang bangun sistem yang dipandu sepenuhnya oleh metodologi Agile Scrum. Pembahasan tidak disajikan sebagai fitur terpisah, melainkan menguraikan bagaimana setiap iterasi Sprint menghasilkan penambahan nilai fungsional yang tervalidasi.

### A. Hasil Pengembangan Berdasarkan Siklus Sprint

Pembangunan sistem pengajuan proposal Tugas Akhir dikelola dalam empat siklus iterasi Sprint. Uraian pelaksanaan Scrum pada masing-masing tahapan dibahas secara rinci di bawah ini.

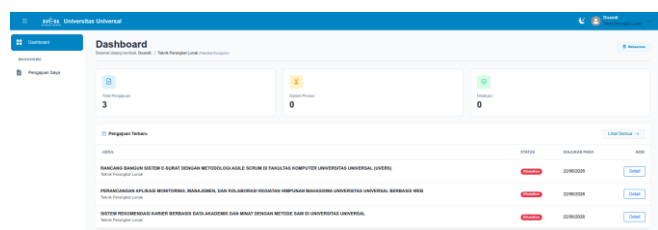
#### 1) Sprint 1: Manajemen Pengguna dan Autentikasi

Sprint pertama difokuskan untuk meletakkan fondasi arsitektur perangkat lunak, skema basis data, dan modul otentikasi. Pelaksanaan *Daily Scrum* diselenggarakan secara mandiri (*self-inspection*) berupa refleksi harian guna mengevaluasi progres konfigurasi Laravel, *setup environment*, dan relasi *role-permission*. Hambatan awal terkait *bottleneck* pemetaan izin Spatie diselesaikan dengan membuat *custom middleware* rute *redirection*. Pada tahap *Sprint Review*, demonstrasi antarmuka login dilakukan kepada pemangku kepentingan yaitu mahasiswa ( $n=5$ ), dosen ( $n=3$ ), dan Kaprodi menggunakan prototipe Figma [20], [21]. Umpan balik yang diterima merekomendasikan perlunya pengamanan tambahan karena form login rentan serangan *brute-force*, yang ditindaklanjuti dengan menyuntikkan pengamanan Captcha. Tahap *Sprint Retrospective* mengevaluasi bahwa pemetaan hak akses user harus tuntas di awal siklus untuk memperlancar otorisasi *multi-role* pada sprint berikutnya. Prioritisasi kebutuhan dilakukan menggunakan metode MoSCoW [22]. Pelaksanaan Sprint 1 secara keseluruhan dirangkum pada Tabel 1.

Tabel 1. Ringkasan Siklus Pengembangan Sprint 1

| Atribut     | Deskripsi / Nilai                                                             |
|-------------|-------------------------------------------------------------------------------|
| Sprint Goal | Membangun sistem otentikasi multi-role aman berbasis RBAC dan dasbor dinamis. |

|                      |                                                                                                                                                                                                                                      |              |                                                                                                                                                             |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Product Backlog Item | Item M-01 (Mekanisme Autentikasi Multi-role & Secure Login)<br>Item S-06 (Kelola Profil Mandiri & Ganti Kata Sandi).                                                                                                                 | Implementasi | Form pengisian proposal, validasi file PDF maks 10MB, filter query batas 3 submit aktif per batch, logic REST API similarity_text(), AJAX checker asinkron. |
| Aktor                | Mahasiswa, Dosen, Kaprodi, Admin                                                                                                                                                                                                     | Output       | Formulir pendaftaran proposal TA, tracking status proposal real-time bagi mahasiswa, popup modal peringatan kecocokan judul serupa >50%.                    |
| Implementasi         | Setup Laravel 12 & MySQL 8.4, pemetaan relasi model Spatie Permission, seeding dummy data akun, pembuatan form login dengan penambahan Google reCAPTCHA v2 (koreksi brute-force), middleware rute redirection, dan form ganti sandi. |              |                                                                                                                                                             |
| Output               | Dasbor multirole (Widget total proposal, grafik status, pintasan cepat), modul login/logout fungsional, secure profile management.                                                                                                   |              |                                                                                                                                                             |



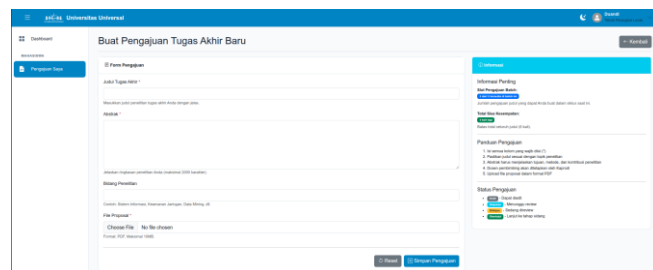
Gambar 4. Tampilan antarmuka dashboard pengguna multirole

## 2) Sprint 2: Pengajuan Proposal Tugas Akhir

Sprint kedua berfokus pada subsistem transaksional pengajuan proposal bagi aktor Mahasiswa. Selama *Daily Scrum* mandiri, evaluasi harian difokuskan pada penanganan validasi berkas PDF dan pembatasan jatah *submit*. Hambatan teknis berupa error pembacaan tipe data file upload diatasi dengan restrukturisasi *request form-data* di Laravel. Pada *Sprint Review*, dilakukan demo pengunggahan proposal, tombol pembatalan draf, dan similarity check judul. Umpan balik *stakeholder* mencatat bahwa dialog *warning* ketika berkas melebihi batas 10MB atau berekstensi non-PDF kurang informatif, sehingga sistem direvisi untuk menyajikan alert modal error validasi yang dinamis. Hasil *Sprint Retrospective* menunjukkan bahwa pengerjaan asinkron similarity check AJAX memerlukan waktu pengujian unit yang lebih intensif, disepakati bahwa *unit testing* pada logic parser string harus dipercepat di awal siklus berikutnya. Ringkasan pelaksanaan Sprint 2 disajikan pada Tabel 2.

Tabel 2. Ringkasan Siklus Pengembangan Sprint 2

| Atribut              | Deskripsi / Nilai                                                                                                                                                      |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Sprint Goal          | Membangun pengajuan proposal terintegrasi dengan validasi PDF, pembatasan jatah submit, dan similarity check judul.                                                    |
| Product Backlog Item | Item M-02 (Unggah draf PDF proposal)<br>Item M-03 (Batas proposal ganda per batch)<br>Item S-01 (Tombol pembatalan draf)<br>Item S-05 (Integrasi API similarity check) |
| Aktor                | Mahasiswa                                                                                                                                                              |



Gambar 5. Tampilan antarmuka form pengajuan mahasiswa dan modal deteksi kemiripan

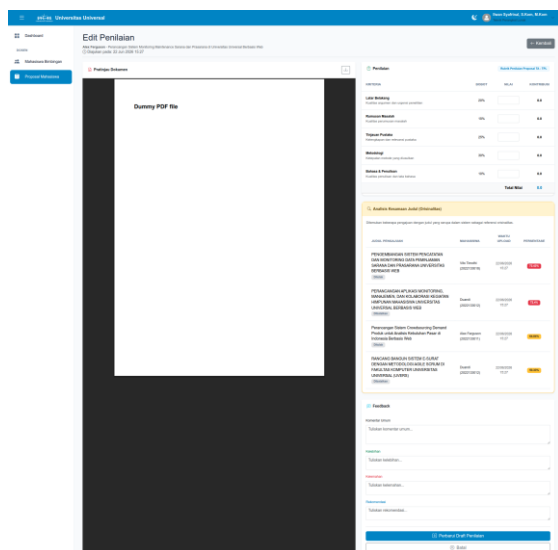
## 3) Sprint 3: Penilaian Proposal oleh Dosen

Sprint ketiga berfokus pada modul penilaian digital bagi aktor Dosen Penilai dan Pembimbing. *Daily Scrum* mandiri dilakukan setiap hari untuk memantau progress form rubrik skor dinamis dan isolasi data nilai. Hambatan yang diselesaikan adalah sinkronisasi relasi database yang menyebabkan query penarikan kriteria lambat, yang dioptimalkan melalui *eager loading* Eloquent. Demonstrasi pada *Sprint Review* mencakup *built-in* PDF viewer dan modul pengisian rubrik penilaian dosen. Masukan krusial dari dosen meminta agar tombol simpan final dilengkapi konfirmasi *pop-up* karena nilai bersifat permanen (*read-only*) setelah dikirim, yang ditindaklanjuti dengan implementasi modal konfirmasi finalisasi. Prioritisasi rubrik didasarkan pada instrumen penilaian ilmiah [7]. Hasil *Sprint Retrospective* merekomendasikan untuk menyederhanakan data binding kriteria rubrik di level database agar meminimalkan kompleksitas pengujian fitur. Pelaksanaan Sprint 3 disajikan pada Tabel 3.

Tabel 3. Ringkasan Siklus Pengembangan Sprint 3

| Atribut              | Deskripsi / Nilai                                                                                                                                                                          |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Sprint Goal          | Merancang modul penilaian digital berbasis rubrik dengan pemisahan relasional nilai dosen penguji.                                                                                         |
| Product Backlog Item | Item M-05 (Modul evaluasi & penilaian independen berbasis rubrik digital).                                                                                                                 |
| Aktor                | Dosen (Penguji & Pembimbing)                                                                                                                                                               |
| Implementasi         | Halaman daftar bimbingan dosen, built-in PDF viewer, formulir rubrik dinamis (5 kriteria penilaian), tabel isolasi database <i>assessment_scores</i> , pop-up konfirmasi finalisasi nilai. |

|              |                                                                                                                                                                                        |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Output       | Dasbor evaluasi dosen, media review PDF proposal tanpa download, form penilaian terisolasi (nilai dosen lain disembunyikan sampai akhir).                                              |
| Implementasi | Setup periode pendaftaran prodi, select dropdown plotting penguji visual, automated average calculator, modul ekspor Excel/PDF (menggunakan library eksternal Laravel Excel & DomPDF). |
| Output       | Modul kelola periode pendaftaran & kriteria rubrik, rekapitulasi penilaian proposal terpusat, analitik dasbor Kaprodi, file Excel/PDF rekapitulasi.                                    |



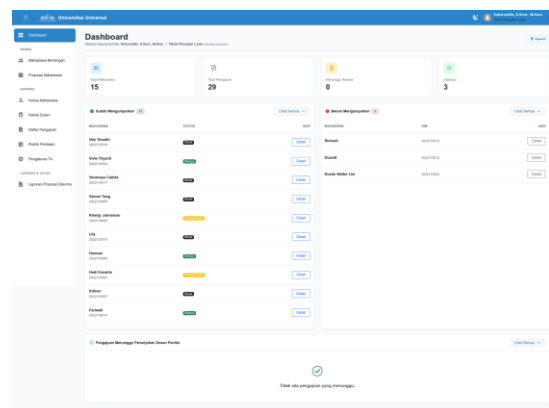
Gambar 6. Tampilan antarmuka penilaian dosen penguji berbasis rubrik

#### 4) Sprint 4: Administrasi dan Monitoring

Sprint keempat difokuskan pada modul administrasi akademik prodi oleh Kaprodi and Admin. *Daily Scrum* harian melacak kemajuan pembuatan kalkulator nilai rata-rata otomatis dan ekspor rekapitulasi Excel. Hambatan utama berupa inkonsistensi formatting file cetak Excel diselesaikan dengan menyusun class template ekspor khusus. Pada *Sprint Review*, dilakukan demo plotting dosen secara visual, pengaturan periode, dan pencetakan rekap. Umpan balik Kaprodi meminta kolom tanggal keputusan ditambahkan secara eksplisit di file rekap, yang kemudian langsung diimplementasikan. *Sprint Retrospective* akhir mengevaluasi seluruh siklus Scrum mandiri yang disepakati berhasil meningkatkan efisiensi pengerjaan backlog, menjaga kualitas kode stabil, serta melampaui batas minimum kriteria *Definition of Done* (DoD). Pelaksanaan Sprint 4 disajikan pada Tabel 4.

Tabel 4. Ringkasan Siklus Pengembangan Sprint 4

| Atribut              | Deskripsi / Nilai                                                                                                                                                                                                |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Sprint Goal          | Membangun modul administrasi Kaprodi, plotting dosen, kelola kriteria rubrik, dan cetak laporan.                                                                                                                 |
| Product Backlog Item | Item M-04 (Plotting dosen penguji), Item M-06 (Plotting pembimbing otomatis), Item M-07 (Penolakan & revisi), Item S-02 (Impor data mahasiswa), Item S-03 (Manajemen periode), Item S-04 (Ekspor/cetak laporan). |
| Aktor                | Kaprodi, Admin                                                                                                                                                                                                   |



Gambar 7. Tampilan antarmuka kelola proposal, rubrik, dan plotting dosen oleh Kaprodi

#### B. Implementasi RBAC Sebagai Keamanan Data

Role-Based Access Control (RBAC) diimplementasikan secara terstruktur untuk menjamin aspek keamanan kerahasiaan data (*confidentiality*) dan pembatasan otorisasi akses (*authorization*) *multi-tenant* pada berkas akademik [10]. Peran pengguna dibagi secara ketat menjadi empat: (1) Mahasiswa: Hanya diizinkan membaca drafnya sendiri, membuat pengajuan baru, dan mengunduh berkasnya sendiri; (2) Dosen: Hanya diizinkan membaca dokumen proposal mahasiswa yang dibimbing atau diuji, serta menginput nilai evaluasinya sendiri; (3) Kaprodi: Memiliki otorisasi penuh untuk melihat seluruh proposal prodi, menetapkan penguji, dan menyetujui status kelulusan; (4) Admin: Mengelola metadata pengguna tanpa akses ke penilaian akademik. Keamanan data diperkuat dengan pencegahan XSS dan SQL Injection melalui penanganan input ORM di Laravel, serta pencegahan manipulasi URL parameter (*Insecure Direct Object References* / IDOR) dengan menguji kepemilikan mahasiswa di level controller.

#### C. Implementasi Deteksi Kemiripan Judul

Fitur deteksi kemiripan judul berkontribusi sebagai filter algoritmik awal guna menghindari duplikasi topik penelitian terdahulu secara tidak sengaja. Sistem menyelesaikan masalah ini dengan mengaktifkan pencarian AJAX asinkron ketika mahasiswa mengisi formulir judul proposal. Judul yang diinput terlebih dahulu dikonversi menjadi huruf kecil,



kemudian fungsi helper `SimilarityHelper::findSimilarSubmissions()` membandingkannya terhadap judul proposal berstatus aktif di database menggunakan fungsi PHP `similar\_text()` [15].

Untuk memperjelas cara kerja algoritma, berikut adalah contoh perhitungan matematis *real-time* yang dilakukan oleh sistem terhadap dua judul proposal: Judul A (Input Mahasiswa) adalah "Sistem Informasi Penjualan Berbasis Web" (39 karakter). Judul B (Data di Database) adalah "Sistem Informasi Penjualan Barang Berbasis Web" (46 karakter). Setelah dikonversi menjadi lowercase, sistem mendeteksi blok karakter identik: "sistem informasi penjualan" (26 karakter) dan " berbasis web" (13 karakter), sehingga total karakter cocok (|C|) adalah 39. Tingkat kemiripan dihitung berdasarkan persamaan (2):

$$\text{Similarity} = \frac{2 \times 39}{39 + 46} \times 100\% = 91,76\% \quad (2)$$

Karena persentase kemiripan (91.76%) melebihi ambang batas 50%, sistem akan menampilkan Judul B sebagai "Judul Serupa" dan memunculkan dialog *warning* asinkron pada formulir. Untuk membantu visualisasi dosen dan Kaprodi, tingkat kemiripan dikategorikan menjadi 3 kelas warna: (1) Merah (persentase  $\geq 70\%$ ): Kemiripan tinggi, risiko duplikasi tinggi; (2) Kuning (persentase 40% - 69%): Kemiripan sedang, umumnya karena istilah standar sejenis; dan (3) Hijau (persentase  $< 40\%$ ): Kemiripan rendah, judul dianggap orisinal.

#### D. Pengujian Sistem

Pengujian kualitas perangkat lunak dijalankan secara menyeluruh untuk memverifikasi kebenaran fungsionalitas visual sistem maupun struktur logika kode program pada level unit dan fitur [23]. Evaluasi ini ditujukan untuk menjamin aspek keandalan (*reliability*) dan keterawatan (*maintainability*) dari platform akademik terintegrasi yang telah dibangun.

##### 1) Black-Box Testing

Pengujian Black-box berfokus pada validasi fungsionalitas sistem pada antarmuka pengguna tanpa mengamati struktur logika internal kode menggunakan metode Black-box [24]. Pengujian dijalankan pada browser dengan menguji 32 skenario pengujian utama (terdiri atas skenario positif seperti login dengan data valid, unggah berkas PDF, serta skenario negatif seperti penginputan judul kosong, unggah file non-PDF, manipulasi URL). Skenario negatif ini penting untuk memastikan ketahanan sistem terhadap input yang tidak terduga. Berdasarkan hasil evaluasi, seluruh skenario pengujian fungsionalitas memberikan respon yang sesuai dengan hasil yang diharapkan (100% pass rate). Sampel skenario dan hasil pengujian fungsionalitas Black-box dirangkum dalam Tabel 5.

Tabel 5. Sampel Hasil Pengujian Black-box

| No | Fitur / Fungsi | Jenis | Hasil Diharapkan (Output) | Status |
|----|----------------|-------|---------------------------|--------|
|----|----------------|-------|---------------------------|--------|

|   |                    |         |                                                          |       |
|---|--------------------|---------|----------------------------------------------------------|-------|
| 1 | Login Multi-role   | Positif | Validasi email/password & arahkan ke dashboard           | Valid |
| 2 | Login Multi-role   | Negatif | Menolak sandi salah & muncul warning input               | Valid |
| 3 | Otorisasi RBAC     | Negatif | Blokir paksa akses rute Mahasiswa ke Kaprodi (403)       | Valid |
| 4 | Unggah Proposal    | Negatif | Menolak file non-PDF & ukuran file > 10MB                | Valid |
| 5 | Similarity Check   | Positif | Tampil modal warning visual jika judul mirip >50%        | Valid |
| 6 | Isolasi Nilai      | Positif | Penyimpanan terisolasi, sembunyikan nilai pengujian lain | Valid |
| 7 | Plotting Pengujian | Positif | Menyimpan data dosen pengujian terpilih ke database      | Valid |

##### 2) Unit dan Feature Testing (PHPUnit)

Pengujian unit dan fitur otomatis (*white-box*) diimplementasikan menggunakan kerangka PHPUnit otomatis bawaan Laravel guna memastikan integritas kode program dan ketahanan logika bisnis terhadap *breaking changes* [25]. Pengujian unit memvalidasi fungsi internal terisolasi (seperti filter validasi constraints jatah pengajuan), sedangkan pengujian fitur mensimulasikan request HTTP tiruan untuk menguji autentikasi, RBAC, validasi form, dan security checks secara otomatis [26]. Pengujian regresi otomatis ini mencegah munculnya kutu kode (*bugs*) baru saat dilakukan pemeliharaan sistem di masa mendatang. Total 183 test case dengan 611 assertions dijalankan tanpa kegagalan (0 failed/error). Pembagian hasil *automated testing* dirangkum pada Tabel 6.

Tabel 6. Ringkasan Hasil Automated Testing

| Kategori Pengujian                    | Jumlah Test Case | Jumlah Assertions | Status |
|---------------------------------------|------------------|-------------------|--------|
| Unit Testing (Logika Service & Model) | 48               | 107               | Passed |
| Feature Testing (Autentikasi & RBAC)  | 15               | 45                | Passed |
| Feature Testing (Pengajuan Mahasiswa) | 38               | 144               | Passed |
| Feature Testing (Penilaian Dosen)     | 40               | 152               | Passed |
| Feature Testing (Manajemen Kaprodi)   | 32               | 112               | Passed |

|                                         |            |            |                         |
|-----------------------------------------|------------|------------|-------------------------|
| Feature Testing<br>(Dashboard & Profil) | 10         | 51         | Passed                  |
| <b>Total</b>                            | <b>183</b> | <b>611</b> | <b>Lolos<br/>(100%)</b> |

### 3) Code Coverage (Whitebox Testing)

Metrik cakupan kode (*Code Coverage*) diukur menggunakan ekstensi Xdebug/PCOV untuk mengevaluasi persentase pernyataan (*statements*) kode program yang tereksekusi oleh *test suite* [26]. Analisis *code coverage* dipisahkan ke dalam tiga layer utama: (1) Service Layer (logika bisnis), (2) Controller Layer (HTTP routing & penanganan input), dan (3) Model Layer (entitas database). Hasil *code coverage* disajikan pada Tabel 7, Tabel 8, dan Tabel 9.

Tabel 7. Hasil Code Coverage pada Layer Service (Logika Bisnis)

| Nama Kelas / File Service               | Statements Coverage | Status Kriteria    |
|-----------------------------------------|---------------------|--------------------|
| Services/Student/SubmissionService      | 100.0%              | Excellent          |
| Services/Examiner/AssessmentService     | 100.0%              | Excellent          |
| Services/Coordinator/CoordinatorService | 100.0%              | Excellent          |
| Services/Supervisor/SupervisorService   | 100.0%              | Excellent          |
| Services/DashboardService               | 100.0%              | Excellent          |
| Services/AuthService                    | 92.3%               | Excellent          |
| Services/Kaprodi/KaprodiService         | 63.0%               | Good               |
| <b>Rata-rata Layer Service</b>          | <b>93.6%</b>        | <b>Sangat Baik</b> |

Tabel 8. Hasil Code Coverage pada Layer Controller (Antarmuka HTTP)

| Nama Kelas / File Controller             | Statements Coverage | Status Kriteria |
|------------------------------------------|---------------------|-----------------|
| Controllers/Student/SubmissionController | 100.0%              | Excellent       |
| Controllers/Dosen/AssessmentController   | 100.0%              | Excellent       |
| Controllers/Dosen/SubmissionController   | 100.0%              | Excellent       |
| Controllers/AuthController               | 100.0%              | Excellent       |
| Controllers/DashboardController          | 100.0%              | Excellent       |
| Controllers/SimilarityController         | 100.0%              | Excellent       |

|                                           |              |                    |
|-------------------------------------------|--------------|--------------------|
| Controllers/ProfileController             | 87.0%        | Excellent          |
| Controllers/Kaprodi/StudentController     | 95.7%        | Excellent          |
| Controllers/Kaprodi/LecturerController    | 77.7%        | Good               |
| Controllers/Admin/StudentController       | 81.2%        | Good               |
| Controllers/Admin/LecturerController      | 81.4%        | Good               |
| Controllers/Admin/UserController          | 77.3%        | Good               |
| Controllers/Admin/SubmissionController    | 94.7%        | Excellent          |
| Controllers/Admin/ConfigurationController | 100.0%       | Excellent          |
| Controllers/FileDownloadController        | 79.1%        | Good               |
| Controllers/Kaprodi/KaprodiController     | 42.1%        | Fair               |
| <b>Rata-rata Layer Controller</b>         | <b>87.0%</b> | <b>Sangat Baik</b> |

Tabel 9. Hasil Code Coverage pada Layer Model (Entitas Basis Data)

| Nama Kelas / File Model      | Statements Coverage | Status Kriteria    |
|------------------------------|---------------------|--------------------|
| Models/AssessmentRubric      | 100.0%              | Excellent          |
| Models/Rubric                | 100.0%              | Excellent          |
| Models/ThesisStatus          | 100.0%              | Excellent          |
| Models/SubmissionFile        | 100.0%              | Excellent          |
| Models/Faculty               | 100.0%              | Excellent          |
| Models/ProgramStudi          | 100.0%              | Excellent          |
| Models/Setting               | 100.0%              | Excellent          |
| Models/Comment               | 100.0%              | Excellent          |
| Models/AssessmentCriterion   | 83.3%               | Excellent          |
| Models/Assessment            | 74.4%               | Good               |
| Models/ThesisSubmission      | 72.6%               | Good               |
| Models/User                  | 55.8%               | Fair               |
| <b>Rata-rata Layer Model</b> | <b>89.7%</b>        | <b>Sangat Baik</b> |

Hasil cakupan kode (*code coverage*) membuktikan keandalan sistem yang sangat baik. Rata-rata cakupan layer

*Service* (93,6%) didorong oleh cakupan penuh (100%) pada kelas-kelas operasional utama. Cakupan 63,0% pada `KaprodiService` disebabkan oleh fungsi analitik sekunder yang tidak terpicu pada unit test. Pada layer *Controller*, rata-rata cakupan sebesar 87,0% membuktikan otorisasi HTTP terlindungi secara optimal. Nilai terendah pada `KaprodiController` (42,1%) disebabkan oleh modul integrasi pustaka ekspor Excel eksternal yang tidak dievaluasi oleh *test suite* unit. Cakupan model database rata-rata mencapai 89,7%, memastikan integritas logika mutator database berjalan stabil.

#### IV. KESIMPULAN

Berdasarkan analisis hasil penelitian dan evaluasi sistem, dapat ditarik beberapa kesimpulan sebagai berikut: (1) Metode Agile Scrum sangat membantu proses pengembangan sistem melalui pembagian siklus Sprint terstruktur, memungkinkan penyelesaian fungsional secara bertahap dengan feedback stakeholder di akhir siklus; (2) Sistem pengajuan proposal TA berbasis web berhasil memodernisasi proses administrasi semi-manual, menghemat waktu rekapitulasi data, serta memfasilitasi pelacakan status akademik secara *real-time*; (3) Otorisasi berbasis RBAC terbukti andal dalam menjaga kerahasiaan (*confidentiality*) dokumen proposal serta membatasi hak akses file PDF secara *multi-tenant*; (4) Fitur deteksi kemiripan judul (*similarity check*) asinkron berhasil membantu memvalidasi orisinalitas judul proposal secara dini dengan mendeteksi kemiripan kecocokan karakter di atas 50%; (5) Hasil pengujian Black-box fungsional (100% pass), Unit/Feature testing PHPUnit (183 tests, 611 assertions, 100% pass), dan White-box (cakupan kode di atas 87% pada layer kritis) membuktikan keandalan kualitas sistem.

Saran untuk pengembangan sistem selanjutnya adalah mengintegrasikan algoritma *Natural Language Processing* (NLP) yang lebih mendalam seperti *Cosine Similarity* dengan TF-IDF atau model *Vector Embeddings* guna mendeteksi tingkat plagiarisme isi konten draf dokumen proposal secara utuh [27], serta penerapan kecerdasan buatan (AI) untuk memberikan rekomendasi otomatis dosen pembimbing dan penguji yang didasarkan pada kecocokan kepakaran bidang penelitian dosen terhadap topik proposal mahasiswa.

#### DAFTAR PUSTAKA

- [1] D. Ardiansyah, S. El, and R. Yogyakarta, "Transformasi Digital Perguruan Tinggi Menggunakan Prinsip Smart Education," vol. 20, no. 1, 2022.
- [2] Y. Soegoto, H. Prabowo, A. Nizar Hidayanto, A. Trisetyarso, and Y. Dedy Pradipto, "Digital Transformation in Higher Education," *International Journal of Research and Applied Technology*, vol. 3, no. 2, 2023.
- [3] R. S. Tanjung, E. S. Tanjung, Khoyan, and A. Nasution, "Transformasi Digital dalam Administrasi dan Manajemen Pendidikan di Perguruan Tinggi," *CIVITAS*, vol. 16, no. 1, 2025.
- [4] M. Z. Arsyad, T. Mary, and S. Junaidi, "Perancangan Sistem Informasi Akademik (SIKAD) Berbasis Web Di SMK Negeri 1 Sijunjung," *Jurnal Ilmiah Sistem Informasi Dan Teknik Informatika (JISTI)*, vol. 8, no. 1, pp. 65–75, 2025.
- [5] P. Hermawan, P. A. Sunarya, and M. Rakhmansyah, "Designing a Web-Based Online Thesis Guidance Application for the Computer Science Study Program, Bina Bangsa University," *JINAV: Journal of Information and Visualization*, vol. 5, no. 1, pp. 157–170, 2024.
- [6] R. Dewi, H. Satyareni, and E. Kurniawan, "METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi," vol. 9, no. 1, pp. 76–85, 2025.
- [7] R. Hariadi and A. S. G. P. R. W. S. T. F. K. M., "Pengembangan Instrumen Penilaian Produk Proposal Penelitian Mahasiswa," *Malewa: Journal of Multidisciplinary Educational Research*, vol. 2, no. 01, pp. 8–20, 2024.
- [8] J. A. M. Fareen, "Digital Learning in Higher Education: A Road to Transformation and Reform," *European Journal of Interactive Multimedia and Education*, vol. 3, no. 1, p. e02206, 2022.
- [9] M. Farhan Londjo, "Implementasi White Box Testing dengan Teknik Basis Path pada Pengujian Form Login," vol. 7, no. 2, 2021.
- [10] P. Kaur, F. Farahlina, A. Alam, S. Kaur, and R. S. Sahota, "Access Control Application Prevention and Mitigation of Cyber Attacks," *International Journal of Research and Innovation in Applied Science*, vol. VIII, no. X, pp. 91–105, 2023.
- [11] E. Fitriani, D. Ardiansyah, A. Saepudin, and R. Aryanti, "Penerapan Sistem Informasi Akademik Berbasis Web Menggunakan Metode Rapid Application Development," *Journal of Information System, Applied, Management, Accounting and Research*, vol. 8, no. 4, pp. 770–782, 2024.
- [12] Hardyanti and A. Fauzi, "Analisis Dampak Penggunaan Sistem Informasi Akademik (SIKAD) Terhadap Efektivitas Administrasi Kampus di STKIP Harapan Bima," 2024.
- [13] B. Yadav, V. Shrivastava, A. Pandey, and P. Sharma, "MySQL: Database Design for Performance and Scalability," *International Journal of Research Publication and Reviews*, no. 5, 2024.
- [14] O. C. Resmi Rachmawati, D. K. Wardani, W. M. Fatihia, A. Fariza, and H. Rante, "Implementing Agile Scrum Methodology in The Development of SICITRA Mobile Application," *Jurnal RESTI*, vol. 7, no. 1, pp. 41–50, 2023.
- [15] N. Sabulina Cahyani, D. Stiawan, A. Abdiansah, N. Afifah, and D. Renaldo Permana, "Classification and Similarity Detection of Indonesian Scientific Journal Articles," *Computer Science and Information Technologies*, vol. 6, no. 2, pp. 147–158, 2025.
- [16] R. Putrianasari, E. K. Budiardjo, K. Mahatma, and T. Raharjo, "Problems in The Adoption of Agile-Scrum Software Development Process in Small Organization: A Systematic Literature Review," *Sinkron*, vol. 9, no. 1, pp. 495–504, 2024.
- [17] A. R. P. Ramadhan, I. Waspada, N. Bahtiar, and A. S. Pramayoga, "Applying the Scrum Method in Software Development for Undergraduate Thesis Project Implementation," *Jurnal Masyarakat Informatika*, vol. 16, no. 1, pp. 119–133, 2025.
- [18] J. Romeo, M. Raglianti, C. Nagy, and M. Lanza, "UML is Back. Or is It? Investigating the Past, Present, and Future of UML in Open Source Software," in *Proceedings of the International Conference on Software Engineering*, 2025, pp. 2342–2354.
- [19] W. Mu and K. H. Lim, "Modelling Text Similarity: A Survey," in *Proceedings of the 2023 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2023, pp. 698–705.
- [20] A. Lathifah, U. Isni Kurnia, and U. Aisyah Pringsewu, "Efektivitas Penggunaan Figma Sebagai Alat Prototyping Dalam Mata Kuliah Interaksi Manusia Dan Komputer," *Jurnal Pendidikan Teknologi Informasi*, vol. 3, no. 2, pp. 40–45, 2024.
- [21] I. F. Pasaribu, "Xiosena: Figma-Based Prototype Design and Development for Boutique Dress Ordering Using UI/UX," *Jurnal Ilmiah Global Education*, vol. 6, no. 4, pp. 2420–2434, 2025.
- [22] K. Ishag, A. Fadlallah, M. M. Sharif, M. Elfaki, and Y. Eldow, "Developing Parallel Requirements Prioritization Machine Learning Model Integrating with MoSCoW Method," 2024.
- [23] A. C. Praniffa, A. Syahri, F. Sandes, U. Fariha, and Q. A. Giansyah, "Pengujian Black Box dan White Box Sistem Informasi Parkir



- Berbasis Web," *Jurnal Testing dan Implementasi Sistem Informasi*, vol. 1, no. 1, 2023.
- [24] S. Ismiati, F. P. Aditiawan, and A. L. Nurlaili, "Black Box Testing with the Equivalence Partitioning and Cause Effect Graph Method in Archive Information System," *Jurnal Teknik Informatika (Jutif)*, vol. 5, no. 4, pp. 981–990, 2024.
- [25] M. R. Haksono, I. Nuryasin, B. F. Muthorihin, M. Titani, A. Akrom, and A. Fadlil, "Pengujian Aplikasi Menggunakan White Box dengan Metode Unit Testing. Studi Kasus: MAHATI," *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, vol. 10, no. 4, pp. 4124–4138, 2025.
- [26] E. A. Putri, "Implementasi Automation Testing Menggunakan Tools Appium pada Aplikasi Mobile Android: Studi Kasus pada Aplikasi Balink," 2024.
- [27] E. Suhardi Rahman, M. Iswal Burhan, and R. T. Mangesa, "Development of Sentence Similarity Detection Application with Semantic Similarity and Machine Learning Approaches (Case Study: Student Thesis Title)," *Jurnal Media Elektrik*, vol. 23, no. 1, pp. 19–36, 2025.