

PENGEMBANGAN ANTARMUKA SISTEM KINERJA DOSEN UNIVERSITAS LAMPUNG MENGGUNAKAN REACT JS

M. Iqbal Parabi¹, Igit Sabda Ilman², Amalia Nurul Rahmawati³, Chandra Prasetya Putra⁴

¹²³⁴Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Lampung, Indonesia

Corresponding Author: iqbal.parabi@fmipa.unila.ac.id

INFORMASI

Artikel History:

Rec. 20 February 2025
Acc. May 2025
Pub. Juni 2025
Page. 116-130

Kata kunci:

- Antarmuka sistem
- React JS
- Scrum
- Sistem kinerja dosen

ABSTRAK

This research focuses on developing the user interface of the Lecturer Performance System at Universitas Lampung using React JS framework. The development process applied the Scrum methodology, which was divided into five sprints. Each sprint emphasized specific deliverables, including interface design, component development, service integration, and iterative evaluation, enabling a structured yet flexible workflow. System validation was carried out using two approaches. First, black-box testing ensured that each function aligned with user requirements. Second, Lighthouse was utilized to evaluate performance, accessibility, and overall technical quality of the web application. The results demonstrate that the system performed reliably across predefined test scenarios, achieved strong performance ratings, and provided a satisfactory user experience. The findings indicate that combining React JS with the Scrum methodology can effectively support the creation of an interactive and responsive interface tailored to institutional needs. This system has the potential to enhance transparency and accountability in monitoring lecturer performance within the Universitas Lampung.

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



PENDAHULUAN

Kemajuan teknologi informasi memengaruhi berbagai aspek kehidupan, termasuk pendidikan tinggi. Komputer telah berkembang dari sekadar alat hitung menjadi alat pengambilan keputusan yang canggih. Dengan perkembangan ini, institusi pendidikan tinggi, termasuk Universitas Lampung (Unila), didorong untuk terus menciptakan sistem informasi yang efektif, efisien, dan terintegrasi. Kinerja dosen sangat penting dalam konteks ini karena berkontribusi langsung pada pelaksanaan Tri Dharma Perguruan Tinggi yaitu pendidikan, penelitian, pengabdian, dan penunjang. Untuk alasan ini, Unila menggunakan sistem remunerasi dosen yang berbasis pada Beban Kerja Dosen (BKD) untuk menilai kinerja dosen (Lampung, 2025).

Menurut penelitian terdahulu (Sugiarti, 2022), kinerja dosen sangat dipengaruhi oleh Tri Dharma Perguruan Tinggi, yang terdiri dari pendidikan, penelitian, dan pengabdian kepada masyarakat. Dalam Unila, penilaian kompensasi didasarkan pada prestasi dosen pada ketiga bidang tersebut, serta

aktivitas pendukung. Remunerasi sendiri adalah kompensasi yang diberikan kepada dosen sebagai penghargaan atas kontribusinya. Ini berfungsi sebagai penghargaan moneter dan sebagai alat strategis untuk menyediakan data kinerja yang terdokumentasi. Data ini sangat penting untuk proses akreditasi program studi karena BAN-PT menekankan bahwa bukti capaian kinerja dosen harus diukur sesuai dengan standar kualitas pendidikan tinggi (BAN-PT, 2024).

Namun, masalah utama dengan sistem yang ada adalah keterbatasan data kinerja dosen untuk tim akreditasi, dekan, dan ketua program studi. Sebagian besar proses pengumpulan data dilakukan secara manual, sehingga prosesnya sering terfragmentasi, yang menyebabkan keterlambatan dan inkonsistensi data. Akibatnya, validasi memerlukan waktu yang lama dan pengambilan keputusan strategis sering tertunda. Oleh karena itu, untuk keperluan strategis dan akreditasi, dibutuhkan sistem baru yang dapat memberikan antarmuka pengguna akses data secara real-time, terintegrasi lintas unit, dan mudah digunakan.

Studi terdahulu menunjukkan bahwa pengembangan sistem berbasis React JS untuk pengembangan antarmuka sistem (Fariz et al., 2022). Pada penelitian terdahulu mengembangkan sistem informasi digital printing berbasis website yang menggunakan React JS, yang dinilai responsif dan memiliki tingkat kepuasan pengguna yang lebih tinggi, dengan nilai rata-rata 3,53 (Nuraminudin et al., 2023). Penelitian oleh Levlin (Levlin, 2020), sebaliknya, menyelidiki kinerja framework JavaScript populer seperti React, Angular, dan Vue dalam hal manipulasi DOM, dan menyimpulkan bahwa React memiliki efisiensi rendering dan kecepatan yang lebih tinggi daripada framework lainnya. Menurut penelitian Zareinia (Zareinia, 2024), React unggul dalam skalabilitas dan penggunaan memori karena arsitektur berbasis komponen dan minimalis.

Berbeda dengan penelitian sebelumnya yang berfokus pada kinerja teknis framework atau sistem layanan umum. Penelitian ini menawarkan inovasi berupa pengembangan antarmuka sistem kinerja dosen yang terintegrasi dengan persyaratan akreditasi pada bidang pendidikan, penelitian, pengabdian dan penunjang. Penggunaan metodologi pengembangan Scrum, yang memungkinkan iterasi dan evaluasi sistem yang berkelanjutan, serta integrasi antarmuka yang dibuat dengan API sistem backend yang sudah ada, adalah keuntungan dari penelitian ini (Ardiansyah et al., 2021). Oleh karena itu, penelitian ini menempati posisi strategis dalam mengembangkan sistem informasi akademik yang dapat berfungsi dan membantu proses manajemen kualitas institusi.

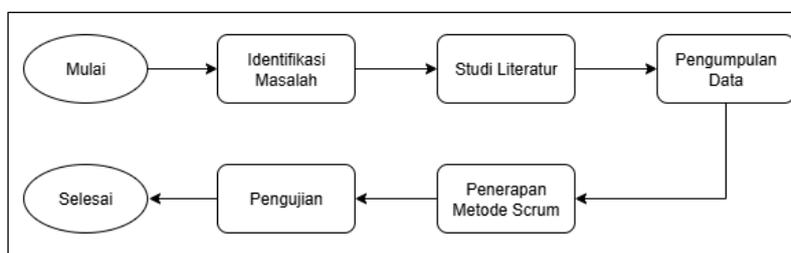
Tujuan dari penelitian ini adalah untuk membuat dan menerapkan antarmuka front-end untuk sistem kinerja dosen Universitas Lampung yang responsif, mudah digunakan, dan dapat menampilkan data secara real-time untuk keperluan akreditasi. Penyediaan antarmuka sistem yang dapat digunakan oleh berbagai peran pengguna, seperti dekan, ketua program studi, dan dosen, adalah kontribusi utama dari penelitian ini. Pada akhirnya, ini akan membantu mempercepat proses penilaian dan pengambilan keputusan institusional.

Penelitian ini hanya membahas pengembangan antarmuka front-end pendidikan, penelitian, pengabdian dan penunjang. Ini juga tidak mencakup

proses evaluasi kompensasi secara keseluruhan atau integrasi dengan sistem informasi akademik lainnya di universitas.

METODE

Pada penelitian ini, metode *scrum* memungkinkan pengembangan sistem yang berulang, fleksibel, dan responsif terhadap perubahan yang terjadi (Sekar Kinanti et al., 2023). Dalam implementasi, proses pengembangan dibagi menjadi lima sprint, yaitu satuan waktu kerja singkat yang berkonsentrasi pada penyelesaian komponen tertentu secara bertahap (Ayurira & Nur Fajri, 2024). Setiap *sprint* mencakup tahapan perencanaan untuk menentukan prioritas fitur dan tugas (Dzaky & Kurniawan, 2023) (Nico Abrarsyah Atallah & Mardi Mardi, 2024). Tahapan penelitian akan menyelidiki literatur tentang teknologi dan metode pengembangan, mengidentifikasi masalah dan kebutuhan pengguna, mengumpulkan data primer dan sekunder, membangun sistem dengan UML dan desain antarmuka, menggunakan React JS yang terintegrasi RESTful API untuk mengimplementasikan antarmuka (Rahmadhani et al., 2024) (Siska Narulita et al., 2024). Selain itu, untuk menguji fungsionalitas dan performa sistem menggunakan metode *black-box testing* dan *Lighthouse*.



Gambar 1. Tahap penelitian

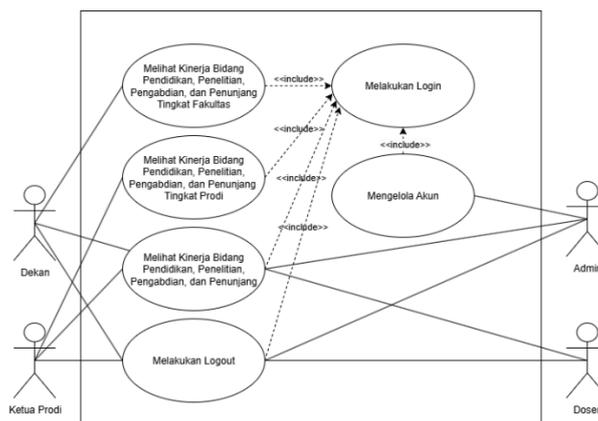
Pada tahap awal penelitian, masalah dan kebutuhan sistem di Universitas Lampung diidentifikasi. Salah satu masalah utama adalah bahwa data kinerja dosen hanya dapat dilihat oleh dosen terkait. Ini menghambat proses evaluasi, akreditasi, dan pengambilan keputusan. Selain itu, proses administrasi menjadi lebih lambat karena kurangnya integrasi dengan sistem lain. Oleh karena itu, sistem dibuat agar dekan dan ketua program studi dapat dengan cepat, aman, dan responsif mengakses informasi kinerja dosen. Diharapkan bahwa desain sistem yang terintegrasi dengan antarmuka modern akan memiliki kemampuan untuk meningkatkan kualitas pendidikan dan mempercepat penilaian akademik.

Studi literatur dilakukan untuk mendapatkan pemahaman yang lebih baik tentang rangka kerja React JS dan menemukan masalah yang terkait dengan pembuatan antarmuka front-end berbasis web. Fokus penelitian adalah mengevaluasi keunggulan React JS, yang termasuk antarmuka interaktif dan responsif, pendekatan berbasis komponen, dan kemudahan dalam mengelola DOM. Selain itu, masalah akses multi-peran pengguna dan integrasi data

menjadi subjek penelitian (Murti et al., 2021) (Raina et al., 2024). Menentukan alat, arsitektur, dan teknik desain antarmuka didasarkan pada pemahaman ini.

Pengumpulan data dalam penelitian ini yaitu dengan memperoleh informasi yang relevan dalam menganalisis kebutuhan serta merancang sistem antarmuka kinerja dosen. Data yang dikumpulkan dibagi menjadi dua bagian, yaitu data primer dan sekunder diperoleh melalui wawancara dengan ketua program studi untuk mengetahui kebutuhan, masalah, dan harapan sistem. Dokumentasi sistem remunerasi dosen yang sudah ada memberikan data sekunder, yang membantu memahami alur kerja dan kemungkinan integrasi data untuk mendukung akreditasi yang lebih efisien.

Dalam penelitian ini, metode *Scrum* digunakan sebagai pendekatan pengembangan perangkat lunak yang fleksibel dan iteratif untuk memastikan bahwa proses pembangunan sistem berjalan secara terstruktur dan dapat menyesuaikan diri dengan perubahan kebutuhan pengguna (Cucolaş & Russo, 2023). Tim yang menggunakan *Scrum* dapat membagi proses pengembangan ke dalam beberapa iterasi yang disebut *sprint*, yang masing-masing berkonsentrasi pada mencapai tujuan fitur tertentu (Verwijs & Russo, 2023). Perencanaan tugas (*sprint planning*), pelaksanaan pengembangan (*development*), evaluasi hasil sementara (*sprint review*), dan refleksi tim tentang perbaikan berkelanjutan dilakukan dalam setiap *sprint* (*sprint retrospective*) (Asri Yana Vita et al., 2023). Metode ini memungkinkan evaluasi dan penyempurnaan sistem secara bertahap berdasarkan umpan balik yang diperoleh. Sehingga akan menghasilkan antarmuka yang lebih sesuai dengan kebutuhan pengguna dan membantu mencapai tujuan institusional seperti pengambilan keputusan dan akreditasi. Metode *Scrum* membuat pengembangan sistem lebih fleksibel dan bekerja sama, dengan fokus pada kepuasan pengguna akhir (Istiqomah et al., 2024).



Gambar 2. Use case diagram sistem kinerja

Dalam penelitian ini, sistem diuji untuk memastikan bahwa fitur antarmuka memenuhi kebutuhan dan memberikan pengalaman pengguna terbaik. Pengujian *black-box* dan pengujian *lighthouse* adalah dua jenis pengujian yang digunakan. Pengujian *black-box* berfokus pada menguji

fungsionalitas sistem tanpa melihat kode internalnya, dengan menguji berbagai input pengguna untuk memastikan apakah sistem memberikan output yang tepat. Pada saat yang sama, pengujian *lighthouse* digunakan untuk mengevaluasi kinerja antarmuka dari perspektif kecepatan, aksesibilitas, dan praktik pengembangan terbaik (*best practices*). Hasil dari kombinasi kedua metode pengujian ini menunjukkan bahwa sistem yang dikembangkan tidak hanya fungsional tetapi juga efisien dan ramah pengguna.

HASIL DAN PEMBAHASAN

Product Backlog

Pada tahap ini, tugasnya adalah membuat daftar lengkap semua kebutuhan dan fitur sistem yang diinginkan berdasarkan kebutuhan sistem. Setiap *item backlog* dibahas secara menyeluruh dan diberi prioritas berdasarkan manfaat dan pentingnya. Jika terjadi perubahan selama proses pengembangan, tim juga harus memperbarui *backlog* secara berkala.

Tabel 1. *Product backlog*

Kode	Product Backlog	Kepentingan	Perkiraan Waktu
PB-01	Modeling	Tinggi	8 Hari
PB-02	Pembuatan <i>Mockup</i>	Sedang	6 Hari
PB-03	<i>Login</i> dan <i>logout</i>	Tinggi	9 Hari
PB-04	Mengelola akun	Tinggi	7 Hari
PB-05	Tambah akun	Sedang	4 Hari
PB-06	Edit akun	Sedang	3 Hari
PB-07	Hapus akun	Sedang	3 Hari
PB-08	Profil	Tinggi	7 Hari
PB-09	Daftar program studi	Sedang	4 Hari
PB-10	Daftar dosen	Sedang	7 Hari
PB-11	Dasbor	Sedang	5 Hari
PB-12	Kinerja Fakultas	Sedang	3 Hari
PB-13	Kinerja Program Studi	Sedang	3 Hari
PB-14	Panduan & Bantuan	Rendah	1 Hari

Sprint Planning

Pada tahap ini, peneliti melakukan perencanaan *sprint* dengan menyusun daftar kegiatan berdasarkan produk *backlog* yang telah diprioritaskan. Tujuan dari perencanaan *sprint* adalah untuk menentukan tugas apa saja yang dapat diselesaikan dalam waktu yang singkat, biasanya satu hingga dua minggu. Tim pengembang berbicara tentang tujuan *sprint* dengan pemilik produk. Kemudian, membagi setiap *item backlog* menjadi tugas teknis yang lebih kecil dan dapat diukur. Setelah tahap ini selesai, dokumen kerja atau *sprint backlog* dibuat. Dokumen ini berisi daftar tugas yang harus diselesaikan selama *sprint*, serta estimasi waktu dan tanggung jawabnya. Seluruh tim harus jelas memahami ruang lingkup dan tujuan *sprint*, jadi tahap ini sangat penting untuk pelaksanaan *sprint*.

Tabel 2. Sprint ke-1

Kode	Task Product Backlog	Task	Waktu
PB-01	<i>Modeling</i>	Membuat <i>use case diagram</i> dan membuat <i>activity diagram</i>	8 Hari
PB-02	Pembuatan <i>Mockup</i>	Membuat <i>wireframe</i> dan membuat <i>hi-fi</i>	6 Hari

Tabel 3. Sprint ke-2

Kode	Task Product Backlog	Task	Waktu
PB-03	<i>Login dan logout</i>	Membuat halaman <i>login dan logout</i> dan mengelola token autentikasi yang disediakan oleh <i>backend</i>	9 Hari
PB-11	Dasbor	Membuat halaman dasbor untuk semua peran serta menarik data yang telah disediakan oleh <i>backend</i> melalui API	5 Hari

Tabel 4. Sprint ke-3

Kode	Task Product Backlog	Task	Waktu
PB-04	Mengelola akun	Membuat halaman mengelola akun pengguna dan menarik data yang disediakan oleh <i>backend</i> melalui API.	7 Hari
PB-05	Tambah akun	Membuat halaman tambah akun pengguna dan sinkronisasi data ke <i>backend</i> .	4 Hari
PB-06	Edit akun	Membuat halaman edit akun pengguna dan sinkronisasi data ke <i>backend</i> .	3 Hari

Tabel 5. Sprint ke-4

Kode	Task Product Backlog	Task	Waktu
PB-07	Hapus akun	Membuat fungsi hapus akun pengguna dan sinkronisasi data ke <i>backend</i> .	3 Hari
PB-08	Daftar program studi	Membuat halaman daftar program studi dan menarik data dari <i>backend</i> melalui API	4 Hari
PB-10	Daftar dosen	Membuat halaman yang menampilkan daftar dosen dan menarik data yang telah disediakan oleh <i>backend</i>	7 Hari

Tabel 6. *Sprint ke-5*

Kode	Task Product Backlog	Task	Waktu
PB-08	Profil	Membuat halaman profil dan menarik data dari <i>backend</i> melalui API.	7 Hari
PB-12	Kinerja Fakultas	Membuat halaman kinerja fakultas dan menarik data dari <i>backend</i> melalui API.	3 Hari
PB-13	Kinerja Program Studi	Membuat halaman kinerja program studi dan menarik data dari <i>backend</i> melalui API	3 Hari
PB-14	Panduan & Bantuan	Membuat halaman panduan sistem	1 Hari

Daily Scrum

Daily scrum adalah pertemuan singkat yang dilakukan setiap hari oleh tim pengembang untuk memastikan bahwa semua tim bekerja sama dan *sprint* dilakukan dengan jelas. Selama sesi ini, setiap anggota tim menyampaikan apa yang telah dilakukan sejak pertemuan sebelumnya, apa yang akan dilakukan hari ini, dan apakah ada masalah atau hambatan yang menghalangi tim untuk maju. Tujuan utama *daily scrum* adalah untuk menjaga fokus tim, menemukan masalah secepat mungkin, dan memastikan bahwa semua anggota tim memiliki pemahaman yang sama tentang kondisi proyek. Meskipun pertemuan ini singkat, biasanya tidak lebih dari 15 menit, mereka sangat penting untuk berkomunikasi dan bekerja sama dengan baik dalam metode *Scrum*.

Sprint Review

Pada akhir setiap *sprint*, *Sprint Review* adalah tahapan dalam metode *Scrum* yang dilakukan untuk mengevaluasi hasil kerja tim pengembang bersama stakeholder. Pada tahap ini, tim menunjukkan fitur atau komponen sistem yang telah dikembangkan dan menunjukkan fungsinya secara langsung. Stakeholder memberikan masukan, memberikan umpan balik, dan memastikan bahwa hasil yang dicapai memenuhi persyaratan dan tujuan *sprint*. Selain itu, *Sprint Review* memberikan kesempatan untuk meninjau kembali produk *backlog*, mengubah prioritas, dan merencanakan tindakan berikutnya sesuai dengan kondisi proyek terbaru. Tahapan ini sangat penting untuk memastikan bahwa pengembangan sesuai dengan ekspektasi pengguna dan sistem tetap fleksibel untuk menyesuaikan diri dengan perubahan kebutuhan.

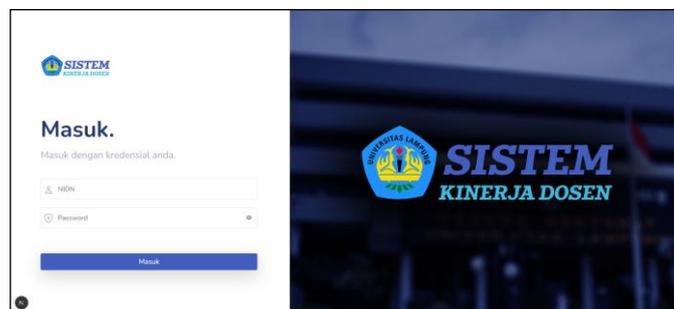
Sprint Retrospective

Sprint Retrospective adalah tahapan evaluasi yang dilakukan oleh tim pengembang setelah *sprint* selesai, dengan tujuan untuk merefleksikan proses kerja selama *sprint* berlangsung. Tujuan dari sesi ini adalah untuk merenungkan proses kerja selama *sprint*, menemukan masalah yang berjalan dengan baik, dan mencari solusi atau perbaikan untuk meningkatkan kinerja

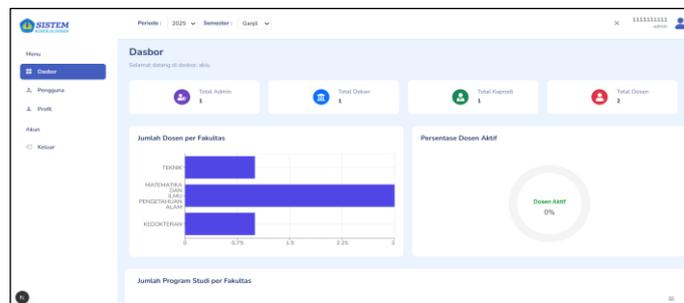
di *sprint* berikutnya. Melalui perbaikan berkelanjutan, *sprint retrospective* membantu tim untuk terus belajar dan berkembang.

Hasil implementasi

Pada tampilan awal sistem disajikan pada Gambar 3. Setelah memiliki akun pengguna dapat masuk ke dalam *website* dengan menggunakan *nidn* dan *password* yang disediakan oleh admin. Setelah melakukan login maka pengguna dapat mengakses halaman dasbor seperti yang diperlihatkan pada Gambar 4. Kemudian peran admin dapat melihat daftar pengguna seperti pada Gambar 5. Di dalam daftar pengguna tersebut admin dapat mengedit, menghapus dan menambahkan data pengguna.



Gambar 3. Login

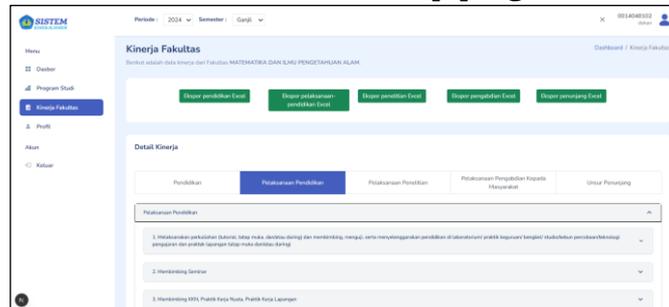


Gambar 4. Dasbor

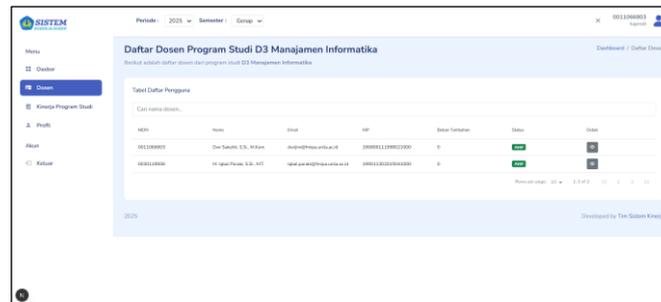
Nama	NIDN	Jenis	Fakultas	Prodi	Lain	Pass
Dar Ganiy, S.S., M.Pd.	801000003	kepariw	MATEMATIKA DAN ILMU PE.	D3 Manajemen Informatika	✓	✗
Yudhanu, S. Am., M.Pd., Ph.D.	801000010	akadem	MATEMATIKA DAN ILMU PE.	D3 Informatika	✓	✗
M. Iqbal Parizi, S.S., M.T.	800010006	akadem	MATEMATIKA DAN ILMU PE.	D3 Manajemen Informatika	✓	✗
Andi	222222222	akadem	KEDOKTERAN	Perawat	✓	✗

Gambar 5. Daftar pengguna

Pada Gambar 6 pengguna dapat melihat kinerja dosen per fakultas serta dapat melakukan ekspor data kinerja di setiap bidangnya. Pada Gambar 7 pengguna dapat melihat daftar dosen di setiap program studi.



Gambar 6. Kinerja fakultas



Gambar 7. Daftar dosen di setiap program studi

Pengujian

Hasil pengujian antarmuka sistem kinerja dosen menunjukkan bahwa semua fitur utama bekerja dengan baik dan memenuhi kebutuhan pengguna. Metode pengujian *black box* digunakan untuk melakukan pengujian *fungsiional*, yang memverifikasi setiap *input* dan *output* tanpa menguji kode internal. Hasil pengujian menunjukkan bahwa semua alur berjalan sesuai harapan. Selain itu, pengujian performa menggunakan *lighthouse* menghasilkan skor tinggi pada kategori performa, aksesibilitas, dan praktik pengembangan, dengan skor rata-rata di atas 90 pada perangkat *desktop*. Meskipun kinerja perangkat *mobile* sedikit lebih buruk, sistem secara keseluruhan dinilai stabil, responsif, dan layak digunakan untuk membantu proses evaluasi dan akreditasi kinerja dosen di Universitas Lampung.

Tabel 7. Hasil Pengujian UAT Peran Kaprodi

Pertanyaan	Penilaian					Presentase
	1	2	3	4	5	
Antarmuka sistem mudah dipahami dan mendukung kemudahan pengguna.	0	0	0	2	1	86.67%
Proses autentikasi (<i>login</i> dan <i>logout</i>) berjalan dengan lancar tanpa kendala.	0	0	0	2	1	86.67%

Pertanyaan	Penilaian					Presentase
	1	2	3	4	5	
Halaman dashboard menampilkan informasi yang relevan dan sesuai dengan peran pengguna.	0	0	1	0	2	86.67%
Halaman data kinerja program studi menampilkan data dengan benar sesuai dengan bidang yang dipilih.	0	1	0	1	1	73.33%
Fitur filter data berdasarkan periode bekerja sesuai yang diharapkan.	0	0	0	2	1	86.67%
Fitur ekspor data kinerja dapat digunakan dan hasilnya sesuai dengan data yang ditampilkan.	0	0	0	3	0	80%
Halaman profil menampilkan informasi kinerja pribadi secara akurat dan lengkap.	0	0	0	3	0	80%
Navigasi antar halaman sistem berlangsung dengan cepat dan mudah digunakan.	0	0	0	1	2	93.33%
Sistem menampilkan pesan kesalahan atau validasi dengan jelas.	0	0	0	2	1	86.67%
Sistem secara otomatis menolak akses ke fitur atau halaman yang tidak sesuai dengan hak akses saya.	0	0	0	2	1	86.67%
Kecepatan akses dan pemuatan halaman termasuk baik dan tidak membuat pengguna menunggu terlalu lama.	0	0	0	3	0	80%
Layout antarmuka tetap konsisten dan responsif saat diakses dari berbagai perangkat.	0	0	0	2	1	86.67%
Sistem berfungsi sesuai dengan tujuan pengembangan sistem kinerja dosen.	0	1	0	1	1	73.33%
Sistem mempermudah proses pengumpulan dan pemantauan data kinerja dosen.	0	0	1	1	1	80%
Secara umum, saya merasa puas menggunakan sistem ini untuk menunjang kebutuhan saya.	0	0	0	1	0	26.67%

Tabel 8. Hasil Pengujian UAT Peran Dosen

Pertanyaan	Penilaian					Presentase
	1	2	3	4	5	
Antarmuka sistem mudah dipahami dan mendukung kemudahan pengguna.	0	0	1	2	2	84%
Proses autentikasi (<i>login</i> dan <i>logout</i>) berjalan dengan lancar tanpa kendala.	0	0	0	1	4	96%
Halaman dashboard menampilkan informasi yang relevan dan sesuai dengan peran pengguna.	0	1	0	3	1	76%
Fitur filter data berdasarkan periode bekerja sesuai yang diharapkan.	0	0	0	4	1	84%
Halaman profil menampilkan informasi kinerja pribadi secara akurat dan lengkap.	0	0	0	3	2	88%
Navigasi antar halaman sistem berlangsung dengan cepat dan mudah digunakan.	0	0	0	2	3	92%

Pertanyaan	Penilaian					Presentase
	1	2	3	4	5	
Sistem menampilkan pesan kesalahan atau validasi dengan jelas.	0	0	0	2	3	92%
Sistem secara otomatis menolak akses ke fitur atau halaman yang tidak sesuai dengan hak akses saya.	0	0	0	2	3	92%
Kecepatan akses dan pemuatan halaman termasuk baik dan tidak membuat pengguna menunggu terlalu lama.	0	0	0	2	3	92%
Layout antarmuka tetap konsisten dan responsif saat diakses dari berbagai perangkat.	0	0	0	4	1	84%
Sistem berfungsi sesuai dengan tujuan pengembangan sistem kinerja dosen.	0	0	1	2	2	84%

Tabel 9. Hasil Pengujian UAT Peran Dekan

Pertanyaan	Penilaian					Presentase
	1	2	3	4	5	
Antarmuka sistem mudah dipahami dan mendukung kemudahan pengguna.	0	0	0	1	0	80%
Proses autentikasi (<i>login</i> dan <i>logout</i>) berjalan dengan lancar tanpa kendala.	0	0	0	0	1	100%
Halaman dashboard menampilkan informasi yang relevan dan sesuai dengan peran pengguna.	0	0	0	0	1	100%
Halaman data kinerja program studi menampilkan data dengan benar sesuai dengan bidang yang dipilih.	0	0	0	0	1	100%
Fitur filter data berdasarkan periode bekerja sesuai yang diharapkan.	0	0	0	0	1	100%
Fitur ekspor data kinerja dapat digunakan dan hasilnya sesuai dengan data yang ditampilkan.	0	0	0	1	0	80%
Halaman profil menampilkan informasi kinerja pribadi secara akurat dan lengkap.	0	0	0	0	1	100%
Navigasi antar halaman sistem berlangsung dengan cepat dan mudah digunakan.	0	0	0	0	1	100%
Sistem menampilkan pesan kesalahan atau validasi dengan jelas.	0	0	0	0	1	100%
Sistem secara otomatis menolak akses ke fitur atau halaman yang tidak sesuai dengan hak akses saya.	0	0	0	0	1	100%
Kecepatan akses dan pemuatan halaman termasuk baik dan tidak membuat pengguna menunggu terlalu lama.	0	0	0	0	1	100%
Layout antarmuka tetap konsisten dan responsif saat diakses dari berbagai perangkat.	0	0	0	1	0	80%
Sistem berfungsi sesuai dengan tujuan pengembangan sistem kinerja dosen.	0	0	0	0	1	100%
Sistem mempermudah proses pengumpulan dan pemantauan data kinerja dosen.	0	0	0	0	1	100%

Tabel 10. Hasil Pengujian *Lighthouse Desktop*

Halaman	FCP (<i>Fist Contentful Paint</i>)	LCP (<i>Largest Contentful Paint</i>)	TBT (<i>Total Blacking Time</i>)	CLS (<i>Cumulative Layout Shift</i>)	SI (<i>Speed Index</i>)	Skor
Login	0.2s	0.2s	0ms	0	0.4s	100
Daftar pengguna	0.2s	0.4s	0ms	0.052	0.5s	100
Tambah pengguna	0.2s	0.4s	0ms	0	0.5s	100
Edit pengguna	0.2s	0.4s	0ms	0	0.5s	100
Daftar prodi	0.2s	0.4s	0ms	0.089	0.5s	98
Daftar dosen	0.2s	0.4s	0ms	0	0.6s	100
Profil	0.2s	0.5s	10ms	0.001	0.7s	100
Dasbor admin	0.2s	0.4s	80ms	0	0.8s	99
Dasbor dekan	0.2s	0.4s	0ms	0	0.6s	100
Dasbor kaprodi	0.2s	0.4s	0ms	0.146	0.7s	94
Dasbor dosen	0.2s	0.4s	0ms	0	0.6s	100
Kinerja fakultas	0.5s	1.2s	30ms	0.004	0.6s	98
Kinerja program studi	0.5s	1.2s	30ms	0.004	0.6s	98
Panduan & bantuan	0.5s	0.8s	0ms	0.001	0.6s	99

Tabel 11. Hasil Pengujian *Lighthouse Mobile*

Halaman	FCP (<i>Fist Contentful Paint</i>)	LCP (<i>Largest Contentful Paint</i>)	TBT (<i>Total Blacking Time</i>)	CLS (<i>Cumulative Layout Shift</i>)	SI (<i>Speed Index</i>)	Skor
Login	0.7s	0.8s	170ms	0	0.8s	98
Daftar pengguna	0.7s	1.5s	300ms	0.037	1.4s	94
Tambah pengguna	0.7s	1.5s	230ms	0	1.4s	96
Edit pengguna	0.7s	1.7s	230ms	0	1.4s	96
Daftar prodi	0.7s	1.4s	280ms	0.054	1.4s	94
Daftar dosen	0.7s	1.5s	430ms	0	1.5s	89

Halaman	FCP (<i>Fist Contentful Paint</i>)	LCP (<i>Largest Contentful Paint</i>)	TBT (<i>Total Blacking Time</i>)	CLS (<i>Cumulative Layout Shift</i>)	SI (<i>Speed Index</i>)	Skor
Profil	0.7s	1.7s	390ms	0.001	2.0s	90
Dasbor admin	0.8s	1.6s	1180ms	0	2.3s	76
Dasbor dekan	0.7s	1.6s	810ms	0	1.9s	81
Dasbor kaprodi	0.7s	1.6s	680ms	0	1.7s	83
Dasbor dosen	0.7s	1.5s	430ms	0.003	1.6s	90
Kinerja fakultas	2.3s	6.0s	340ms	0	2.3s	68
Kinerja program studi	2.3s	6.0s	340ms	0.06	2.3s	67
Panduan & bantuan	2.3s	4.2s	140ms	0	2.3s	82

KESIMPULAN

Hasil penelitian dan implementasi menunjukkan bahwa antarmuka sistem kinerja dosen di bidang pendidikan, penelitian, pengabdian dan penunjang telah dirancang dan dikembangkan dengan sukses menggunakan React JS. Antarmuka ini memungkinkan pengguna mengelola dan menampilkan data kinerja dosen dengan benar dan mendukung integrasi data melalui API yang disediakan oleh tim backend. Sistem ini memungkinkan program studi untuk mengakses informasi kinerja sesuai dengan kebutuhan mereka. Serta terbukti memenuhi fungsinya dalam pengujian black-box dan cukup baik di perangkat desktop dan mobile menurut pengujian lighthouse.

DAFTAR PUSTAKA

- Ardiansyah, A., Yahya, F. Y., Irawati, A. R., & Yusman, M. (2021). Pengembangan Sistem Informasi Terpadu Fmipa Universitas Lampung (Simipa) Menggunakan Metode Scrum. *Jurnal Teknoinfo*, 15(2), 112. <https://doi.org/10.33365/jti.v15i2.1041>
- Asri Yana Vita, D., Raihan, R., & Zulfachmi, Z. (2023). Implementasi Metode Scrum pada Transformasi Bisnis Lokal UMKM Tanjungpinang. *Jurnal Bangkit Indonesia*, 12(2), 46–52. <https://doi.org/10.52771/bangkitindonesia.v12i2.249>
- Ayurira, C. L. A., & Nur Fajri, I. (2024). Implementasi Metode Scrum Dalam Pengembangan Website E-Commerce Pada Twins Petshop. *Journal of Innovation And Future Technology (IFTECH)*, 6(2), 259–270. <https://doi.org/10.47080/iftech.v6i2.3397>
- BAN-PT. (2024). *Badan Akreditasi Nasional Perguruan Tinggi*.

<https://www.banpt.or.id/>

- Cucolaş, A. A., & Russo, D. (2023). The impact of working from home on the success of Scrum projects: A multi-method study. *Journal of Systems and Software*, 197, 111562. <https://doi.org/10.1016/j.jss.2022.111562>
- Dzaky, F. A., & Kurniawan, D. (2023). Implementation of Scrum Framework Agile Method to Develop Integrated Asset Management Information System at Universitas Diponegoro Inventory Module. *Jurnal Masyarakat Informatika*, 14(1), 53–69. <https://doi.org/10.14710/jmasif.14.1.52605>
- Fariz, M., Lazuardy, S., & Anggraini, D. (2022). Modern Front End Web Architectures with React.Js and Next.Js. *International Research Journal of Advanced Engineering and Science*, 7(1), 132–141.
- Istiqomah, D. A., Vikky Aprelia Windarni, Surya Tri Atmaja Ramadhani, & Zein Nur Ichsan. (2024). Peningkatan Pengetahuan Agile Scrum dalam Pengembangan Perangkat Lunak pada Startup Kala Kreatif Indonesia. *JURPIKAT (Jurnal Pengabdian Kepada Masyarakat)*, 5(3), 915–924. <https://doi.org/10.37339/jurpikat.v5i3.1777>
- Lampung, U. (2025). *Unila Sosialisasikan Rubrik Remunerasi Baru untuk Dosen dan Verifikatur*. <https://www.unila.ac.id/unila-sosialisasikan-rubrik-remunerasi-baru-untuk-dosen-dan-verifikatur/>
- Levlin, M. (2020). *DOM Benchmark Comparison Of The Front-End JavaScript Frameworks React, Angular, Vue, and Svelte*. 95.
- Murti, S. K., Informatika, J., Industri, T., Sujarwo Badan, A., & Informasi, S. (2021). Membangun Antarmuka Pengguna Menggunakan ReactJs untuk Modul Manajemen Pengguna. *Journal Portal Universitas Islam Indonesia*, 2(2), 1–6. <https://journal.uui.ac.id/AUTOMATA/article/view/19443>
- Nico Abrarsyah Atallah, & Mardi Mardi. (2024). Penggunaan Metode Agile Scrum Pada Perancangan Sistem Informasi Surat Izin Penelitian di BAKESBANGPOL Lombok Tengah. *Neptunus: Jurnal Ilmu Komputer Dan Teknologi Informasi*, 2(3), 371–384. <https://doi.org/10.61132/neptunus.v2i3.276>
- Nuraminudin, M., Dewi, M. M., Suharsono, S., Dahlan, A., & Lukman, L. (2023). Implementasi Reactjs Pada Pembuatan Sistem Informasi Digital Printing Berbasis Website. *Information System Journal*, 6(01), 25–32. <https://doi.org/10.24076/infosjournal.2023v6i01.1214>
- Rahmadhani, S., Wildana, D. W., Arumdanie, H. W., & Hakim, L. (2024). Penerapan React JS dan Axios untuk Pengembangan Front-end Aplikasi iCare. *Software Development Digital Business Intelligence and Computer Engineering*, 2(02), 40–46. <https://doi.org/10.57203/session.v2i02.2024.40-46>
- Raina, A. N., Utama, N. I., & Suakanto, S. (2024). Pengembangan React.js pada frontend website pengaduan dan pelayanan publik menggunakan metode Scrum (Studi kasus: DPRD Jawa Barat). *E-Proceeding of Engineering*, 11(4), 4292–4300.
- Sekar Kinanti, V., Setyadi, R., Widiyari, L., Informasi, S., Purwokerto Jl Panjaitan No, I. DI, Sel, P., Banyumas, K., & Tengah, J. (2023).

- Implementasi React JS dalam Mengembangkan Front-End Website PT XYZ. *Jurnal Ilmu-Ilmu Informatika Dan Manajemen*, 17(2), 1–10.
- Siska Narulita, Ahmad Nugroho, & M. Zakki Abdillah. (2024). Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS). *Bridge : Jurnal Publikasi Sistem Informasi Dan Telekomunikasi*, 2(3), 244–256. <https://doi.org/10.62951/bridge.v2i3.174>
- Sugiarti, E. (2022). The Impact of Tri Dharma Performance on Higher Education Performance Based on Monitoring Results. *AKADEMIK: Jurnal Mahasiswa Humanis*, 2(3), 120–126. <https://doi.org/10.37481/jmh.v2i3.476>
- Verwijs, C., & Russo, D. (2023). A Theory of Scrum Team Effectiveness. *ACM Transactions on Software Engineering and Methodology*, 32(3). <https://doi.org/10.1145/3571849>
- Zareinia, K. (2024). *Benchmarking Front-End Frameworks: A Deep Dive Into The Performance And Scalability Of React, Angular, And Vue.js*. November.