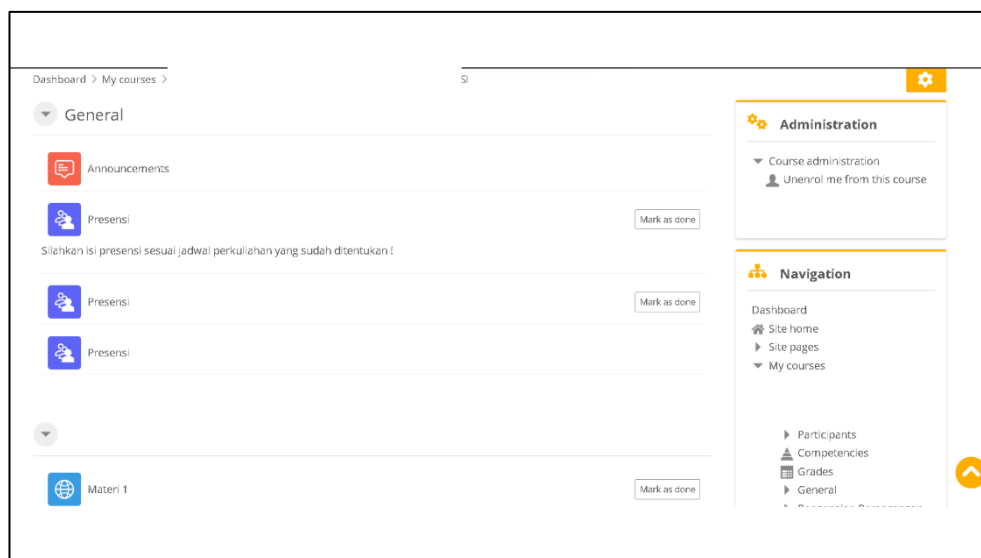


BAB IV HASIL DAN PEMBAHASAN

4.1 Analisis Sistem Yang ada

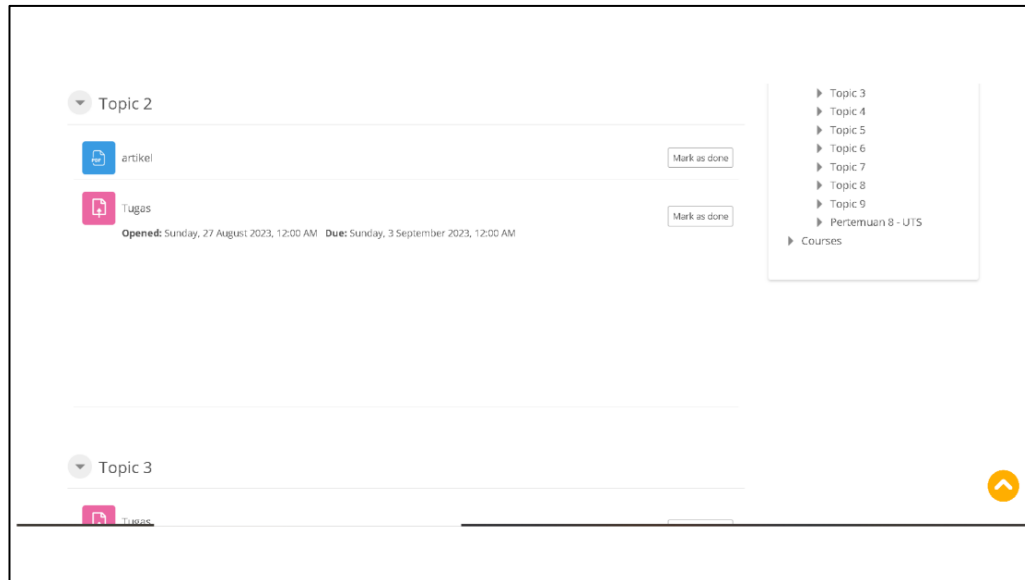
4.1.2 Hasil Observasi

Observasi dilakukan terhadap *platform E-Learning* yang digunakan oleh dosen di Fakultas Teknik Universitas Sriwijaya untuk memahami pemanfaatan Rencana Pembelajaran Semester (RPS) dalam kegiatan pembelajaran. Berdasarkan pengamatan, ditemukan bahwa tidak semua kelas menyertakan RPS dalam halaman *E-Learning* seperti pada gambar 4.1. Pada beberapa kelas, RPS diunggah di bagian awal perkuliahan sebagai dokumen terpisah, biasanya dalam format *PDF*. Namun, keberadaannya bersifat pasif dan tidak diintegrasikan lebih lanjut ke dalam alur pembelajaran.



Gambar 4. 1 Salah Satu Contoh Kelas di *E-Learning*, Tidak Mencantumkan RPS

Selain itu pada gambar 4.2, pelaksanaan sesi perkuliahan sering kali tidak mengikuti urutan atau isi yang tercantum dalam RPS. Tidak terdapat fitur dalam sistem yang mampu merekam atau memantau kesesuaian antara pelaksanaan pembelajaran dan perencanaan yang ada dalam RPS. Presensi, penyampaian materi, serta evaluasi dilakukan secara terpisah tanpa keterkaitan langsung dengan struktur RPS atau capaian pembelajaran (CPMK). Hal ini menyulitkan proses pengawasan dan evaluasi terhadap konsistensi pelaksanaan perkuliahan.



Gambar 4. 2 Salah Satu Contoh Kelas di *E-Learning*, Sesi Tidak Lengkap

Temuan dari observasi ini menunjukkan adanya kebutuhan akan sistem yang lebih terintegrasi, yang tidak hanya menyimpan RPS sebagai referensi, tetapi juga mengaitkannya dengan aktivitas pembelajaran, presensi, dan dokumentasi sesi secara otomatis dan terstruktur. Meskipun temuan ini tidak dapat digeneralisasi untuk seluruh mata kuliah atau kelas yang ada, namun temuan ini dapat dijadikan sebagai representasi awal untuk menggambarkan kondisi sistem yang sedang berjalan, serta menjadi dasar identifikasi terhadap kekurangan dalam implementasi RPS pada sistem pembelajaran daring yang ada.

4.1.2 Hasil Studi Dokumentasi

Studi dokumentasi dilakukan terhadap template Rencana Pembelajaran Semester (RPS) resmi yang digunakan di lingkungan Fakultas Teknik Universitas Sriwijaya. Dokumen ini menjadi acuan utama dalam perancangan fitur manajemen RPS pada sistem yang dikembangkan. Berdasarkan struktur dokumen, RPS terdiri dari dua bagian besar, yaitu Identitas Mata Kuliah dan Program Pembelajaran.

Bagian Identitas Mata Kuliah mencakup elemen-elemen penting seperti nama dan kode mata kuliah, bahan kajian, jumlah SKS, semester, deskripsi mata kuliah, capaian pembelajaran (CPL dan CPMK), serta data dosen pengajar dan otorisasi pihak fakultas seperti yang ada pada gambar 4.3. Elemen-elemen ini menjadi landasan bagi sistem untuk menampilkan informasi RPS secara lengkap dan

terstandarisasi. Sementara itu pada gambar 4.4, bagian Program Pembelajaran berisi rincian mingguan yang mencakup Sub-CPMK, materi pembelajaran, referensi, metode dan alokasi waktu pembelajaran, jenis asesmen, indikator pencapaian, bobot penilaian, serta dosen pengampu tiap sesi.

		UNIVERSITAS SRIWIJAYA FAKULTAS JURUSAN PROGRAM STUDI				
		RENCANA PEMBELAJARAN SEMESTER				
A. IDENTITAS MATA KULIAH						
Nama mata kuliah	Kode mata kuliah	Bahan kajian	sks		Semester	Tanggal revisi terakhir
(1)	(2)	(3)	Kuliah (4)	Praktikum (5)	(6)	(7)
Deskripsi mata kuliah	(8)					
Capaian Pembelajaran Mata Kuliah (CPMK)	(9) CPL yang dibebankan ke mata kuliah					
	(10) CPMK yang dibuat dari CPL yang dibebankan ke mata kuliah					
Tim pengajar	(11)		Ketua tim pengajar : (12) Instruktur (bila ada) : (13)			
Otorisasi	(14)		(15)			
	Ketua program studi		Wakil dekan bidang akademik			

42

Gambar 4. 3 Bagian Identitas RPS Acuan

B. PROGRAM PEMBELAJARAN								
CPMK	Kompetensi mingguan (Sub-CPMK)	Materi pembelajaran	Referensi	Metodologi pembelajaran dan alokasi waktunya	Deskripsi tugas atau asesmen beserta alokasi waktunya	Indikator	Bobot	Dosen
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
CMKP-1	Sub-CPMK-1							
	Sub-CPMK-2							
	Sub-CPMK-3							
CPMK-2	Sub-CPMK-4							
	Sub-CPMK-5							
	Sub-CPMK-6							
	Sub-CPMK-7							
UJIAN TENGAH SEMESTER (10)								Tim
CPMK-3	Sub-CPMK-8							
	Sub-CPMK-9							
	Sub-CPMK-10							
CPMK-4	Sub-CPMK-11							
CPMK-5	Sub-CPMK-12							
	Sub-CPMK-13							
	Sub-CPMK-14							
UJIAN AKHIR SEMESTER (11)								Tim
Beban belajar mahasiswa selama satu semester (12)								

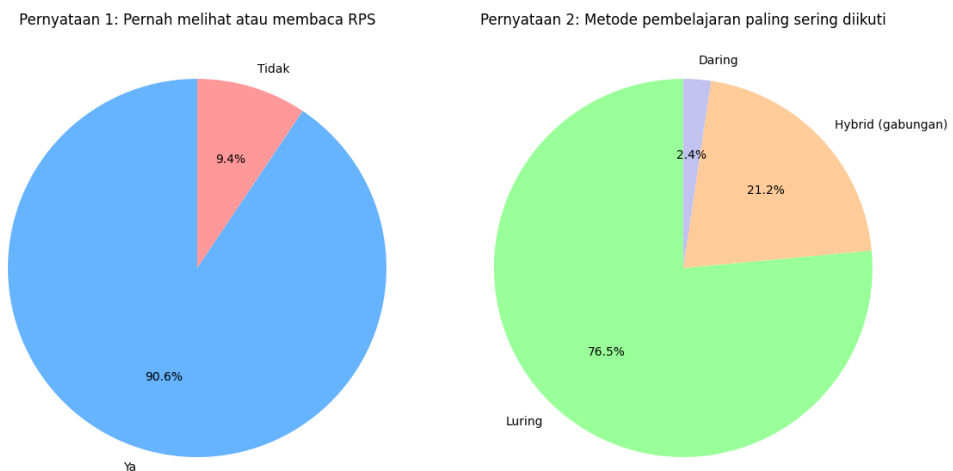
Gambar 4. 4 Bagian Program Pembelajaran RPS Acuan

Selain itu pada gambar 4.4, dokumen ini mencakup baris khusus untuk Ujian Tengah Semester (UTS) dan Ujian Akhir Semester (UAS), serta perhitungan total beban belajar mahasiswa selama satu semester. Seperti yang ditunjukkan oleh struktur ini, RPS tidak hanya berfungsi sebagai dokumen deskripsi tetapi juga sebagai alat untuk mengontrol proses pembelajaran. Oleh karena itu, sistem yang dikembangkan dimaksudkan untuk menangani *input* data yang berasal dari struktur

ini, mengelola setiap komponen secara modular, dan memungkinkan integrasi antara pelaksanaan pembelajaran digital dan perencanaan (RPS).

4.2 Hasil Kuesioner Analisis Kebutuhan

Kuesioner disebarikan kepada mahasiswa aktif Program Studi Teknik Elektro sebagai representatif dari mahasiswa Fakultas Teknik Universitas Sriwijaya. Total sebanyak 85 responden berhasil mengisi kuesioner dengan lengkap. Kuesioner ini bertujuan untuk menggali pemahaman, persepsi, serta dukungan mahasiswa terhadap implementasi Rencana Pembelajaran Semester (RPS), efektivitas sistem informasi akademik yang digunakan saat ini, serta kebutuhan terhadap digitalisasi proses pembelajaran. Pertanyaan disusun menggunakan skala Likert 1–5, di mana 1 berarti sangat tidak setuju dan 5 berarti sangat setuju. Data dianalisis menggunakan statistik deskriptif berupa rata-rata (*mean*) dan standar deviasi, guna mengetahui kecenderungan umum serta tingkat keseragaman pendapat responden terhadap masing-masing pernyataan.



Gambar 4. 5 Hasil Analisis Kuesioner untuk Pertanyaan Demografi

Untuk pernyataan pertama mengenai pengalaman mahasiswa dalam melihat atau membaca RPS, gambar 4.5 menunjukkan bahwa mayoritas mahasiswa (77 dari 85 responden atau 90.6%) telah pernah melihat atau membaca RPS, sementara hanya 8 mahasiswa (9.4%) yang menyatakan belum pernah melihatnya. Temuan ini sangat positif karena mengindikasikan bahwa sebagian besar mahasiswa sudah memiliki *exposure* terhadap dokumen RPS, yang berarti mereka memiliki pemahaman dasar tentang struktur dan konten pembelajaran.

Sementara itu, distribusi jawaban untuk pernyataan kedua tentang metode pembelajaran yang paling sering diikuti memberikan gambaran penting tentang preferensi dan realitas pembelajaran di lingkungan akademik responden. Hasil menunjukkan dominasi yang sangat kuat dari pembelajaran luring atau tatap muka langsung, dengan 65 mahasiswa (76.5%) menyatakan metode ini sebagai yang paling sering mereka ikuti. Pembelajaran *hybrid* atau gabungan menempati posisi kedua dengan 18 mahasiswa (21.2%), sedangkan pembelajaran daring murni hanya dipilih oleh 2 mahasiswa (2.4%).

Pola distribusi pada gambar 4.5 mencerminkan kondisi pembelajaran pasca-pandemi dimana institusi pendidikan kembali ke pembelajaran tatap muka sebagai modalitas utama, namun tetap mempertahankan elemen digital dalam bentuk pembelajaran *hybrid*. Rendahnya preferensi terhadap pembelajaran daring murni (2.4%) kemungkinan disebabkan oleh pengalaman selama pandemi atau preferensi mahasiswa terhadap interaksi langsung dengan dosen dan sesama mahasiswa. Namun, cukup signifikannya porsi pembelajaran *hybrid* (21.2%) menunjukkan bahwa mahasiswa tetap terbuka terhadap integrasi teknologi digital dalam proses pembelajaran.

Kombinasi kedua temuan ini memberikan landasan yang kuat untuk pengembangan sistem informasi akademik. Tingginya familiaritas mahasiswa terhadap RPS (90.6%) menunjukkan bahwa implementasi sistem digital yang mengintegrasikan RPS tidak akan mengalami hambatan pemahaman yang berarti. Sementara itu, dominasi pembelajaran luring dengan adanya elemen *hybrid* menunjukkan bahwa sistem yang akan dikembangkan harus dapat mendukung kedua modalitas pembelajaran tersebut, dengan fokus utama pada pembelajaran tatap muka namun tetap menyediakan fitur-fitur digital yang dapat diakses kapan saja untuk mendukung proses pembelajaran yang lebih fleksibel dan terintegrasi.

Tabel 4. 1 Hasil Kuesioner

No	Pernyataan	Rata-Rata	Standar Deviasi
1	Saya merasa belum pernah benar-benar melihat atau memahami isi dari RPS pada setiap mata kuliah.	2.76	1.11
2	Informasi materi perkuliahan seharusnya disediakan secara terpusat agar mudah diakses kapan saja.	4.28	1.02
3	Akan sangat membantu jika setiap perkuliahan dapat dikaitkan langsung dengan RPS dan CPMK yang telah dirancang.	4.08	0.95
4	Saya setuju jika mahasiswa diberikan kesempatan untuk memberikan umpan balik terhadap pelaksanaan perkuliahan setiap sesi.	4.20	0.92
5	Sistem informasi akademik yang saya gunakan saat ini belum menyediakan fitur pelaporan kesesuaian pembelajaran dengan RPS.	3.53	1.11
6	<i>Feedback</i> dari mahasiswa setelah perkuliahan dapat membantu evaluasi dan peningkatan kualitas pembelajaran.	4.22	0.96
7	Sistem informasi berbasis <i>web</i> akan memudahkan dosen dan mahasiswa dalam proses dokumentasi perkuliahan.	4.16	1.01

8	Saya mendukung digitalisasi proses akademik yang mencakup absensi, RPS, dan pelaporan perkuliahan.	4.31	0.90
9	Saya merasa sistem presensi saat ini masih belum efektif karena masih dilakukan secara manual atau tidak terpusat.	3.60	1.18
10	Sistem informasi yang terintegrasi dapat membantu dosen dan mahasiswa mengelola pembelajaran lebih efektif.	4.31	0.91

Tabel 4.1 merupakan lanjutan dari hasil analisis statistik pernyataan Likert menunjukkan implikasi strategis yang sangat penting untuk pengembangan sistem informasi akademik. Dukungan yang sangat tinggi terhadap digitalisasi dan sistem terintegrasi (keduanya dengan *mean* 4.31) mengindikasikan bahwa mahasiswa tidak hanya siap menerima perubahan teknologi, tetapi secara aktif menginginkannya. Konsistensi dukungan ini, yang tercermin dari standar deviasi yang relatif rendah (0.90-0.91), menunjukkan bahwa kebutuhan akan digitalisasi merupakan konsensus yang kuat di kalangan mahasiswa, bukan hanya keinginan sebagian kecil responden.

Tingginya dukungan terhadap penyediaan informasi secara terpusat (4.28) dan sistem *feedback* (4.20-4.22) mengungkap dua kebutuhan fundamental dalam proses pembelajaran *modern*. Mahasiswa menginginkan akses yang mudah dan konsisten terhadap materi pembelajaran, sekaligus kesempatan untuk berpartisipasi aktif dalam evaluasi dan perbaikan proses pembelajaran. Standar deviasi yang rendah pada aspek-aspek ini (0.92-1.02) menunjukkan bahwa kebutuhan tersebut dirasakan secara universal, tidak terbatas pada kelompok mahasiswa tertentu dengan preferensi teknologi tinggi.

Pernyataan dengan dukungan sedang memberikan gambaran tentang pain points dalam sistem saat ini. Skor 3.60 untuk ketidakefektifan sistem presensi yang ada, meskipun dengan standar deviasi tertinggi (1.18), menunjukkan bahwa meskipun sebagian besar mahasiswa mengalami masalah dengan sistem presensi saat ini, pengalaman mereka cukup beragam. Hal ini kemungkinan disebabkan oleh perbedaan implementasi sistem presensi oleh beberapa dosen, dimana beberapa mungkin sudah menggunakan sistem yang lebih baik sementara yang lain masih mengandalkan cara manual.

Pengakuan terhadap keterbatasan sistem informasi akademik saat ini (3.53) dengan standar deviasi 1.11 juga mencerminkan kondisi yang heterogen. Beberapa mahasiswa mungkin sudah merasakan manfaat dari fitur-fitur yang ada dalam SIA, sementara yang lain merasa sistem tersebut belum memenuhi kebutuhan mereka akan transparansi dan kesesuaian pembelajaran dengan RPS. Variasi ini mengindikasikan perlunya standarisasi dan peningkatan fitur yang komprehensif.

Yang paling menarik adalah temuan pada pernyataan dengan skor terendah (2.76) tentang ketidakpahaman RPS. Skor ini sebenarnya positif karena menunjukkan bahwa mahasiswa tidak setuju dengan pernyataan bahwa mereka belum memahami RPS. Kombinasi dengan data distribusi yang menunjukkan 90.6% mahasiswa pernah melihat RPS, serta standar deviasi 1.11 yang mengindikasikan masih ada variasi pemahaman, memberikan gambaran bahwa meskipun mayoritas mahasiswa sudah familiar dengan RPS, masih ada ruang untuk peningkatan pemahaman dan akses yang lebih baik.

Secara holistik, pola hasil tabel 4.1 mengungkap paradoks yang menarik, mahasiswa memiliki dukungan sangat tinggi terhadap digitalisasi dan sistem terintegrasi, namun sistem yang ada saat ini belum memenuhi ekspektasi mereka. Mereka sudah familiar dengan konsep-konsep akademik seperti RPS, tetapi menginginkan akses dan implementasi yang lebih efektif melalui teknologi. Kondisi ini menciptakan momentum yang sangat baik untuk implementasi sistem baru, karena ada kombinasi antara awareness yang tinggi terhadap kebutuhan, familiaritas dengan konsep akademik, dan dukungan kuat terhadap solusi teknologi. Temuan ini juga menunjukkan bahwa mahasiswa bukan hanya pengguna pasif, tetapi *stakeholders* yang siap berpartisipasi aktif dalam proses evaluasi dan

perbaikan sistem pembelajaran melalui mekanisme *feedback* yang mereka dukung secara konsisten.

4.3 Hasil Pengembangan Sistem

Pengembangan sistem dilakukan secara bertahap menggunakan metode *Agile* dengan *framework Scrum*, yang membagi proses pengembangan menjadi empat *sprint*. Meskipun pada awalnya setiap *sprint* dirancang berlangsung selama 4 minggu seperti yang dirincikan pada tabel 3.6, dalam pelaksanaannya terjadi penyesuaian durasi sesuai kebutuhan dan dinamika proses pengembangan. *Sprint* pertama berlangsung dari 6 April hingga 1 Juni, *sprint* kedua dari 3 Juni hingga 27 Juni, *sprint* ketiga dari 29 Juni hingga 16 Juli, dan *sprint* keempat dari 16 Juli hingga 24 Juli. Setiap akhir *sprint* dilakukan evaluasi dan perbaikan melalui sesi konsultasi bersama dosen pembimbing yang berperan sebagai representatif pengguna. Pendekatan iteratif ini memungkinkan sistem dikembangkan secara fleksibel dan responsif terhadap masukan selama proses berlangsung.

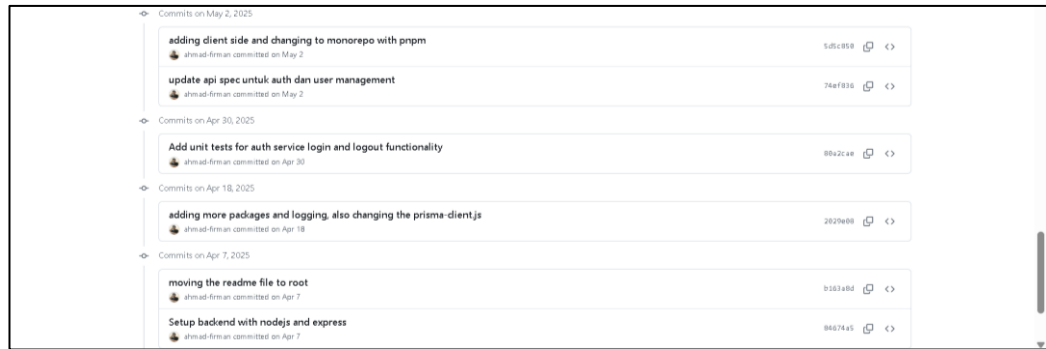
4.3.1 Implementasi per *Sprint*

4.3.1.1 *Sprint* Pertama: Autentikasi dan *User Management*

Sprint pertama dilaksanakan pada tanggal 6 April hingga 1 Juni 2025. *Sprint* ini memiliki durasi lebih panjang dari rencana awal karena difokuskan untuk membangun fondasi awal sistem, khususnya pada aspek keamanan dan kontrol akses. Tujuan utama dari *sprint* ini adalah mengembangkan fitur autentikasi dan otorisasi pengguna berbasis *role*, sehingga seluruh proses pengelolaan sistem dapat dibatasi sesuai hak akses masing-masing pengguna, yaitu admin, dosen, dan mahasiswa. Riwayat pengerjaan *sprint* pertama dapat dilihat pada gambar 4.6 sampai gambar 4.9



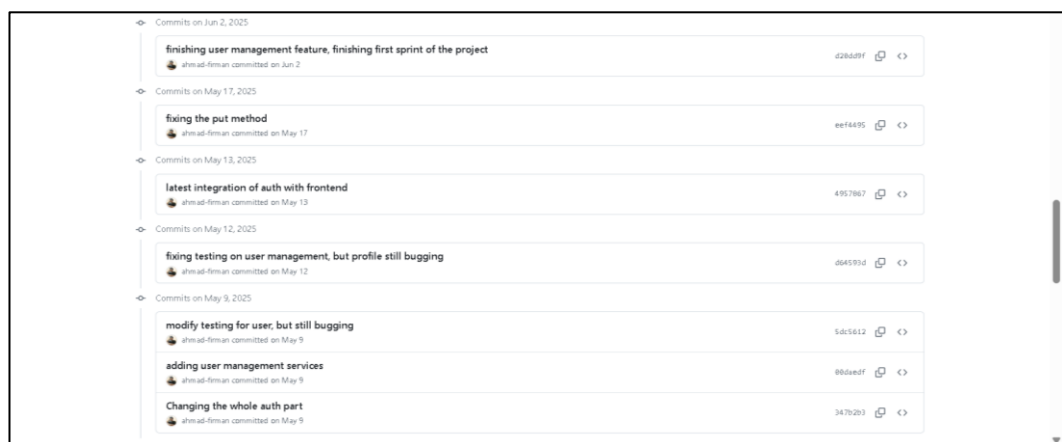
Gambar 4. 6 Riwayat *Commit Github* Pertama



Gambar 4. 7 Riwayat *Commit Github* untuk Fitur Data Master



Gambar 4. 8 Riwayat *Commit Github* untuk Penambahan *Frontend* Data Master



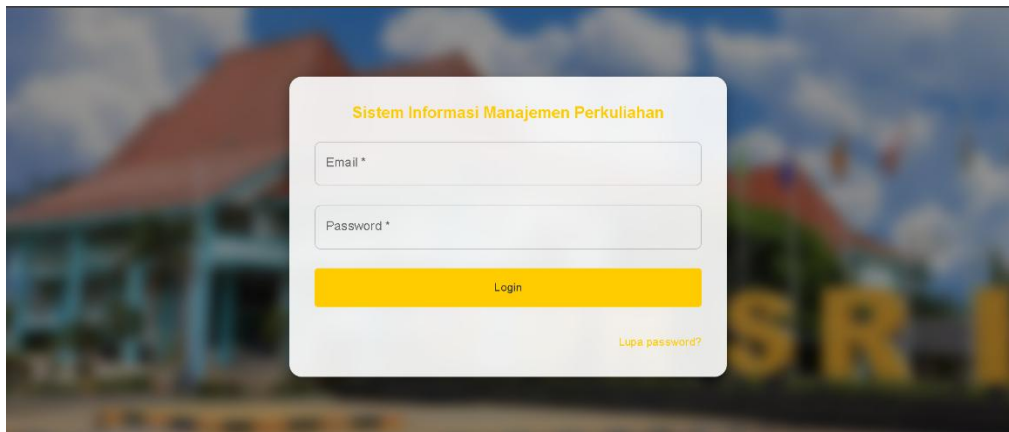
Gambar 4. 9 Riwayat *Commit Github* Berupa Penyelesaian *Sprint* Pertama

Dari gambar 4.6 sampai gambar 4.9 dapat kita lihat bahwa fitur yang dikembangkan:

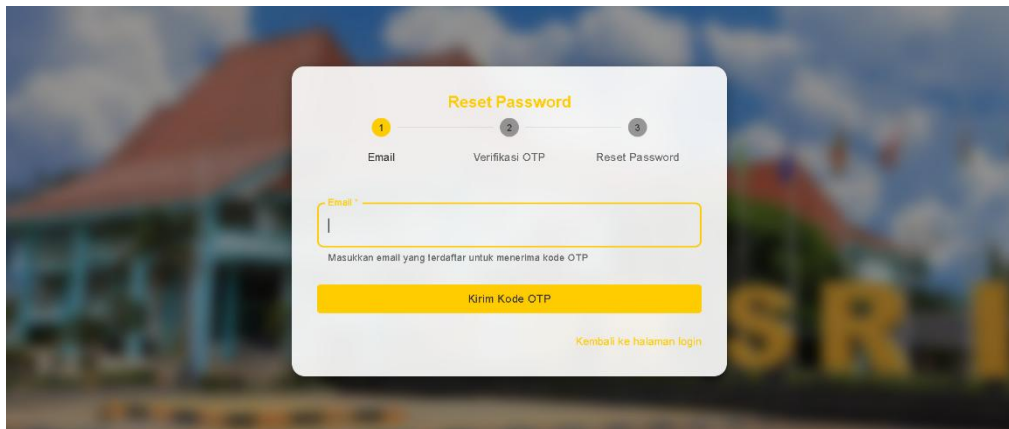
1. *Login dan logout menggunakan token JWT (session)*
2. *Pengelolaan role-based access (admin, dosen, mahasiswa)*
3. *Reset password*

4. Manajemen pengguna (*create, update, delete*)

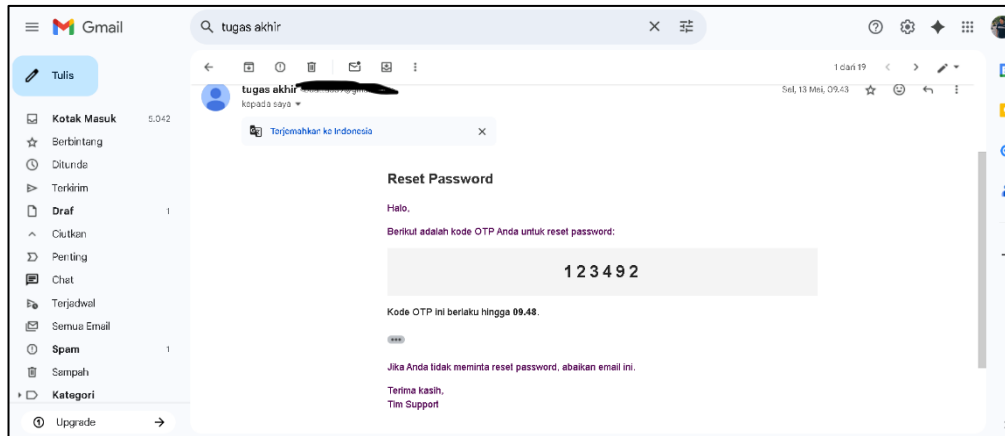
Gambar 4.10 sampai gambar 4.13 adalah gambaran dari *sprint* ini, hasilnya sistem autentikasi yang telah berjalan dengan baik dan terintegrasi dengan *frontend*. Pengguna dapat melakukan *login sesuai peran*, dan *dashboard* akan menyesuaikan berdasarkan hak akses. Struktur proyek juga telah dirancang menggunakan pendekatan *monorepo* agar pengembangan *frontend* dan *backend* berjalan paralel. Sebagian besar fitur *user management* berhasil diuji dan berfungsi dengan baik.



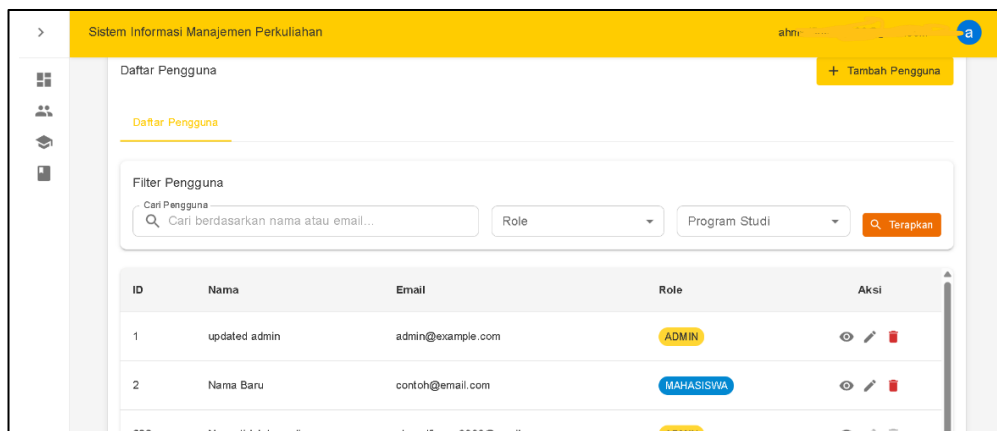
Gambar 4. 10 Halaman *Login*



Gambar 4. 11 Halaman *Reset Password*



Gambar 4. 12 Hasil Email OTP yang Dikirim

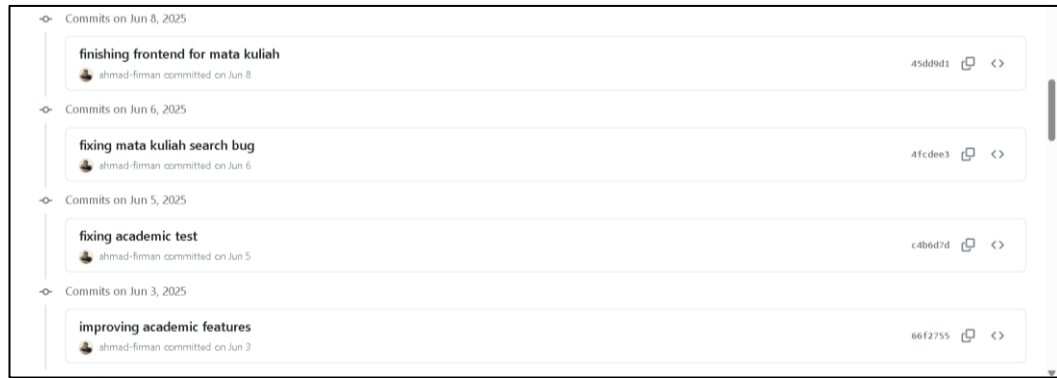


Gambar 4. 13 Halaman Manajemen Pengguna

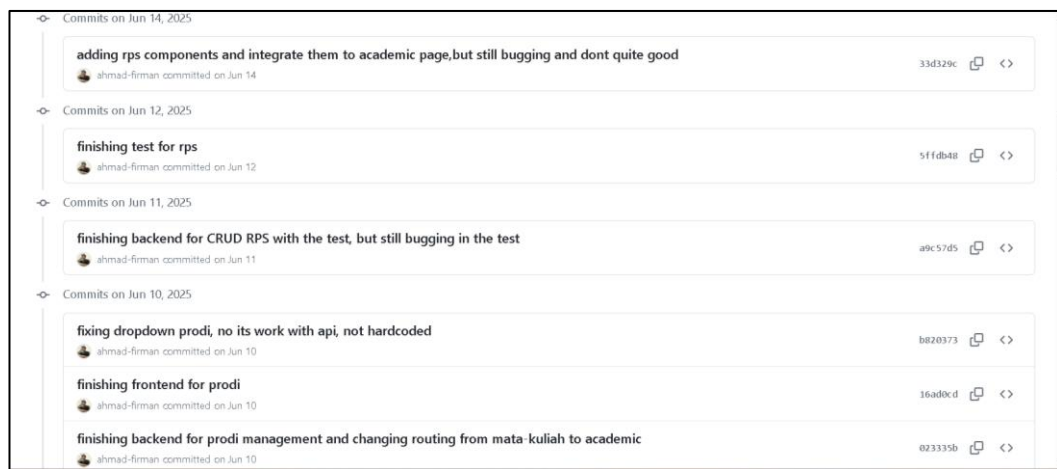
Evaluasi dilakukan pada akhir *sprint* bersama dosen pembimbing. Beberapa kendala teknis ditemukan, terutama dalam proses integrasi awal dan pengujian fitur *reset password*. Hal ini tercermin dalam beberapa *commit* yang berisi perbaikan dan *debugging*. Meskipun demikian, secara keseluruhan fitur utama dinilai telah memenuhi kebutuhan, dan sistem dianggap siap untuk dikembangkan lebih lanjut pada tahap *sprint* kedua.

4.3.1.2 *Sprint* Kedua: Pengelolaan RPS

Sprint kedua dilaksanakan pada tanggal 3 Juni hingga 27 Juni 2025. Fokus utama *sprint* ini adalah pengembangan modul Rencana Pembelajaran Semester (RPS), yang merupakan inti dari sistem yang dirancang. Tujuan *sprint* ini adalah membangun fitur CRUD RPS dan *export PDF* sesuai dengan struktur template resmi dari fakultas, serta memungkinkan pengelolaan komponen-komponen seperti CPMK, Sub-CPMK, dan metode pembelajaran.



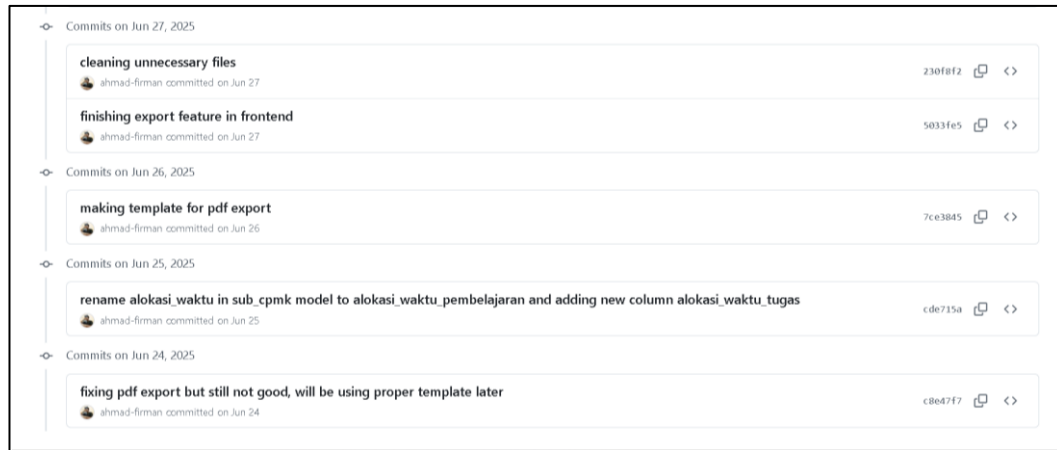
Gambar 4. 14 Riwayat *Commit Github* pada Awal *Sprint Kedua*



Gambar 4. 15 Riwayat *Commit Github* pada Penyelesaian Fitur Prodi



Gambar 4. 16 Riwayat *Commit Github* pada Penyelesaian Fitur Ekspor RPS

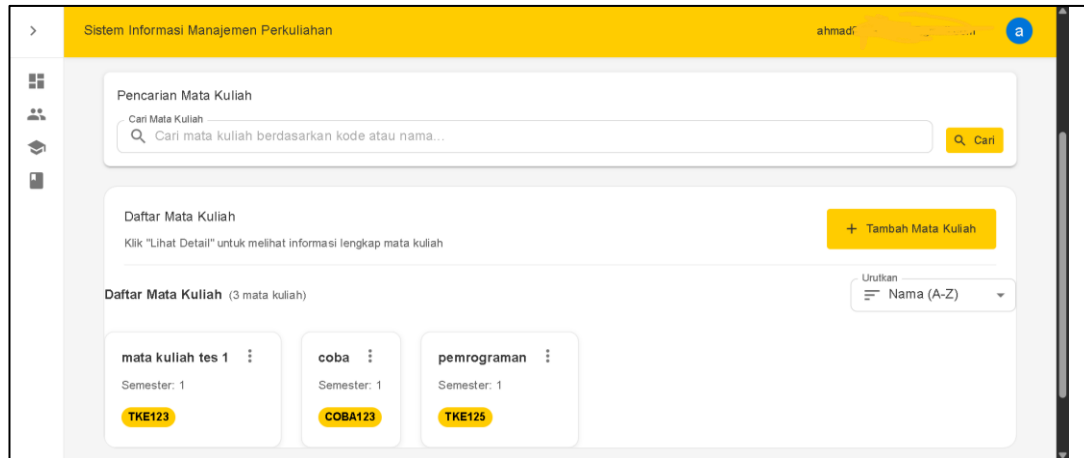


Gambar 4. 17 Riwayat *Commit Github* pada Penyelesaian *Sprint* Kedua

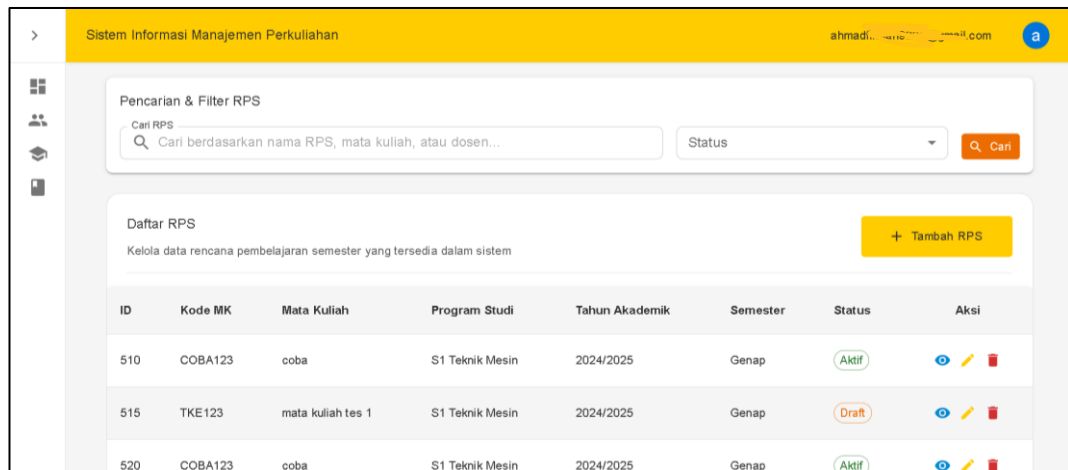
Berdasarkan gambar 4.14 sampai gambar 4.17 dapat kita simpulkan fitur yang dikembangkan adalah:

1. CRUD RPS (*input*, pembaruan, penghapusan, dan tampilan RPS)
2. Manajemen CPMK dan Sub-CPMK
3. *Input* metode pembelajaran, jenis penilaian, bobot, dan alokasi waktu
4. Relasi data RPS dengan data program studi dan mata kuliah
5. Fitur ekspor RPS ke *PDF*

Gambar 4.18, 4.19, dan gambar 4.20 adalah gambaran hasil dari *sprint* ini, tersusunnya modul manajemen RPS yang dapat digunakan oleh dosen untuk menyusun RPS secara digital dan terstruktur. Pengguna dapat mengelola CPMK, sesi pembelajaran, dan asesmen sesuai dengan struktur RPS resmi. Modul juga berhasil terhubung dengan data mata kuliah dan program studi secara dinamis melalui API. Fitur ekspor ke *PDF* juga sudah mulai diimplementasikan, meskipun masih dalam tahap penyempurnaan.



Gambar 4. 18 Halaman Manajemen Mata Kuliah



Gambar 4. 19 Halaman Manajemen RPS

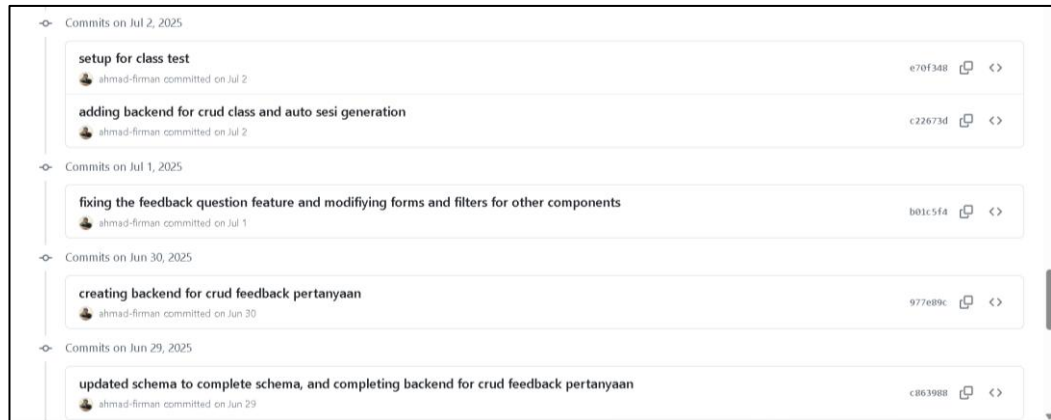
	UNIVERSITAS SRIWIJAYA Fakultas Teknik Jurusan Teknik Mesin Program Studi S1 Teknik Mesin				
RENCANA PEMBELAJARAN SEMESTER					
A. IDENTITAS MATA KULIAH					
Nama Mata Kuliah	Kode Mata Kuliah	Bahan Kajian	SKS Kuliah Praktikum	Semester	Tanggal Revisi
coba	COBA123	coba	1 0	1	27/6/2025
Deskripsi Mata Kuliah		coba			
Capaian Pembelajaran Mata Kuliah (CPMK)		Mahasiswa mampu menguasai konsep dasar Aplikasi lagi CPMK001: Mahasiswa mampu memahami konsep dasar algoritma CPMK002: Mahasiswa mampu memahami struktur data dasar CPMK-3: testestes CPMK-4: testestes			
Tim Pengajar		Dosen test Dosen coba		Ketua Tim Pengajar	Dosen test
				Instruktur	Dosen coba
Otorisasi		Tanda Tangan Ketua Program Studi		Tanda Tangan Wakil Dekan Bidang Akademik	

Gambar 4. 20 Hasil Ekspor RPS

Evaluasi terhadap *sprint* ini menunjukkan bahwa sebagian besar fitur utama telah berhasil dibangun dan terintegrasi, meskipun ditemukan beberapa kendala pada proses *testing* dan tampilan form RPS yang kompleks. Berdasarkan *commit*, beberapa perbaikan signifikan dilakukan terhadap struktur form, routing, dan template ekspor *PDF*. *Sprint* dinilai berhasil memenuhi target utama meskipun masih dibutuhkan perbaikan *minor* pada sisi antarmuka dan format hasil ekspor untuk tahap selanjutnya.

4.3.1.3 *Sprint* Ketiga: Modul kelas, pencatatan kehadiran dan *feedback*

Sprint ketiga berlangsung dari 29 Juni hingga 16 Juli 2025 dan difokuskan pada pengembangan fitur inti yang berkaitan dengan pelaksanaan kegiatan belajar mengajar. Tujuan dari *sprint* ini adalah membangun modul manajemen kelas, pencatatan kehadiran mahasiswa per sesi, serta pemberian umpan balik terhadap pembelajaran. Seluruh aktivitas ini bertujuan untuk mengintegrasikan pelaksanaan pembelajaran langsung dengan RPS yang telah disusun sebelumnya.



Gambar 4. 21 Riwayat *Commit Github* pada Awal *Sprint* Ketiga



Gambar 4. 22 Riwayat *Commit Github* pada Pembuatan Kelas dan Sesi



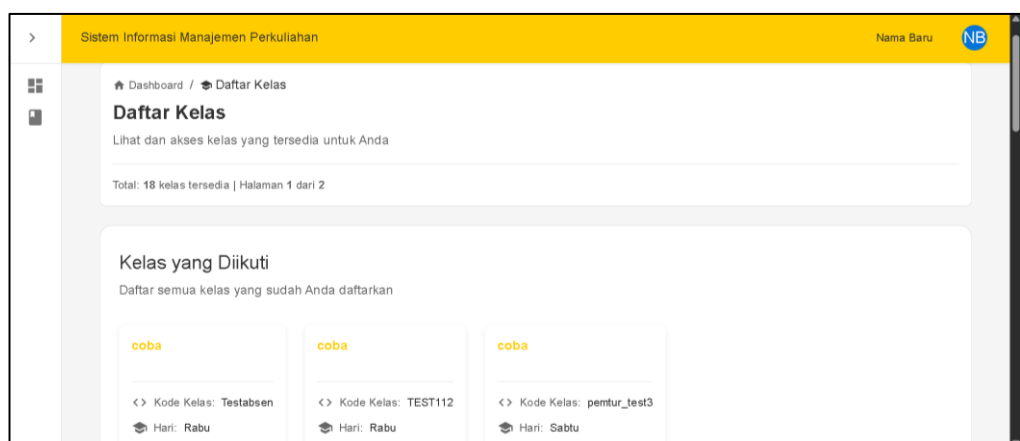
Gambar 4. 23 Riwayat *Commit Github* pada Pembuatan Fitur Kehadiran dan *Feedback*



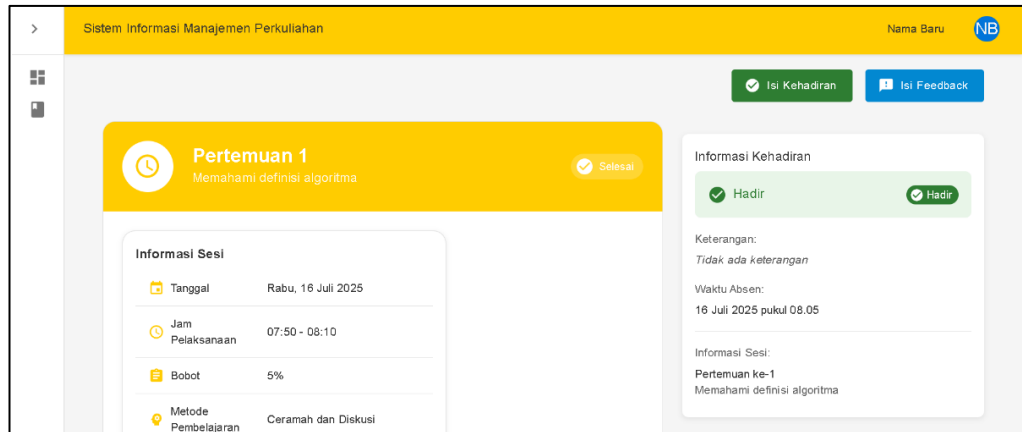
Gambar 4. 24 Riwayat *Commit Github* pada Penyelesaian *Sprint* Ketiga

Pada gambar 4.21 sampai gambar 4.24 dapat kita lihat bahwa fitur yang dikembangkan adalah:

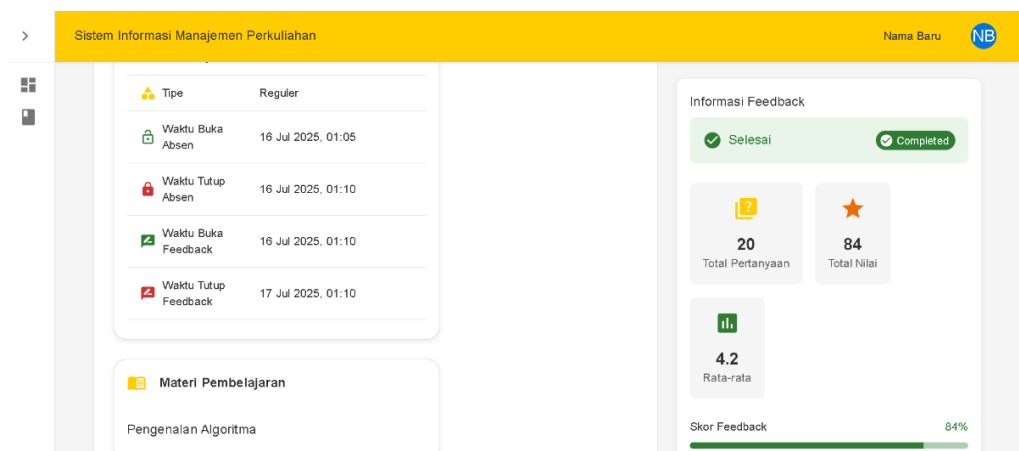
1. CRUD kelas dan sesi perkuliahan
2. Sistem generasi sesi perkuliahan otomatis berdasarkan struktur RPS
3. Pendaftaran mahasiswa ke dalam kelas (*enrollment*)
4. *Pencatatan* kehadiran mahasiswa per sesi
5. *Input* dan pengelolaan *feedback* dari mahasiswa terhadap sesi pembelajaran
6. CRUD pertanyaan umpan balik
7. Integrasi tampilan daftar mahasiswa dan sesi kehadiran
8. Penjadwalan sesi otomatis dan perbaikan data waktu
9. Optimalisasi performa melalui *indexing* dan penyesuaian *cron job*



Gambar 4. 25 Halaman Kelas



Gambar 4. 26 Halaman Sesi



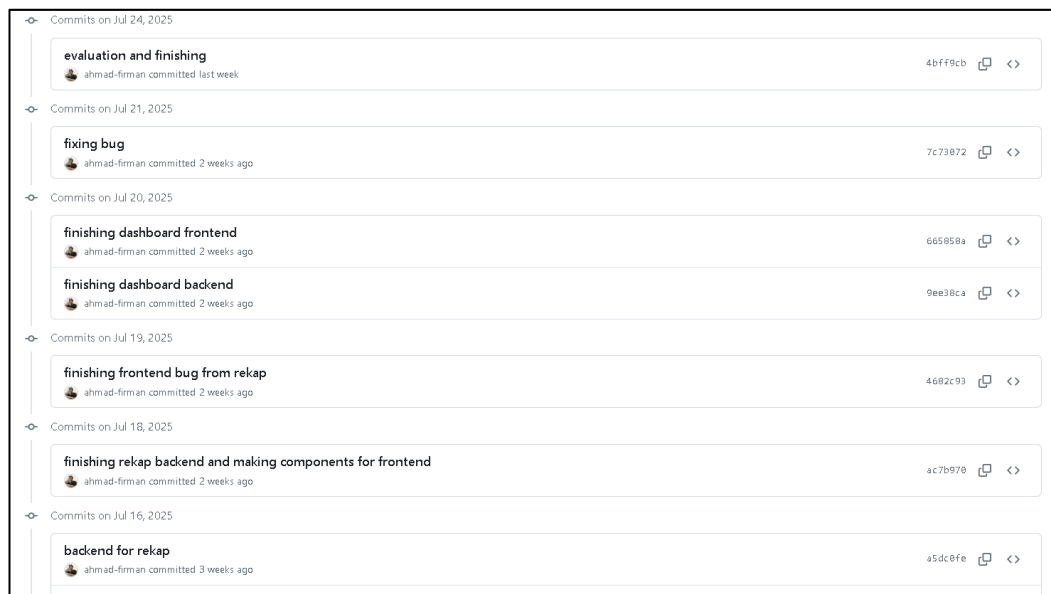
Gambar 4. 27 Detail Halaman Sesi

Gambar 4.25, 4.26 dan gambar 4.27 adalah hasil dari *sprint* ini, sistem yang telah mampu mengelola aktivitas pembelajaran secara lebih terstruktur. Dosen dapat membuat kelas dan sistem secara otomatis menghasilkan sesi perkuliahan sesuai jumlah pertemuan dan konten dalam RPS. Mahasiswa dapat dicatat ke dalam kelas dan mengisi kehadiran secara *online*. Selain itu, fitur *feedback* telah tersedia dan dapat diisi oleh mahasiswa setelah sesi berlangsung.

Evaluasi menunjukkan bahwa sebagian besar fitur berhasil diimplementasikan, meskipun beberapa *bug* ditemukan pada proses *enrollment* dan manajemen waktu sesi. Perbaikan dilakukan secara bertahap hingga *backend* dan *frontend* untuk presensi serta *feedback* terintegrasi dengan baik. Dosen pembimbing memberikan masukan terhadap alur *form* dan *usability*, yang menjadi bahan perbaikan untuk *sprint* terakhir.

4.3.1.4 *Sprint* Keempat: Rekapitulasi dan *Dashboard*

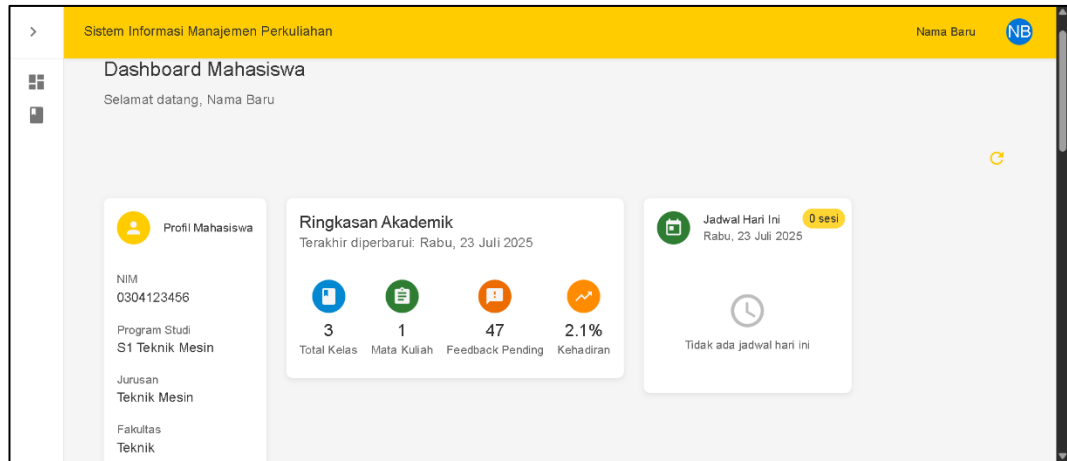
Sprint keempat dilaksanakan pada tanggal 16 Juli hingga 24 Juli 2025, dengan tujuan untuk menyatukan seluruh modul yang telah dikembangkan pada *sprint* sebelumnya, serta membangun fitur pelaporan dan rekapitulasi pembelajaran. Fokus utama *sprint* ini adalah memastikan integrasi data antarmodul (RPS, kelas, presensi, dan *feedback*), sekaligus memberikan visualisasi yang informatif melalui *dashboard* dan fitur ekspor hasil pembelajaran.



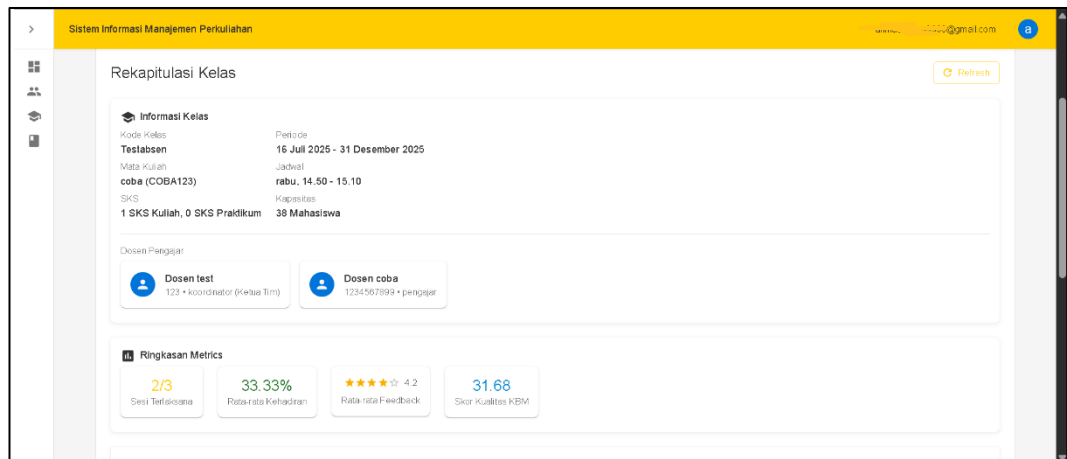
Gambar 4. 28 Riwayat *Commit Github* pada *Sprint* Keempat

Dari gambar 4.28, fitur yang dikembangkan adalah:

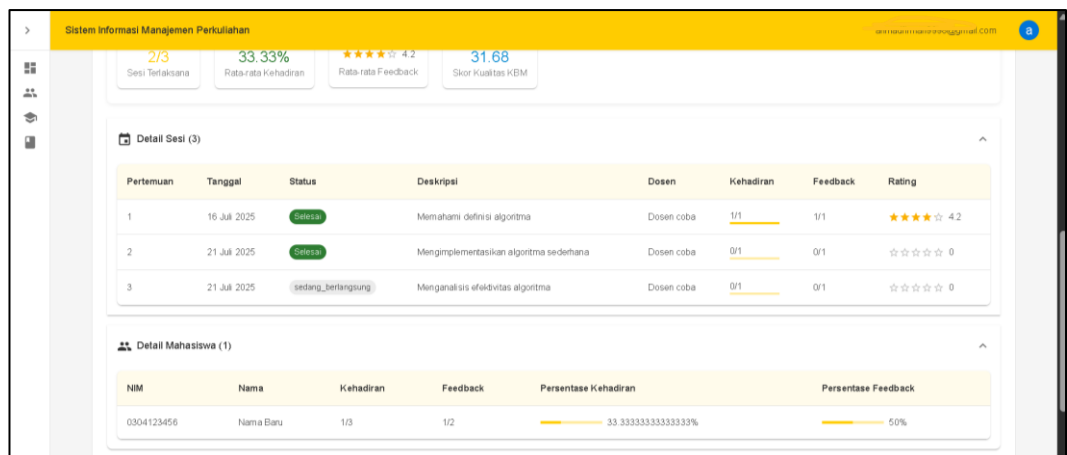
1. Integrasi data kelas, sesi, presensi, dan *feedback* ke dalam sistem rekap
2. CRUD dan tampilan rekapitulasi pelaksanaan pembelajaran
3. Penyusunan komponen *frontend* untuk tampilan rekap
4. *Dashboard* pelaporan untuk dosen dan admin
5. *Backend dashboard*
6. Perbaikan *bug* pada tampilan *frontend* untuk rekap
7. Optimalisasi akses data rekap dan penghitungan performa sistem



Gambar 4. 29 Halaman *Dashboard* Mahasiswa



Gambar 4. 30 Halaman Rekapitulasi



Gambar 4. 31 Halaman Rekapitulasi dengan Detail



Gambar 4. 32 Hasil Ekspor Rekap dalam *PDF*

Gambar 4.29 sampai gambar 4.32 adalah Gambaran hasil dari *sprint* ini, tersusunnya fitur rekapitulasi pelaksanaan pembelajaran yang menggabungkan data presensi, aktivitas pembelajaran, dan capaian RPS dalam satu tampilan terstruktur. Dosen dan admin dapat memantau kesesuaian pelaksanaan perkuliahan dengan RPS melalui *dashboard* yang dilengkapi indikator visual dan laporan singkat. Komponen rekap juga memungkinkan ekspor data untuk keperluan dokumentasi atau evaluasi akademik lebih lanjut.

Evaluasi akhir menunjukkan bahwa sistem telah bekerja secara menyeluruh dan menyatu, meskipun beberapa perbaikan *minor* masih dilakukan terhadap tampilan antarmuka rekap. Dosen pembimbing menyatakan bahwa fitur pelaporan sudah memenuhi kebutuhan dasar pengguna, dan *sprint* keempat menjadi penutup tahap pengembangan utama sebelum masuk ke proses finalisasi dokumentasi dan penyempurnaan.

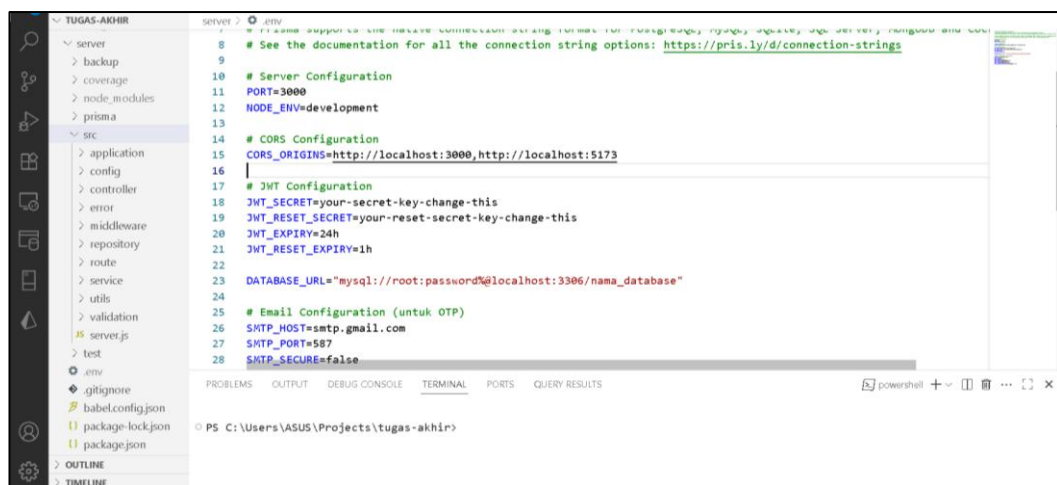
4.3.2 Konfigurasi *Environment Development*

Pengembangan sistem dilakukan menggunakan lingkungan *monorepo* dengan manajer paket *pnpm*, yang memungkinkan pengelolaan *frontend* dan *backend* dalam satu repositori proyek secara terstruktur. Dengan konfigurasi ini, pengembangan dapat dilakukan secara paralel antara sisi *frontend* (*React* dengan *Vite*) dan *backend* (*Node.js* dengan *Express*), sehingga proses integrasi, pengujian, dan evaluasi menjadi lebih efisien. Seperti yang dapat dilihat pada gambar 4.33, struktur *monorepo* ini juga memudahkan penggunaan *dependencies* bersama serta pengaturan *environment* yang konsisten di seluruh bagian sistem.



Gambar 4. 33 Konfigurasi *Monorepo* PNPM

Seperti yang telah dijelaskan sebelumnya pada sub bab 2.7.1, teknologi *backend* yang digunakan dalam pengembangan sistem adalah *Node.js*. Struktur proyek dijelaskan pada gambar 4.34 dengan menggunakan pendekatan modular dan memanfaatkan *environment variable* melalui file *.env*. Pembagian struktur folder mencakup *application*, *config*, *controller*, *error*, *middleware*, *repository*, *route*, *service*, *utils*, dan *validation* yang seluruhnya disimpan pada folder *src*. Arsitektur modular ini memudahkan proses pengembangan dan *debugging* ketika terjadi kesalahan dalam sistem.



Gambar 4. 34 Konfigurasi *File .env Backend*

File .env pada gambar 4.34 dalam folder *server* ini mendefinisikan beberapa konfigurasi *port* yang penting untuk operasional sistem. Penempatan *port* 3000 sebagai *port server backend* yang memungkinkan aplikasi *frontend* berkomunikasi dengan API melalui *endpoint* yang konsisten. Basis data dikonfigurasi pada *port* 3306 sebagai *port default MySQL*, tujuannya memastikan koneksi *database* dapat

diakses dengan benar oleh aplikasi. Sementara itu, *port* 587 dipakai oleh layanan *email* SMTP untuk layanan pengiriman OTP dalam fitur *reset password* yang telah dikembangkan pada bagian *sprint* pertama di 4.3.1.1. Pendefinisian *port-port* di *file .env* ini memberikan fleksibilitas untuk mengubah pengaturan tanpa memodifikasi kode aplikasi dan memastikan keamanan dengan memisahkan konfigurasi sensitif dari kode aplikasi utama.

4.3.3 Struktur Tabel Database

Database sistem informasi rekapitulasi kegiatan belajar mengajar ini menggunakan *MySQL* sebagai sistem manajemen basis data dengan *Prisma* sebagai *Object-Relational Mapping* (ORM). Struktur *database* dirancang untuk mendukung manajemen akademik yang komprehensif dengan fokus pada rekapitulasi dan pengawasan kegiatan pembelajaran.

	id	nama_prodi	jurusan	fakultas
▶	1	S1 Teknik Mesin	Teknik Mesin	Teknik
	517	S1 Teknik Kimia	Teknik Kimia	Teknik
	519	S1 Teknik Elektro	Teknik Elektro	Teknik
	1066	S1 Teknik Sipil	Teknik Sipil	Teknik
	1073	S1 Teknik Pertambangan	Teknik Pertambangan	Teknik
	1074	S1 Arsitektur	Arsitektur	Teknik

Gambar 4. 35 Tabel Prodi

	id	password	role	token	tokenExpiry	createdAt	updatedAt
▶	1	\$2b\$10\$U6p9O0SDFT.FBMVSeArhMLucaG...	ADMIN	eyJhbG90aWZ1IHNisR5cC16pWC9.eyJpZ...	2025-05-14 00:42:09.734	2025-05-08 09:00:00.413	2025-07-19 03:10:27.728
	2	\$2b\$10\$Yg9t9F7y38PeRdAS9MOavVLRqzC...	MAHASISWA	eyJhbG90aWZ1IHNisR5cC16pWC9.eyJpZ...	2025-07-04 00:25:01.294	2025-05-08 09:00:00.413	2025-07-03 00:25:01.297
	636	\$2b\$10\$2RMCFdFVy740K0t9J5w5.022pnBp...	ADMIN	eyJhbG90aWZ1IHNisR5cC16pWC9.eyJpZ...	2025-07-24 00:23:31.559	2025-05-08 13:00:52.128	2025-07-23 00:23:31.560
	3228	\$2b\$10\$6s1vc7dbvUX1Y3d3ZNeC.Z5H6.UVCS...	DOSEN	[REDACTED]	[REDACTED]	2025-05-17 01:41:29.936	2025-07-21 02:56:20.632
	3229	\$2b\$10\$vcvU2OFF.8ghNeOb6FWLOV9tBtEdb...	DOSEN	[REDACTED]	[REDACTED]	2025-05-17 01:43:09.045	2025-05-17 01:43:09.045
	3379	\$2b\$10\$3gwCTAJWn3H0m3TUfCQW4g...	ADMIN	[REDACTED]	[REDACTED]	2025-06-10 06:57:53.255	2025-06-10 06:57:53.255
	3720	\$2b\$10\$FSpndRCFYEHfmy60NEu.x9eyf35Ns...	ADMIN	[REDACTED]	[REDACTED]	2025-07-01 01:56:31.070	2025-07-20 07:12:23.401
	3721	\$2b\$10\$9oe99dhtTmFm70J7PeeQ9pkuR5...	MAHASISWA	eyJhbG90aWZ1IHNisR5cC16pWC9.eyJpZ...	2025-07-14 01:23:11.123	2025-07-12 02:33:51.308	2025-07-13 01:23:11.125
	3722	\$2b\$10\$WLYucJ54nNlm30E2UOQ8yP3dRh...	MAHASISWA	[REDACTED]	[REDACTED]	2025-07-13 01:22:19.642	2025-07-23 00:23:12.233
	3723	\$2b\$10\$m18hPwQ38wB5pQ39g4uez255kqB...	ADMIN	[REDACTED]	[REDACTED]	2025-07-20 07:12:13.211	2025-07-20 07:53:36.858

Gambar 4. 36 Tabel User

Sistem ini memiliki beberapa tabel *master* yang menjadi fondasi struktur data. Tabel *User* pada gambar 4.36 berfungsi sebagai "*single source of truth*" untuk autentikasi dengan menyimpan informasi *login*, *role*, dan *token management* untuk semua pengguna sistem. Tabel *Prodi* pada gambar 4.35 menyimpan informasi program studi yang mencakup nama program studi, jurusan, dan fakultas dengan *default* nilai "fakultas teknik". Tabel *Admin*, *Dosen*, dan *Mahasiswa* merupakan

tabel yang berelasi dengan tabel *User* melalui *foreign key userId*, dimana setiap *record* memiliki hubungan *one-to-one* dengan *User* dan *many-to-one* dengan *Prodi*.

id	kode	nama	sks_jumlah	sks_praktikum	semester	bahan_kajian	deskripsi	id_prodi	createdAt	updatedAt
557	TKE123	mata kuliah tes 1	2	2	1	Algoritma dan Pemrograman	Mata kuliah pemrograman dasar pe...	1	2025-06-06 07:38:46.992	2025-06-08 07:22:55.356
559	COBA123	coba	1	0	1	coba	coba	1	2025-06-08 07:23:37.178	2025-06-08 07:23:37.178
970	TKE125	pemrograman	1	0	1	testes	testes	1	2025-06-10 02:12:00.494	2025-06-10 02:12:00.494

Gambar 4. 37 Tabel RPS

id	id_mata_kuliah	tanggal_revisi	tahun_akademik	semester	cpl	status	createdAt	updatedAt	deleted_at
508	557	2024-01-15 00:00:00.000	2023/2024	Ganjil	Mahasiswa mampu menguasai konsep dasar pe...	draft	2025-06-12 06:20:48.879	2025-06-12 06:20:48.879	
509	557	2024-01-15 00:00:00.000	2023/2024	Ganjil	Mahasiswa mampu menguasai konsep dasar pe...	draft	2025-06-12 06:21:17.363	2025-06-12 06:24:47.894	2025-06-12 06:24:47.875
510	559	2025-06-27 00:00:00.000	2024/2025	Genap	Mahasiswa mampu menguasai konsep dasar Apl...	active	2025-06-12 06:21:31.215	2025-06-27 06:47:44.667	

Gambar 4. 38 Tabel RPS Bagian Lanjutan

Struktur akademik direpresentasikan melalui Tabel MataKuliah yang menyimpan informasi mata kuliah termasuk kode, nama, SKS kuliah, SKS praktikum, semester, bahan kajian, dan deskripsi. Tabel RPS seperti pada gambar 4.37 dan gambar 4.38 berelasi dengan MataKuliah dan menyimpan informasi RPS termasuk tahun akademik, semester, CPL (Capaian Pembelajaran Lulusan), dan status dengan kemampuan *soft delete* melalui *field deleted_at*. Tabel CPMK (Capaian Pembelajaran Mata Kuliah) dan SubCPMK membentuk hierarki pembelajaran dimana SubCPMK menyimpan detail pembelajaran seperti materi, referensi, metode pembelajaran, indikator, dan alokasi waktu.

kode_kelas	id_mata_kuliah	id_rps	tanggal_mula	tanggal_berakhir	createdAt	updatedAt	hari_pelaksanaan	jam_berakhir	jam_pelaksanaan	kapasitas	status
PEMULR_Alayo	559	510	2025-08-12 00:00:00.000	2025-12-23 00:00:00.000	2025-07-02 13:31:13.905	2025-07-02 13:36:37.188	kamis	10:00:00	06:00:00	60	
PEMULR_Blayo	559	510	2025-08-12 00:00:00.000	2025-12-23 00:00:00.000	2025-07-02 13:37:02.292	2025-07-02 13:40:18.297	selasa	10:00:00	06:00:00	60	
PEMULR_Abuluk	559	510	2025-08-12 00:00:00.000	2025-12-23 00:00:00.000	2025-07-03 00:17:36.773	2025-07-03 00:49:56.021	kamis	10:00:00	06:00:00	60	20

Gambar 4. 39 Tabel Kelas

	id	id_mahasiswa	id_kelas	enrolled_at
▶	3	030412345	9	2025-07-12 02:59:44.952
	9	0304123456	9	2025-07-13 02:44:40.448
	12	030412345	31	2025-07-13 07:58:30.513
	15	0304123456	31	2025-07-14 02:43:12.560
	21	0304123456	32	2025-07-16 00:47:24.052
•	NULL	NULL	NULL	NULL

Gambar 4. 40 Tabel Mahasiswa Kelas (Enrolment)

Tabel Kelas seperti pada gambar 4.39 menjadi pusat operasional pembelajaran yang menyimpan jadwal, kapasitas, dan periode pelaksanaan dengan relasi ke MataKuliah dan RPS. Tabel Sesi merepresentasikan pertemuan pembelajaran dengan informasi detail seperti pertemuan ke-, tanggal, jam mulai-berakhir, status sesi, tipe sesi (reguler, UTS, UAS), dan waktu buka atau tutup absen serta *feedback*. Tabel MahasiswaKelas pada gambar 4.40 berfungsi sebagai *junction table* untuk relasi *many-to-many* antara Mahasiswa dan Kelas dengan informasi waktu *enrollment*.

	id	id_mahasiswa_kelas	id_sesi	status	waktu_absen	keterangan	createdAt	updatedAt
▶	33	3	81	absen	NULL	NULL	2025-07-12 02:59:44.988	2025-07-12 02:59:44.988
	34	3	82	absen	NULL	NULL	2025-07-12 02:59:44.988	2025-07-12 02:59:44.988
	35	3	83	absen	NULL	NULL	2025-07-12 02:59:44.988	2025-07-12 02:59:44.988
	36	3	84	absen	NULL	NULL	2025-07-12 02:59:44.988	2025-07-12 02:59:44.988
	37	3	85	absen	NULL	NULL	2025-07-12 02:59:44.988	2025-07-12 02:59:44.988
	38	3	86	absen	NULL	NULL	2025-07-12 02:59:44.988	2025-07-12 02:59:44.988
	39	3	87	absen	NULL	NULL	2025-07-12 02:59:44.988	2025-07-12 02:59:44.988
	40	3	88	absen	NULL	NULL	2025-07-12 02:59:44.988	2025-07-12 02:59:44.988
	41	3	89	absen	NULL	NULL	2025-07-12 02:59:44.988	2025-07-12 02:59:44.988
	42	3	90	absen	NULL	NULL	2025-07-12 02:59:44.988	2025-07-12 02:59:44.988
	43	3	91	absen	NULL	NULL	2025-07-12 02:59:44.988	2025-07-12 02:59:44.988
	44	3	92	absen	NULL	NULL	2025-07-12 02:59:44.988	2025-07-12 02:59:44.988
	45	3	93	absen	NULL	NULL	2025-07-12 02:59:44.988	2025-07-12 02:59:44.988
	46	3	94	absen	NULL	NULL	2025-07-12 02:59:44.988	2025-07-12 02:59:44.988
	47	3	95	absen	NULL	NULL	2025-07-12 02:59:44.988	2025-07-12 02:59:44.988
	48	3	96	absen	NULL	NULL	2025-07-12 02:59:44.988	2025-07-12 02:59:44.988
	129	9	81	absen	NULL	NULL	2025-07-13 02:44:40.556	2025-07-13 02:44:40.556
	130	9	82	absen	NULL	NULL	2025-07-13 02:44:40.556	2025-07-13 02:44:40.556

Gambar 4. 41 Tabel Kehadiran

	id	id_dosen	id_rps	id_kelas	peran	is_ketua_tim
▶	833	1234567899	508	HULL	Koordinator	1
	834	123	508	HULL	Pengajar	0
	835	1234567899	509	HULL	Koordinator	1
	836	123	509	HULL	Pengajar	0
	847	1234567899	511	HULL	Koordinator	1
	848	123	511	HULL	Pengajar	0
	849	1234567899	512	HULL	Koordinator	1
	850	123	512	HULL	Pengajar	0
	851	1234567899	513	HULL	Koordinator	1
	852	123	513	HULL	Pengajar	0
	853	1234567899	514	HULL	Koordinator	1
	854	123	514	HULL	Pengajar	0
	857	1234567899	516	HULL	Koordinator	1
	858	123	516	HULL	Pengajar	0
	859	1234567899	517	HULL	Koordinator	1
	860	123	517	HULL	Pengajar	0
	861	1234567899	518	HULL	Koordinator	1
	862	123	518	HULL	Pengajar	0

dosen_pengajar 11 x

Gambar 4. 42 Tabel Dosen Pengajar

Sistem pengawasan pembelajaran didukung oleh Tabel Kehadiran seperti pada gambar 4.41 yang menyimpan status kehadiran mahasiswa per sesi dengan *enum* StatusKehadiran (hadir, absen, sakit, izin, terlambat) dan *timestamp* waktu absen. Tabel *Feedback* dan *related tables* (*Feedback* Pertanyaan, *Feedback* Jawaban) mengelola sistem evaluasi pembelajaran dengan kategorisasi *feedback* berdasarkan materi, penyampaian, metode, dan fasilitas. Tabel DosenPengajar seperti pada gambar 4.42 mengatur *assignment* dosen ke RPS atau Kelas dengan informasi peran dan status ketua tim.

	id	total_pesi	total_pesi_terlaksana	total_mahasiswa	total_presensi	total_feedback_submitted	rata_rata_kehadiran	rata_rata_feedback	rata_rata_feedback_materi	rata_rata_feedback_penyampaian	rata_rata_fee
▶	3	1	1	1	1	1	100	4.2	4.2	4.2	4.2
	7	1	1	1	1	1	100	4.2	4.2	4.2	4.2
	9	2	2	1	2	1	50	4.2	4.2	4.2	4.2
*											

Gambar 4. 43 Tabel Rekapitulasi *Snapshot*

	rata_rata_feedback_metode	rata_rata_feedback_fasilitas	persentase_kesesuaian_rps	skor_juwalas_jbm	tanggal_snapshot	periode_mulai	periode_selesai	generated_by	purpose	createdAt
▶	4.2	4.2	100	61.68	2025-07-18 07:44:34.299	2025-07-16 00:00:00.000	2025-07-18 07:44:34.246	636	rekap	2025-07-11
	4.2	4.2	100	61.68	2025-07-19 01:46:09.557	2025-07-16 00:00:00.000	2025-07-19 01:46:09.524	636	rekap	2025-07-11
	4.2	4.2	100	46.68	2025-07-21 02:14:32.294	2025-07-16 00:00:00.000	2025-07-21 02:14:32.219	636	rekap	2025-07-2
*										

Gambar 4. 44 Tabel Rekapitulasi *Snapshot* Bagian Lanjutan

Tabel Rekapitulasi *Snapshot* seperti pada gambar 4.43 dan 4.44 merupakan tabel khusus untuk menyimpan hasil rekapitulasi KBM yang telah dihitung. Tabel ini menyimpan berbagai metrik seperti total sesi, total sesi terlaksana, total mahasiswa, rata-rata kehadiran, rata-rata *feedback* per kategori, persentase kesesuaian RPS, dan skor kualitas KBM. Tabel ini juga menyimpan detail data dalam format JSON pada *field* *detail_sesi_data* dan *detail_mahasiswa_data* untuk keperluan *export* dan analisis mendalam.

Seluruh tabel telah dioptimasi dengan *indexing* yang strategis untuk mendukung performa *query dashboard* dan operasi *batch*. *Index* dibuat berdasarkan pola *query* umum seperti *filtering* berdasarkan periode, pencarian berdasarkan *role* dan status, serta agregasi data untuk *dashboard analytics*. *Composite index* diterapkan pada kombinasi *field* yang sering digunakan bersama dalam *query* kompleks, terutama untuk operasi rekapitulasi yang memerlukan *join* beberapa tabel.

4.4 Hasil Pengujian Sistem

Pengujian *blackbox* dilakukan untuk memvalidasi fungsionalitas sistem informasi rekapitulasi kegiatan belajar mengajar berdasarkan requirement yang telah ditetapkan dengan fokus pada aspek *input* dan *output* tanpa memperhatikan struktur internal kode program. Pengujian dibagi menjadi delapan kategori utama yang mencakup autentikasi, manajemen data *master*, pengelolaan RPS, pengelolaan kelas dan sesi, sistem absensi, evaluasi *feedback*, pembuatan rekapitulasi, serta *dashboard* statistik. Setiap kategori dirancang dengan skenario *positive* dan *negative testing* menggunakan sampel data yang representatif, dengan dokumentasi hasil pengujian berupa perbandingan antara *expected output* dan *actual output* yang dihasilkan sistem.

Tabel 4. 2 Pengujian Fungsionalitas *Login dan Autentikasi*

No	Fitur yang Diuji	Deskripsi Pengujian	<i>Input</i>	<i>Expected output</i>	<i>Actual output</i>	Status
1	<i>Login</i>	Pengguna memasukkan <i>email</i> dan <i>password</i> yang valid	<i>Email & Password valid</i>	<i>Redirect ke dashboard sesuai role pengguna</i>	<i>Redirect ke dashboard</i>	Berhasil
2	<i>Login</i>	Pengguna memasukkan <i>email</i> dan <i>password</i> yang salah	<i>Email & Password salah</i>	<i>Kembali ke halaman login</i>	<i>Kembali ke halaman login, namun website menampilkan error yang salah</i>	Gagal

3	<i>Reset Password</i>	<i>Pengguna memasukkan email untuk mendapat OTP</i>	<i>Email yang valid</i>	<i>Redirect ke halaman verify OTP</i>	<i>Redirect ke halaman verify OTP</i>	Berhas il
4	<i>Verify OTP</i>	<i>Pengguna mendapat email berupa kode OTP untuk reset password</i>	-	<i>Mendapat email</i>	<i>Mendapat email</i>	Berhas il
5	<i>Verify OTP</i>	<i>Pengguna memasukkan kode OTP yang valid</i>	<i>Kode OTP reset password yang valid</i>	<i>Ke halaman reset password</i>	<i>Ke halaman reset password</i>	Berhas il
6	<i>Verify OTP</i>	<i>Pengguna memasukkan kode OTP yang salah</i>	<i>Kode OTP reset password yang salah</i>	<i>Kembali ke halaman verify OTP</i>	<i>Kembali ke halaman verify OTP</i>	Berhas il
7	<i>Reset Password</i>	<i>User memasukkan password baru yang cocok</i>	<i>Password baru yang cocok</i>	<i>Redirect ke halaman dashboard</i>	<i>Redirect ke halaman dashboard</i>	Berhas il
8	<i>Reset Password</i>	<i>User memasukkan password baru yang tidak cocok</i>	<i>Password baru yang tidak cocok</i>	<i>Kembali ke halaman reset password</i>	<i>Kembali ke halaman reset password</i>	berhas il

Berdasarkan tabel 4.2, pengujian *blackbox* menunjukkan tingkat keberhasilan 87.5% (7 dari 8 *test case*). Mayoritas fungsi autentikasi berjalan dengan baik, namun ditemukan satu *issue* pada pesan *error* yang tidak sesuai ketika pengguna memasukkan kredensial yang salah. Perbaikan *minor* diperlukan untuk meningkatkan *user experience*.

Tabel 4. 3 Pengujian Manajemen Data *Master*

No	Fitur yang Diuji	Deskripsi Pengujian	<i>Input</i>	<i>Expected output</i>	<i>Actual output</i>	Status
1	Menambah mahasiswa baru	Menambah data <i>user</i> dengan <i>role</i> mahasiswa dengan data yang valid	Data <i>user</i> mahasiswa yang valid	Data berhasil ditambahkan	Data berhasil ditambahkan	Berhasil
2	Menambah mahasiswa baru	Menambah data <i>user</i> dengan <i>role</i> mahasiswa dengan data yang tidak valid	Data <i>user</i> mahasiswa yang tidak valid	Form menolak data yang tidak valid	Form menolak data yang tidak valid	Berhasil
3	Menambah admin baru	Menambah data <i>user</i> dengan <i>role</i> admin yang valid	Data <i>user</i> admin yang valid	Data berhasil ditambahkan	Data berhasil ditambahkan	Berhasil
4	Menambah admin baru	Menambah data <i>user</i> dengan <i>role</i> admin yang tidak valid	Data <i>user</i> admin yang tidak valid	Form menolak data yang tidak valid	Form menolak data yang tidak valid	Berhasil
5	Menambah	Menambah data <i>user</i> dengan <i>role</i> dosen yang valid	Data <i>user</i> dosen	Data berhasil	Data berhasil	Berhasil

	dosen baru		yang valid	ditambahkan	ditambahkan	
6	Menambah dosen baru	Menambah data <i>user</i> dengan <i>role</i> dosen yang tidak valid	Data <i>user</i> dosen yang tidak valid	<i>Form</i> menolak data yang tidak valid	<i>Form</i> menolak data yang tidak valid	Berhasil
7	Mengubah data <i>user</i>	Merubah data <i>user</i> dengan data yang valid	Data <i>user</i> yang valid	Data berhasil dirubah	Data berhasil dirubah	Berhasil
8	Mengubah data <i>user</i>	Merubah data <i>user</i> dengan data yang tidak valid	Data <i>user</i> yang tidak valid	Form menolak data yang tidak valid	Form menolak data yang tidak valid	Berhasil
9	Menghapus data <i>user</i>	Menghapus data <i>user</i>	Data <i>user</i> yang ada	Data berhasil dihapus	Data berhasil dihapus	Berhasil
10	Menambah data prodi	Menambah data prodi dengan data yang valid	Data prodi yang valid	Data berhasil ditambahkan	Data berhasil ditambahkan	Berhasil
10	Menambah data prodi	Menambah data prodi dengan data yang tidak valid	Data prodi yang tidak valid	<i>Form</i> menolak data yang tidak valid	<i>Form</i> tidak menolak data yang tidak valid, pesan <i>error</i> tidak sesuai	Gagal

11	Mengubah data prodi	Mengubah data prodi dengan data yang valid	Data prodi yang valid	Data berhasil dirubah	Data berhasil dirubah	Berhasil
12	Mengubah data prodi	Mengubah data prodi dengan data yang tidak valid	Data prodi yang tidak valid	<i>Form</i> menolak data yang tidak valid	<i>Form</i> menolak data yang tidak valid	Berhasil
13	Menghapus data prodi	Menghapus data prodi yang ada	Data prodi yang ada	Data berhasil dihapus	Data berhasil dihapus	Berhasil
14	Mencari data <i>user</i>	Mencari data <i>user</i> dengan data yang valid	Data <i>user</i> yang valid	Data berhasil diambil	Data berhasil diambil	Berhasil
15	Mencari data <i>user</i>	Mencari data <i>user</i> dengan data yang tidak valid	Data <i>user</i> yang tidak valid	Halaman melempar <i>error</i> "tidak ada data <i>user</i> "	Halaman mendapat <i>error render</i>	Gagal
16	Mencari data prodi	Mencari data prodi dengan data yang valid	Data prodi yang valid	Data berhasil diambil	Data berhasil diambil	Berhasil
17	Mencari data prodi	Mencari data prodi dengan data yang tidak valid	Data prodi yang tidak valid	Halaman melempar <i>error</i> "tidak ada data prodi"	Halaman melempar <i>error</i> "tidak ada data prodi"	Berhasil

18	Menambah data mata kuliah	Menambah data mata kuliah dengan data yang valid	Data mata kuliah yang valid	Data berhasil ditambahkan	Data berhasil ditambahkan	Berhasil
19	Menambah data mata kuliah	Menambah data mata kuliah dengan data yang tidak valid	Data mata kuliah yang tidak valid	Form menolak data yang tidak valid	Form tidak menolak data yang tidak valid dan data ditambahkan	Gagal
20	Mengubah data mata kuliah	Mengubah data mata kuliah dengan data yang valid	Data mata kuliah yang valid	Data berhasil dirubah	Data berhasil dirubah	Berhasil
21	Mengubah data mata kuliah	Mengubah data mata kuliah dengan data yang tidak valid	Data mata kuliah yang tidak valid	Form menolak data yang tidak valid	Form tidak menolak data yang tidak valid dan data dirubah	Gagal
22	Menghapus data mata kuliah	Menghapus data mata kuliah yang ada	Data mata kuliah yang ada	Data berhasil dihapus	Data berhasil dihapus	Berhasil

23	Mencari data mata kuliah	Mencari data mata kuliah yang ada	Data mata kuliah yang ada	Data berhasil diambil	Data berhasil diambil	Berhasil
24	Mencari data mata kuliah	Mencari data mata kuliah yang tidak ada	Data mata kuliah yang tidak ada	Halaman melempar <i>error</i> "tidak ada data mata kuliah"	Halaman melempar <i>error</i> "tidak ada data mata kuliah"	Berhasil

Berdasarkan tabel 4.3, sistem menunjukkan tingkat keberhasilan 84% (20 dari 24 *test case*). Fungsi CRUD dasar untuk *user*, program studi, dan mata kuliah berjalan baik, namun terdapat *bug* kritis pada validasi data form penambahan dan pengubahan yang memungkinkan data tersimpan. *Error handling* pada pencarian data *user* juga perlu diperbaiki. Perbaikan mendesak diperlukan sebelum produksi.

Tabel 4. 4 Pengujian Manajemen RPS

No	Fitur yang Diuji	Deskripsi Pengujian	<i>Input</i>	<i>Expected output</i>	<i>Actual output</i>	Status
1	Menambah data RPS	Menambah RPS baru dengan data yang valid	Data RPS yang valid	Data berhasil ditambahkan	Data berhasil ditambahkan	Berhasil
2	Menambah data RPS	Menambah RPS baru dengan data yang tidak valid	Data RPS yang tidak valid	Form menolak data yang tidak valid	Form menolak data yang tidak valid	Berhasil

3	Mengubah data RPS	Merubah data RPS dengan data yang valid	Data RPS yang valid	Data berhasil dirubah	Data berhasil dirubah	Berhasil
4	Mengubah data RPS	Merubah data RPS dengan data yang tidak valid	Data RPS yang tidak valid	Form menolak data yang tidak valid	Form menolak data yang tidak valid	Berhasil
5	Menghapus data RPS	Menghapus data RPS yang ada	Data RPS yang ada	Data berhasil dihapus	Data berhasil dihapus	Berhasil
6	Mencari data RPS	Mencari data RPS yang ada	Data RPS yang ada	Data berhasil diambil	Data berhasil diambil	Berhasil
7	Mencari data RPS	Mencari data RPS yang tidak ada	Data RPS yang tidak ada	Halaman melempar <i>error</i> "tidak ada data RPS"	Halaman melempar <i>error</i> "tidak ada data RPS"	Berhasil
8	Mengekspor data RPS	Mengekspor data RPS yang ada dalam bentuk <i>PDF</i>	Data RPS yang ada	<i>Browser</i> berhasil mengunduh data RPS terkait dalam format <i>PDF</i> yang sudah	<i>Browser</i> berhasil mengunduh data RPS terkait dalam format <i>PDF</i> yang sudah	berhasil

				ditentukan	ditentukan	
--	--	--	--	------------	------------	--

Tabel 4.4 menunjukkan tingkat keberhasilan 100% (8 dari 8 *test case*). Semua fungsi CRUD, validasi data, dan *error handling* berjalan sempurna. Fitur ekspor ke *PDF* juga berfungsi dengan baik. Modul ini siap untuk lingkungan produksi tanpa perbaikan tambahan.

Tabel 4. 5 Pengujian Manajemen Kelas dan Sesi

No	Fitur yang Diuji	Deskripsi Pengujian	<i>Input</i>	<i>Expected output</i>	<i>Actual output</i>	Status
1	Menambah data kelas	Menambah data kelas dengan data yang valid	Data kelas yang valid	Data berhasil ditambahkan, sesi berhasil dibuat	Data berhasil ditambahkan, sesi berhasil dibuat	Berhasil
2	Menambah data kelas	Menambah data kelas dengan data yang tidak valid	Data kelas yang tidak valid	Form menolak data kelas yang tidak valid	Form menolak data kelas yang tidak valid	Berhasil
3	Mengubah data kelas	Mengubah kelas dengan data yang valid	Data kelas yang valid	Data kelas berhasil dirubah, data sesi ikut dirubah	Data kelas berhasil dirubah, data sesi ikut dirubah	Berhasil

4	Mengubah data kelas	Mengubah data kelas dengan data yang tidak valid	Data kelas yang tidak valid	Form menolak data kelas yang tidak valid	Form menolak data kelas yang tidak valid	Berhasil
5	Menghapus data kelas	Menghapus data kelas yang ada	Data kelas yang ada	Data kelas berhasil dihapus, data sesi terkait berhasil dihapus	Data kelas berhasil dihapus, data sesi terkait berhasil dihapus	Berhasil
6	Mencari data kelas	Mencari data kelas dengan data yang ada	Data kelas yang ada	Data berhasil diambil	Data berhasil diambil	Berhasil
7	Mencari data kelas	Mencari data kelas dengan data yang tidak ada	Data kelas yang tidak ada	Halaman melempar <i>error</i> "belum ada kelas, tidak ada data kelas yang tersedia saat ini"	Halaman melempar <i>error</i> "belum ada kelas, tidak ada data kelas yang tersedia saat ini"	Berhasil
8	Merubah data sesi spesifik	Merubah data sesi spesifik dengan data yang valid	Data sesi yang valid	Data sesi dengan id tersebut berhasil dirubah,	Data sesi dengan id tersebut berhasil dirubah,	Berhasil

				tanpa merubah data kelas dan sesi dengan id lain.	tanpa merubah data kelas dan sesi dengan id lain.	
9	Merubah data sesi spesifik	Merubah data sesi spesifik dengan data yang tidak valid	Data sesi yang tidak valid	Form menolak data sesi yang tidak valid	Form menolak data sesi yang tidak valid	Berhasil

Berdasarkan tabel 4.5, pengujian mendapat tingkat keberhasilan 100% (9 dari 9 *test case*). Semua fungsi pengelolaan kelas dan sesi, termasuk cascade *delete* dan operasi spesifik pada sesi individual, berjalan sempurna. Validasi data dan integritas relasi antar data juga solid. Modul siap untuk produksi.

Tabel 4. 6 Pengujian Manajemen Kelas Mahasiswa

No	Fitur yang Diuji	Deskripsi Pengujian	<i>Input</i>	<i>Expected output</i>	<i>Actual output</i>	Status
1	Pendaftaran kelas	Mahasiswa mendaftar ke kelas yang kapasitasnya belum penuh	Data mahasiswa dan kelas terkait	Mahasiswa berhasil mendaftar ke kelas	Mahasiswa berhasil mendaftar ke kelas	Berhasil
2	Pendaftaran kelas	Mahasiswa mendaftar ke kelas dimana dirinya sudah terdaftar	Data mahasiswa dan data kelas yang	Halaman melempar <i>error</i> “anda sudah terdaftar	Halaman melempar <i>error</i> “anda sudah terdaftar	Berhasil

			sudah terdaftar	di kelas ini”	di kelas ini”	
3	Pendaftaran kelas	Mahasiswa mendaftar ke kelas yang kapasitasnya sudah penuh	Data mahasiswa dan data kelas yang sudah penuh	Tombol daftar kelas tidak bisa ditekan, dan status kelas menjadi ”penuh”	Tombol daftar kelas tidak bisa ditekan, dan status kelas menjadi ”penuh”	Berhasil
4	Keluar kelas	Mahasiswa mengeluarkan dirinya dari kelas terkait	Data mahasiswa dan data kelas terkait	Data kelas terhapus dari kelas terdaftar, dan kembali ke halaman daftar kelas	Data kelas terhapus dari kelas terdaftar, dan kembali ke halaman daftar kelas	Berhasil
5	Penambahan mahasiswa ke kelas	Menambahkan mahasiswa ke dalam kelas dengan data yang valid	Data mahasiswa yang valid	Mahasiswa berhasil di daftarkan	Mahasiswa berhasil di daftarkan	Berhasil
6	Penambahan mahasiswa	Menambahkan mahasiswa kedalam kelas dengan data yang tidak valid	Data mahasiswa yang tidak valid	Form menolak data mahasiswa yang	Form menolak data mahasiswa yang	Berhasil

				tidak valid	tidak valid	
7	Penambahan mahasiswa	Menambahkan mahasiswa ke dalam kelas yang sudah penuh	Data mahasiswa yang valid	Data berhasil ditolak, dan melempar <i>error</i> “kelas sudah penuh”	Data berhasil ditolak, namun <i>error</i> masih belum sesuai	Gagal
8	Penghapusan mahasiswa dalam kelas	Menghapus mahasiswa yang terdaftar dalam kelas	Data mahasiswa yang valid	Data berhasil dihapus	Data berhasil dihapus	Berhasil

Tabel 4.6 menunjukkan bahwa pengujian mencapai tingkat keberhasilan 87.5% (7 dari 8 *test case*). Fungsi pendaftaran, validasi kapasitas, dan pencegahan pendaftaran ganda berjalan baik. Hanya terdapat kesalahan kecil pada pesan *error untuk kelas* yang sudah penuh. Perbaikan kecil diperlukan pada *error handling*.

Tabel 4. 7 Pengujian Pencatatan Kehadiran dan *Feedback* Mahasiswa

No	Fitur yang Diuji	Deskripsi Pengujian	<i>Input</i>	<i>Expected output</i>	<i>Actual output</i>	Status
1	Mengambil data kehadiran pada sesi	Mengambil data kehadiran mahasiswa pada satu sesi	Data mahasiswa dan sesi terkait	Berhasil mengambil data kehadiran mahasiswa	Berhasil mengambil data kehadiran mahasiswa	Berhasil

2	Mencat at data kehadiran pada sesi	Mencatat data kehadiran pada satu sesi sesuai dengan jadwal yang ditentukan	Data mahasiswa dan sesi terkait	Berhasil mencatat data kehadiran mahasiswa	Berhasil mencatat data kehadiran mahasiswa	Berhas il
3	Mencat at data kehadiran pada sesi	Mencatat data kehadiran pada sesi tidak sesuai dengan jadwal yang ditentukan	Data mahasiswa dan sesi terkait	gagal mencatat data kehadiran , halaman mengemb alikan <i>error</i> ”pencatat an kehadiran belum dibuka”	gagal mencatat data kehadiran , halaman mengemb alikan <i>error</i> ”pencatat an kehadiran belum dibuka”	Berhas il
4	Menga mbil data <i>feedback</i>	Mengambil data <i>feedback</i> mahasiswa pada satu sesi	Data mahasiswa dan sesi terkait	Berhasil mengamb il data <i>feedback</i> mahasiswa pada sesi tersebut	Berhasil mengamb il data <i>feedback</i> mahasiswa pada sesi tersebut	Berhas il
5	Mencat at data <i>feedback</i>	Mengirimkan data <i>feedback</i> mahasiswa sesuai pada jadwal sesi	Data <i>feedback</i> , mahasiswa dan	Berhasil menyimp an data <i>feedback</i>	Berhasil menyimp an data <i>feedback</i>	Berhas il

			sesi terkait	pada sesi tersebut	pada sesi tersebut	
6	Mencat at data <i>feedback</i> k	Mengirimkan data <i>feedback</i> mahasiswa tidak sesuai pada jadwal sesi	Data <i>feedback</i> , mahasis wa dan sesi terkait	Gagal mengirim data <i>feedback</i> , halaman mengemb alikan <i>error</i> ”pengirim an <i>feedback</i> belum dibuka”	Gagal mengirim data <i>feedback</i> , namun halaman mengemb alikan <i>error</i> ” <i>pencatat</i> an kehadiran belum dibuka”	Gagal
7	Mencat at data <i>feedback</i> k	Mengirimkan data <i>feedback</i> mahasiswa sesuai dengan jadwal sesi namun data yang dikirim tidak lengkap	Data mahasis wa, sesi terkait dan data <i>feedback</i> yang tidak lengkap	Form menolak pengirima n data	Form menolak pengirima n data	Berhas il

Tabel 4.7 menunjukkan pengujian menghasilkan tingkat keberhasilan 85.7% (6 dari 7 *test case*). Sistem berhasil mengelola kehadiran dan *feedback* dengan validasi jadwal yang baik. Terdapat satu *bug* pada pesan *error feedback* di luar jadwal yang menampilkan pesan kehadiran. Perbaikan diperlukan untuk akurasi pesan *error*.

Tabel 4. 8 Pengujian Rekapitulasi

No	Fitur yang Diuji	Deskripsi Pengujian	<i>Input</i>	<i>Expected output</i>	<i>Actual output</i>	Status
1	Pengambilan data rekap	Mengambil data rekap berdasarkan kelas	Data kelas	Berhasil menampilkan data rekap kelas dengan data dari tanggal dimulainya kelas sampai data diambil	Berhasil menampilkan data rekap kelas dengan data dari tanggal dimulainya kelas sampai data diambil	Berhasil
2	Pengambilan data <i>snapshot</i> rekap	Mengambil seluruh data <i>snapshot</i> rekap berdasarkan kelas	Data kelas	Berhasil menampilkan data <i>snapshot</i> rekap kelas dengan data dari tanggal dimulainya kelas sampai data disimpan	Berhasil menampilkan data <i>snapshot</i> rekap kelas dengan data dari tanggal dimulainya kelas sampai data disimpan	Berhasil

3	Penambahan <i>snapshot</i> rekap	Menambah data <i>snapshot</i> rekap berdasarkan kelas	Data kelas	Berhasil menyimpan data <i>snapshot</i> rekap kelas dengan data dari tanggal dimulainya kelas sampai data disimpan	Berhasil menyimpan data <i>snapshot</i> rekap kelas dengan data dari tanggal dimulainya kelas sampai data disimpan	Berhasil
4	Penghapusan <i>snapshot</i> rekap	Menghapus data <i>snapshot</i> rekap berdasarkan kelas	Data kelas dan data <i>snapshot</i> rekap	Berhasil menghapus data <i>snapshot</i> rekap	Berhasil menghapus data <i>snapshot</i> rekap	Berhasil
5	Ekspor <i>PDF</i> rekap	Mengekspor data rekap kelas dalam bentuk <i>PDF</i>	Data kelas dan data rekap <i>snapshot</i>	Berhasil mengekspor dan mengunduh data rekap kelas dalam bentuk <i>PDF</i>	Berhasil mengekspor dan mengunduh data rekap kelas dalam bentuk <i>PDF</i>	Berhasil

Berdasarkan tabel 4.8, pengujian menghasilkan tingkat keberhasilan 100% (5 dari 5 *test case*). Semua fungsi rekap *real-time*, *snapshot management*, dan ekspor *PDF* berjalan sempurna. Modul memberikan solusi lengkap untuk pelaporan dan siap untuk produksi.

Tabel 4. 9 Pengujian *Dashboard*

No	Fitur yang Diuji	Deskripsi Pengujian	<i>Input</i>	<i>Expected output</i>	<i>Actual output</i>	Status
1	<i>Dashbo ard</i>	Mengambil data <i>dashboard</i> untuk <i>user role Admin</i>	Data admin	Berhasil menampilkan <i>dashboar d khusus admin</i>	Berhasil menampilkan <i>dashboar d khusus admin</i>	Berhas il
2	<i>Dashbo ard</i>	Mengambil data <i>dashboard</i> untuk <i>user role Mahasiswa</i>	Data mahasiswa	Berhasil menampilkan <i>dashboar d khusus mahasisw a</i>	Berhasil menampilkan <i>dashboar d khusus mahasisw a</i>	Berhas il
3	<i>Dashbo ard</i>	Mengambil data <i>dashboard</i> untuk <i>user role Dosen</i>	Data dosen	Berhasil menampilkan <i>dashboar d khusus dosen</i>	Berhasil menampilkan <i>dashboar d khusus dosen</i>	Berhas il

Tabel 4.9 menunjukkan bahwa pengujian mencapai tingkat keberhasilan 100% (3 dari 3 *test case*). *Role-based access control* berjalan sempurna dengan tampilan *dashboard* yang sesuai untuk setiap jenis pengguna. Modul siap untuk produksi tanpa perbaikan.

Berdasarkan hasil pengujian *Blackbox* terhadap delapan modul sistem dengan total 72 skenario pengujian, sistem menunjukkan kinerja yang baik dengan tingkat keberhasilan keseluruhan sebesar 90.3 persen, yaitu 65 dari 72 *test case* berhasil. Perhitungan ini didapat menggunakan rumus 3.3 yang telah dijabarkan pada bagian 3.6.1 sebelumnya. Empat modul, yaitu manajemen RPS, manajemen kelas dan sesi, rekapitulasi, dan *dashboard*, mencapai tingkat keberhasilan 100 persen dan dinilai telah siap digunakan dalam lingkungan produksi. Tiga modul lainnya, yaitu Fitur *login*, manajemen kelas mahasiswa, dan pencatatan kehadiran, menunjukkan performa sangat baik dengan tingkat keberhasilan di atas 85 persen. Namun demikian, masih ditemukan sejumlah kesalahan *minor* dalam hal penanganan kesalahan dan kejelasan pesan validasi yang perlu disempurnakan.

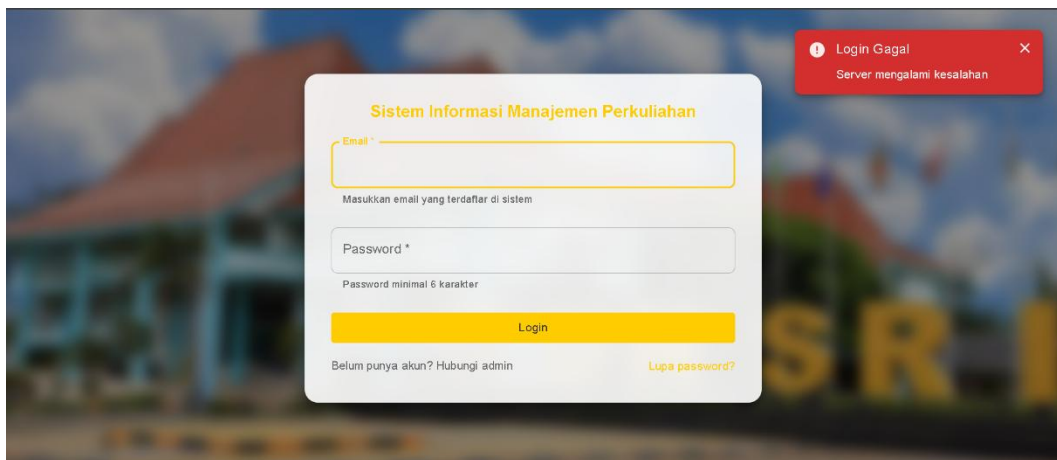
Modul manajemen data *master* menjadi perhatian khusus karena hanya mencapai tingkat keberhasilan sebesar 84 persen. Ditemukan *bug* yang bersifat kritis, terutama dalam proses validasi data yang dapat memengaruhi integritas sistem secara keseluruhan. Oleh sebab itu, perbaikan terhadap mekanisme validasi formulir dan sistem penanganan kesalahan harus segera dilakukan. Seluruh temuan pada tahap pengujian ini akan ditindaklanjuti pada fase evaluasi dan penyempurnaan sistem. Secara umum, fungsionalitas inti sistem telah berjalan baik dan hanya membutuhkan penyempurnaan agar kualitas sistem lebih optimal dari sisi keamanan dan pengalaman pengguna.

4.5 Evaluasi Sistem dan Penyempurnaan

Setelah seluruh proses pengujian *blackbox* selesai, tahap evaluasi sistem dilakukan. Tujuannya adalah untuk memperbaiki fitur yang belum berfungsi dengan baik dan meningkatkan kualitas sistem secara keseluruhan. Fokus evaluasi ini adalah memperbaiki kesalahan yang ditemukan selama pengujian terhadap delapan modul utama. Dari 72 skenario pengujian, tujuh gagal dan memerlukan perawatan tambahan, berikut uraiannya:

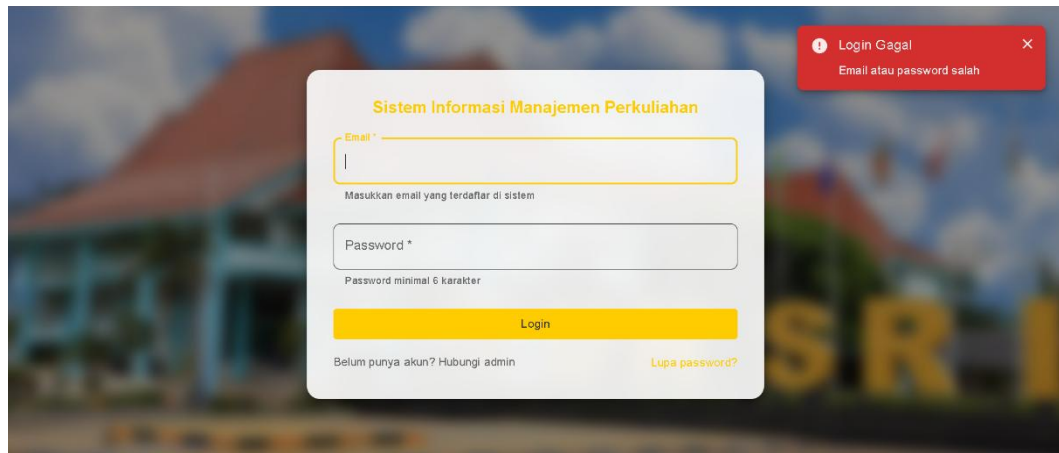
4.5.1 Pengujian Pengguna Memasukkan *Email* dan *Password* yang Salah

Pada pengujian modul pertama, tepatnya pada skenario pengujian pengguna memasukkan *email* dan *password* yang salah atau data yang tidak valid, ditemukan kesalahan dimana pesan *error* yang ditemukan kurang sesuai dengan yang seharusnya. Pesan *error* yang didapat dianggap kurang menjelaskan kesalahan apa yang terjadi pada *website* atau kesalahan apa yang dilakukan oleh pengguna seperti pada gambar 4.45. Oleh karena itu kesalahan ini akan membuat pengguna kebingungan atas apa yang sedang terjadi.



Gambar 4. 45 Evaluasi Kesalahan *Login*

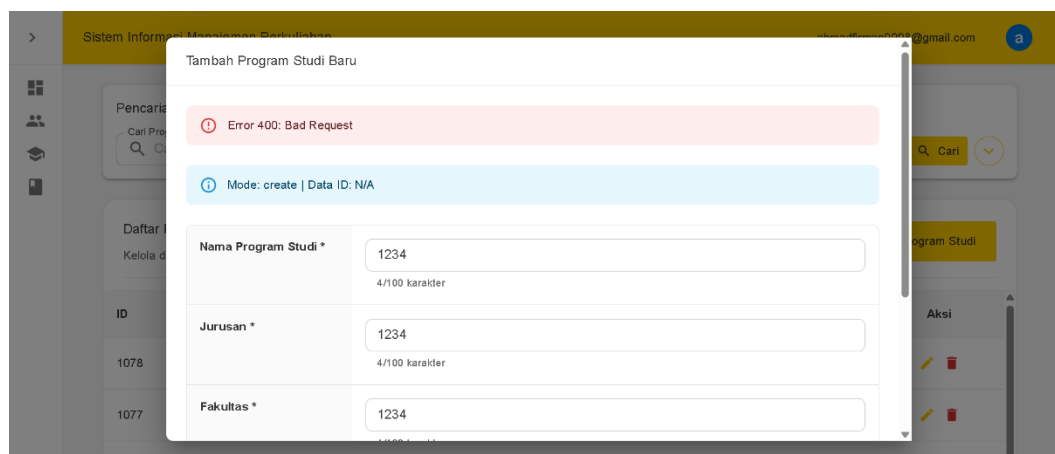
Perbaikan yang dilakukan adalah dengan membuat *error handling* yang lebih sesuai dengan struktur *response* yang seharusnya. Sebelumnya, program memproses logika *error* yang salah dan menampilkan pesan *error default* berupa "Server mengalami kesalahan". Sedangkan seharusnya pesan *error* yang didapat adalah "Email atau password salah" seperti pada gambar 4.46 yang mana pesan *error* ini tentu lebih dapat dipahami dari pada *error* sebelumnya dan membuat *user* lebih memahami kesalahan yang terjadi.



Gambar 4. 46 Hasil dari Perbaikan *Login*

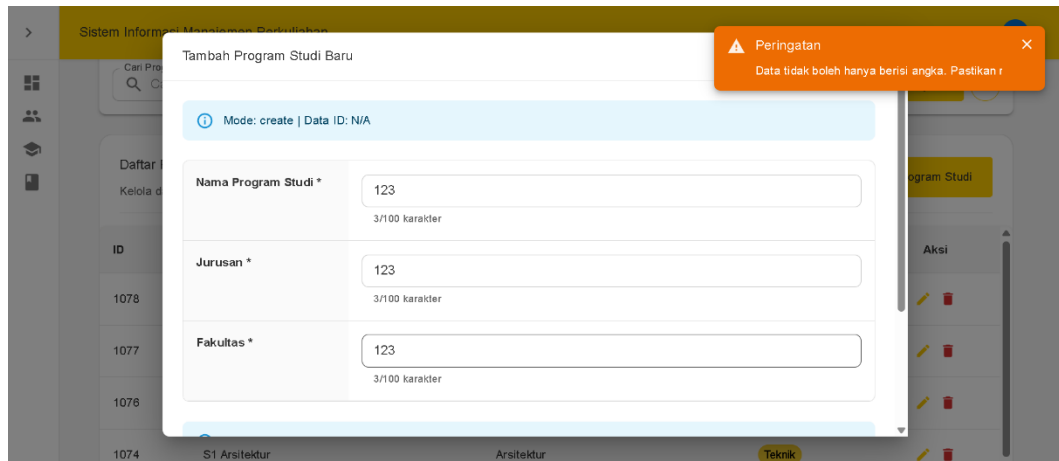
4.5.2 Pengujian Menambahkan Data Prodi

Evaluasi selanjutnya dilakukan pada kegagalan dalam pengujian menambah data prodi. Kesalahan yang terjadi adalah *field form* tidak melakukan validasi data dan menerima data mentah yang dikirim oleh pengguna yang dapat dilihat pada gambar 4.47. Hal tersebut membuat *backend* melempar *error 400 bad request* karena *request* pengguna tidak sesuai dengan apa yang diharapkan. Kasus yang sama juga terjadi seperti pada evaluasi 4.5.1, dimana pesan *error* yang ditampilkan kurang dapat menjelaskan apa kesalahan pengguna sehingga akan menimbulkan kebingungan.



Gambar 4. 47 Kesalahan *Form Prodi*

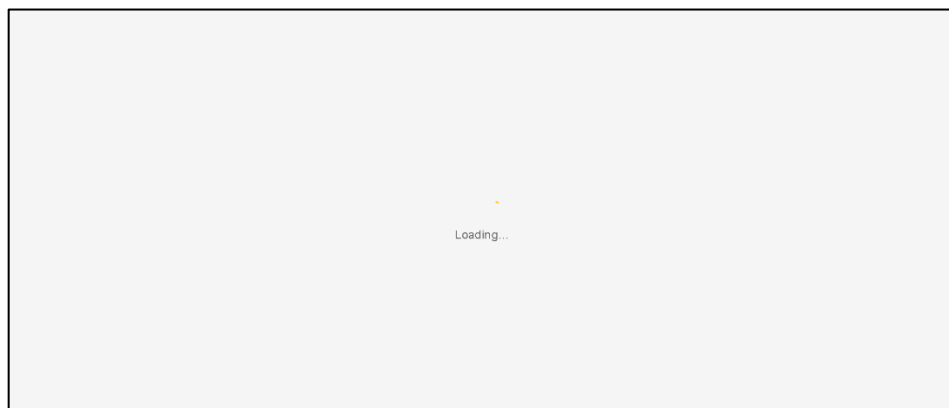
Langkah penyelesaian yang dilakukan adalah dengan menambah validasi pada komponen form prodi sehingga *input* yang dimasukkan oleh *user* akan *di cek* dulu apakah sudah sesuai. Selain penambahan validasi, perbaikan pada *error handling* juga dilakukan. Dengan dilakukannya hal ini, pengguna dapat mengerti kesalahan apa yang terjadi selama pengisian data prodi seperti pada gambar 4.48.



Gambar 4. 48 Hasil Evaluasi *Form Prodi*

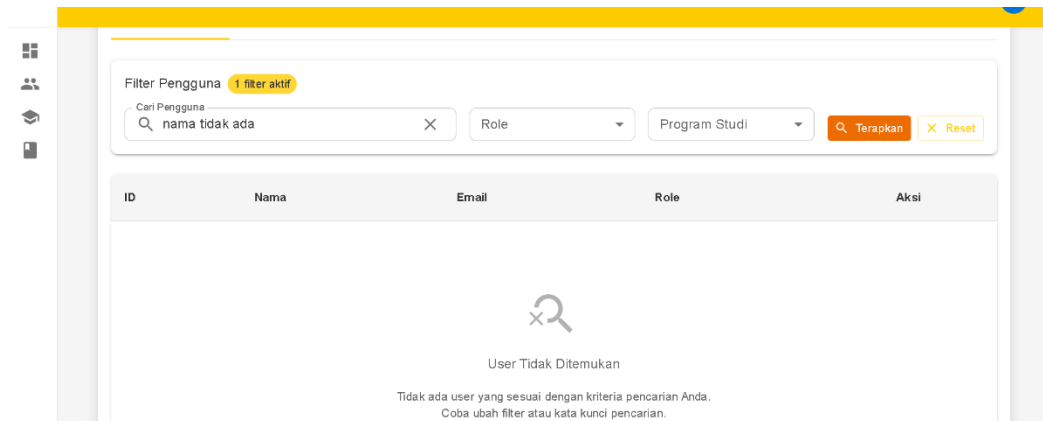
4.5.3 Pengujian Mencari Data *User*

Masih pada modul yang sama seperti evaluasi 4.5.2, kesalahan ditemukan pada pengujian mencari data *user*. *Disini* dapat dilihat pada gambar 4.49, ketika pengujian dilakukan dengan data *user* yang *tidak* ada dalam *database website* terjebak dalam *loading* halaman yang tanpa henti. Hal ini dikarenakan *React* selalu *me-render* ulang komponen yang tidak memiliki *depedencies* yang sesuai. Hal ini menyebabkan komponen *me-render* tanpa henti dan menyebabkan *website crash* karena terbebani.



Gambar 4. 49 Kesalahan Logika Antarmuka

Kesalahan ini dapat diselesaikan dengan menambahkan referensi dari komponen ke *dependencies* yang sesuai. Namun jika hanya melakukan hal itu, komponen hanya akan menampilkan tabel kosong tanpa adanya pesan yang informatif. Sehingga ditambahkanlah pesan "user tidak ditemukan" pada komponen ketika data yang dicari tidak ada dalam *database* seperti yang ada pada gambar 4.50.



Gambar 4. 50 Hasil Evaluasi Pencarian *User*

4.5.4 Pengujian Menambah Data Mata Kuliah

Pada pengujian ini terjadi kesalahan yang sama pada pengujian menambah data prodi seperti yang sudah di evaluasi pada 4.5.2 yaitu kurangnya validasi pada komponen *form*. Ketika pengujian dilakukan komponen *form* tidak menolak data yang seharusnya kurang sesuai dengan apa yang diinginkan *website* seperti pada gambar 4.51. Bedanya pada kesalahan ini, komponen menyimpan data yang salah ke dalam *database* tanpa memeriksanya sama sekali. Hal ini akan menyebabkan banyaknya data sampah yang tersimpan dalam *database*.

Gambar 4. 51 Kesalahan Penambahan Mata Kuliah

Solusi dari permasalahan ini adalah dengan menambah validasi pada komponen *form*. Dengan menambah logika validasi, *form* akan menolak data yang salah atau tidak sesuai bahkan sebelum pengiriman terjadi. Dapat dilihat pada gambar 4.52, nilai yang dimasukkan pada form tidak lagi dapat disimpan karena komponen form akan melakukan validasi untuk setiap *input* yang dimasukan oleh *user*. Hal ini akan membuat *database* terlindungi dari pengguna yang memasukkan data yang salah.

Gambar 4. 52 Hasil Evaluasi Penambahan Mata Kuliah

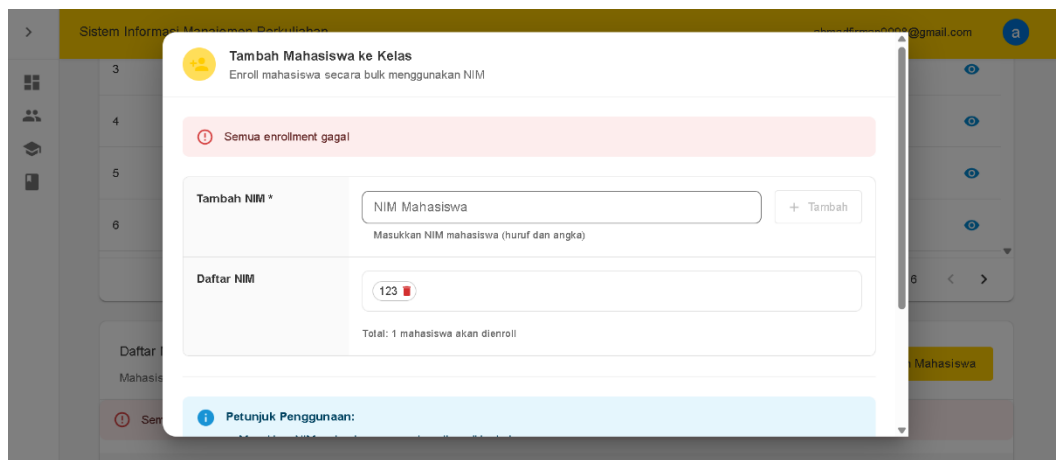
4.5.5 Pengujian Mengubah Data Mata Kuliah

Kesalahan yang terjadi pada pengujian ini terkait dengan kesalahan yang ada pada pengujian menambah data mata kuliah. Hal ini disebabkan kedua fitur ini menggunakan satu komponen yang sama yaitu *MataKuliahForm.jsx*. Yang membedakan fitur mengubah data dan menambah data mata kuliah hanya pada logika nya saja. Oleh karena itu, ketika komponen *MataKuliahForm.jsx* diperbaiki

pada kesalahan menambah data mata kuliah, kesalahan pada fitur mengubah data mata kuliah juga teratasi.

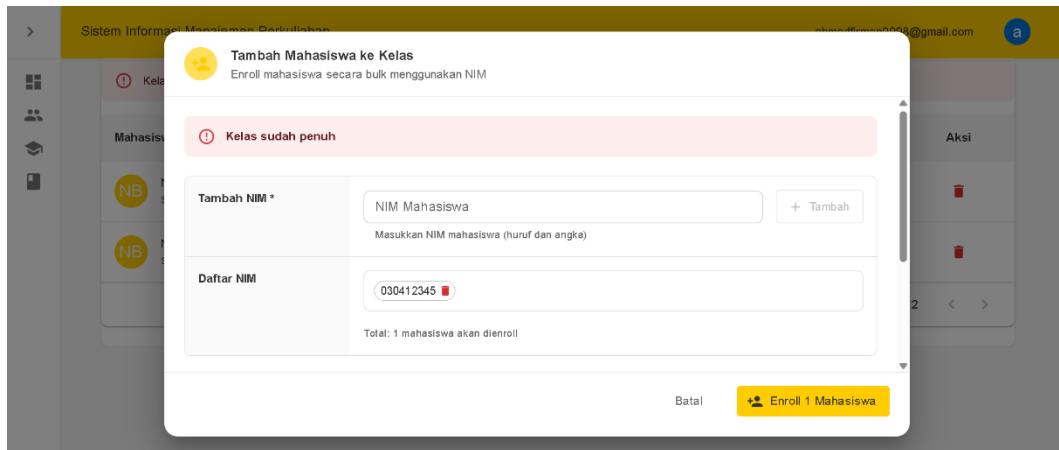
4.5.6 Pengujian Menambahkan Mahasiswa ke Kelas yang Sudah Penuh

Selanjutnya kesalahan terjadi pada modul pendaftaran mahasiswa atau *enrollment*. Fitur yang mengalami kesalahan adalah penambahan mahasiswa ke kelas oleh admin atau dosen. Bentuk kesalahan yang terjadi adalah pesan *error* yang didapat tidak sesuai dengan kesalahannya, sehingga dapat menyebabkan kebingungan pada *user* seperti pada gambar 4.53.



Gambar 4. 53 Kesalahan yang Terjadi pada *Enrollment*

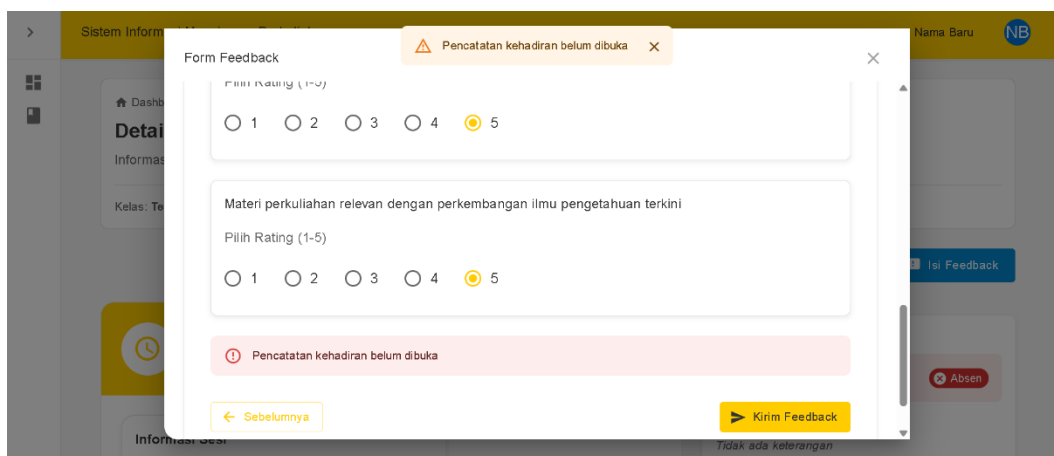
Penyelesaian dari kesalahan ini adalah dengan memperbaiki *error handling* yang ada pada logika *enrollment*. Tabel kelas memiliki *field* kapasitas, yang mana seharusnya jika kapasitasnya sudah sama dengan total mahasiswa yang terdaftar dalam kelas tersebut, maka *backend* harus menolak pendaftaran mahasiswa. Logika yang ada sudah benar, namun pesan kesalahannya tidak memberikan kejelasan tentang apa yang terjadi. Dengan memperbaiki *error handling*, komponen akan menampilkan pesan *error* yang lebih sesuai dan informatif seperti pada gambar 4.54.



Gambar 4. 54 Hasil Evaluasi Pendaftaran Mahasiswa

4.5.7 Pengujian Pencatatan *Feedback*

Kesalahan pengujian terakhir adalah pencatatan *feedback* ketika jadwal belum dibuka. Logika yang ditetapkan pada fitur pencatatan *feedback* adalah *feedback* akan dibuka ketika sesi dalam status "selesai". Ketika pengujian dapat dilihat pada gambar 4.55, ditemukan bahwa pesan *error* yang didapat tidak sesuai, dan malah menampilkan kesalahan yang lain berupa "pencatatan kehadiran belum dibuka".



Gambar 4. 55 Kesalahan Pencatatan *Feedback*

Hal ini disebabkan karena logika *error handling*nya salah mengartikan kesalahan. Disini pesan kesalahan diambil dari kode status *error* "403 forbidden" dan langsung memberikan pesan yang sama ketika *error* nya memiliki kode *error* tersebut walaupun konteksnya berbeda. Kesalahan ini dapat diselesaikan dengan mengambil pesan *error* yang sesuai, bukan hanya dari kode status melainkan dari *response* API terkait *request* fitur yang dilakukan seperti yang ada pada gambar 4.56.

Gambar 4. 56 Evaluasi Pencatatan *Feedback*

4.6 Hasil Evaluasi *Technology Acceptance Model*

Seperti yang telah dijelaskan sebelumnya pada bagian 3.6.4 bahwa untuk mengevaluasi tanggapan *user* terhadap *web* yang dihasilkan, evaluasi *Technology Acceptance Model* dilakukan. Responden diminta memberikan penilaian terhadap 10 pertanyaan menggunakan skala Likert 1 sampai 5. Total terdapat 8 responden yang mengisi kuesioner. Data dianalisis dengan menghitung rata-rata dari setiap konstruk, berdasarkan skor pernyataan yang relevan. Berikut adalah hasil perhitungan rata-rata untuk masing-masing konstruk:

Tabel 4. 10 Hasil Evaluasi *Technology Acceptance Model*

Konstruk	Pernyataan (No)	Rata-rata
<i>Perceived Usefulness</i>	1, 2, 3, 4	4.88
<i>Perceived Ease of Use</i>	5, 6, 7, 8	4.31
<i>Behavioral Intention</i>	9, 10	4.88

Jika kita merujuk pada tabel 4.10, nilai rata-rata tertinggi didapat pada konstruk *Perceived Usefulness* dan *Behavioral Intention* dengan masing-masing nilai 4.88 dari 5, yang menunjukkan bahwa mayoritas responden menilai sistem ini sangat bermanfaat untuk pengelolaan pembelajaran dan menunjukkan minat kuat untuk menggunakannya apabila tersedia. Sementara itu, nilai *Perceived Ease of Use* sedikit lebih rendah yaitu dengan nilai 4.31 dari 5. Hal tersebut menunjukkan bahwa Sebagian responden masih melihat adanya potensi tantangan kecil dalam antarmuka atau kemudahan penggunaan.

Secara keseluruhan, dapat disimpulkan bahwa hasil evaluasi TAM menunjukkan rancangan sistem dapat diterima dengan sangat baik oleh responden. Hal ini dapat menjadi indikasi bahwa sistem memiliki potensi besar untuk diadopsi dan digunakan dalam lingkungan akademik, terutama jika perbaikan kecil pada aspek antarmuka pengguna dapat dibuat lebih intuitif dan mudah diakses oleh semua kalangan.

4.7 Pembahasan

Sistem informasi akademik yang dirancang dalam penelitian ini bertujuan utama untuk mendukung digitalisasi dan implementasi Rencana Pembelajaran Semester (RPS) di Fakultas Teknik Universitas Sriwijaya. Berdasarkan hasil pengembangan dan pengujian, sistem telah berhasil menyediakan fitur inti seperti manajemen RPS, pengelolaan kelas dan sesi, pencatatan kehadiran mahasiswa, serta rekapitulasi pembelajaran dan umpan balik. Hasil pengujian fungsional menunjukkan bahwa mayoritas modul telah berjalan sesuai dengan kebutuhan awal yang dirumuskan, sehingga sistem dapat dikatakan telah memenuhi tujuan penelitian yang telah ditetapkan di awal.

Dari pengujian *blackbox* terhadap 8 modul utama, sistem menunjukkan tingkat keberhasilan sebesar 90.3%. Modul-modul seperti RPS, kelas, sesi, rekapitulasi, dan *dashboard* bahkan menunjukkan hasil sempurna tanpa kegagalan fungsi. *Bug* dan *error minor* yang ditemukan pada tahap pengujian telah diperbaiki pada tahap evaluasi dan penyempurnaan. Selain itu, hasil kuesioner *Technology Acceptance Model* (TAM) menunjukkan bahwa sistem mendapat penerimaan yang sangat baik dari pengguna awal. *Perceived Usefulness* dan *Behavioral Intention* sama-sama mencatat nilai rata-rata sebesar 4,88, menandakan bahwa pengguna merasa sistem ini sangat bermanfaat dan ingin menggunakannya. *Perceived Ease of Use* memperoleh skor 4,31, yang menunjukkan kemudahan penggunaan sudah cukup baik, meskipun masih dapat disempurnakan lagi di aspek antarmuka.

Jika dibandingkan dengan sistem *E-Learning* Universitas Sriwijaya yang telah lebih dulu digunakan, sistem yang dikembangkan dalam penelitian ini menawarkan pendekatan yang lebih terintegrasi. *E-Learning* Unsri berfungsi sebagai media unggah materi dan tugas, tanpa pengelolaan pembelajaran mingguan yang terhubung dengan RPS. Sebaliknya, sistem baru ini menjadikan RPS sebagai

pusat aktivitas pembelajaran, mulai dari perencanaan, pelaksanaan, hingga evaluasi, yang secara struktural mendukung proses akademik yang lebih sistematis.

Beberapa kelebihan sistem ini antara lain integrasi fitur yang mendukung pembelajarannya sudah berbasis RPS, alur penggunaan yang relatif sederhana, serta dapat menghasilkan dokumentasi yang baik melalui fitur pelaporan. Sistem juga berhasil memetakan hubungan antara dosen, mahasiswa, dan kegiatan belajar secara *real time*. Namun, sistem masih memiliki keterbatasan, terutama pada aspek pengelolaan materi ajar dan penugasan. Fitur untuk mengunggah materi perkuliahan dan mengelola penilaian tugas belum tersedia, sehingga dosen masih harus mengandalkan *platform* lain. Di samping itu, sistem juga belum terintegrasi dengan sistem akademik kampus secara menyeluruh.

Dengan tingkat keberhasilan dan penerimaan yang tinggi, sistem ini memiliki potensi untuk diterapkan secara lebih luas di lingkungan fakultas. Sistem ini dapat menjadi dasar awal pengembangan sistem akademik yang lebih lengkap, mencakup fitur-fitur seperti pengelolaan tugas, penilaian, manajemen materi ajar, serta integrasi dengan sistem akademik unsri yang sudah ada. Hasil dari penelitian ini juga dapat menjadi rujukan untuk penelitian lanjutan dalam pengembangan sistem akademik digital berbasis *outcome learning* dan data analitik pendidikan.