

RANCANG BANGUN *BACK-END* SISTEM INFORMASI CAPAIAN PEMBELAJARAN LULUSAN (CPL) MENGGUNAKAN LARAVEL

Faniel Sianipar^{1*}, Wahyu Eko Sulistiono², Yessi Mulyani³, Gigih Forda Nama³

^{1,2,3} Fakultas Teknik Universitas Lampung, Jalan Prof. Dr Jl. Prof. Dr. Ir. Sumantri Brojonegoro No.1, Kota Bandar Lampung, Lampung 35141

Keywords:

Capaian Pembelajaran Lulusan, *Back-end*, Laravel, *Agile*, *Kanban*, CPMK, *Postman*, *PHPUnit*

Correspondent Email:

fanielsianipar01@gmail.com

Abstrak. Pendidikan tinggi memiliki peran strategis dalam mencetak lulusan berkualitas melalui kurikulum berbasis capaian pembelajaran. Namun, di Universitas Lampung perhitungan Capaian Pembelajaran Lulusan (CPL) belum dilakukan secara sistematis sehingga evaluasi kurikulum menjadi kurang optimal. Penelitian ini bertujuan membangun *back-end* Sistem Informasi CPL menggunakan *framework* Laravel dengan pendekatan *Agile Kanban*. Tahapan penelitian meliputi studi literatur, analisis kebutuhan fungsional dan non-fungsional, perancangan sistem dengan arsitektur yang modular, serta implementasi *API* yang mendukung pengelolaan data CPL, CPMK, kelas, mahasiswa, dan akun. Pengujian dilakukan menggunakan *PHPUnit testing* dengan data *dummy* untuk memastikan fungsionalitas utama berjalan sesuai kebutuhan. Hasil penelitian menunjukkan *back-end API* yang modular, terorganisasi, dan siap diintegrasikan dengan *front-end*, meskipun masih diperlukan pengujian dengan data asli dan *refactor* kode untuk peningkatan kualitas. Sistem ini diharapkan dapat mendukung program studi dalam evaluasi kurikulum dan proses akreditasi secara lebih efektif.



Copyright © [JITET](#) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstract. Higher education plays a strategic role in producing qualified graduates through outcome based curricula. However, at the University of Lampung, the calculation of Graduate Learning Outcomes (Capaian Pembelajaran Lulusan/CPL) has not been carried out systematically, making curriculum evaluation less optimal. This study aims to develop the back end of a CPL Information System using the Laravel framework and the Agile Kanban approach. The research stages include literature review, analysis of functional and non functional requirements, system design with a modular architecture, and implementation of APIs to support management of CPL, CPMK, classes, students, and accounts. Testing was performed using PHPUnit with dummy data to ensure core functionalities meet requirements. The results show a modular and well organized back end API ready for front-end integration, although further testing with real data and code refactoring are needed to improve quality. This system is expected to support study programs in curriculum evaluation and accreditation processes more effectively.

1. PENDAHULUAN

Pendidikan tinggi memiliki peran strategis dalam menghasilkan lulusan berkualitas melalui kurikulum yang adaptif, inovatif, dan kolaboratif. Di Universitas Lampung, perhitungan Capaian Pembelajaran Lulusan

(CPL) belum dilakukan secara sistematis sehingga evaluasi kurikulum menjadi kurang optimal, padahal data CPL penting untuk menilai kelebihan dan kelemahan kurikulum dan mendukung pengambilan keputusan berbasis bukti.

Penelitian ini mengembangkan *back-end* Sistem Informasi CPL menggunakan *framework* Laravel dengan pendekatan *Agile Kanban*. Sistem dirancang sebagai *API* modular yang mengelola CPL, CPMK, kelas, mahasiswa, dan akun. Pemilihan Laravel didasarkan pada kemudahan pengembangan (*Eloquent ORM*, *routing*, autentikasi, dan dukungan *RESTful API*) serta kemampuan memisahkan logika bisnis dan data untuk menjamin skalabilitas dan keamanan.

Untuk memastikan keandalan fungsional, implementasi diuji menggunakan *PHPUnit testing* dengan data *dummy* yang mensimulasikan struktur data nyata. Karena data CPL dan CPMK aktual belum tersedia, penggunaan data *dummy* memungkinkan verifikasi fungsionalitas inti sementara integrasi dengan sumber data akademik nyata menjadi langkah lanjutan.

Tujuan penelitian adalah membangun *back-end* yang dapat dikonsumsi *front-end* untuk memfasilitasi perhitungan dan analisis CPL sehingga program studi dapat melakukan evaluasi kurikulum secara berkelanjutan dan mendukung proses akreditasi. Batasan penelitian meliputi fokus pada pengembangan *back-end* (tanpa penyusunan kurikulum) dan belum tersambungannya sistem dengan sistem akademik eksternal.

2. TINJAUAN PUSTAKA

2.1. Outcome-Based Education (OBE)

Pendekatan atau paradigma yang digunakan dalam pengembangan dan pelaksanaan kurikulum umumnya tersusun atas 4 tahapan yang saling berinteraksi, yang selanjutnya dijelaskan secara singkat sebagai berikut:

1. Outcome Based Curriculum (OBC) merupakan pengembangan kurikulum yang berlandaskan profil lulusan dan Capaian Pembelajaran Lulusan (CPL), di mana CPL diturunkan menjadi *body of knowledge*, struktur mata kuliah beserta bobot SKS, peta kurikulum, Rencana Pembelajaran Semester (RPS), bahan ajar, serta instrumen penilaian dan evaluasi untuk memastikan keselarasan antara tujuan pembelajaran dan komponen kurikulum.
2. Outcome Based Learning and Teaching (OBLT) merupakan pelaksanaan kegiatan

pembelajaran yang menekankan interaksi antara dosen, mahasiswa, dan sumber belajar, di mana pemilihan bentuk dan metode pembelajaran harus disesuaikan dengan Capaian Pembelajaran Lulusan (CPL). Bentuk pembelajaran ini mencakup pula kegiatan di luar program studi atau kampus (mis. program Merdeka Belajar Kampus Merdeka) untuk mendukung pencapaian CPL.

3. Outcome Based Assessment and Evaluation (OBAE) merupakan pendekatan penilaian dan evaluasi yang dirancang untuk mengukur ketercapaian CPL baik selama proses pembelajaran maupun pada hasil belajarnya, dengan tujuan meningkatkan mutu pembelajaran secara berkelanjutan, evaluasi kurikulum dan penilaian pembelajaran dilakukan berdasarkan bukti capaian untuk menentukan sejauh mana hasil belajar memenuhi standar yang ditetapkan dan untuk merumuskan langkah perbaikan yang diperlukan.
4. Continuous Improvement (CI) merupakan hasil evaluasi yang didasarkan pada bukti capaian pembelajaran digunakan untuk melakukan perbaikan secara iteratif dan berkelanjutan. Langkah perbaikan ini dapat meliputi revisi kurikulum, penyesuaian metode dan strategi pembelajaran, pembaruan materi ajar, serta penyempurnaan instrumen penilaian guna memastikan peningkatan ketercapaian CPL di masa mendatang [1].

2.2. Capaian Pembelajaran

Capaian Pembelajaran merupakan kemampuan yang diperoleh peserta didik melalui internalisasi pengetahuan, pembentukan sikap, penguasaan keterampilan, pengembangan kompetensi, serta akumulasi pengalaman kerja [2].

2.3. Capaian Pembelajaran Lulusan (CPL)

Capaian Pembelajaran Lulusan (CPL) merupakan kompetensi yang wajib dimiliki oleh lulusan perguruan tinggi sebagai hasil dari proses pembelajaran. CPL dirumuskan untuk menjawab tantangan era Industri 4.0 menuju masyarakat 5.0 serta mengakomodasi keterampilan abad ke-21. Secara holistik, CPL mencakup kemampuan literasi data (pemahaman dalam membaca, menganalisis, dan memanfaatkan *big data*), literasi teknologi

(penguasaan prinsip kerja mesin, *coding*, kecerdasan buatan, dan rekayasa), serta literasi manusia (pemahaman humaniora, komunikasi, dan desain). Selain itu, CPL juga mengintegrasikan keterampilan berpikir tingkat tinggi (*high order thinking skills/HOTS*) seperti komunikasi, kolaborasi, berpikir kritis-kreatif, logika komputasional, empati, dan tanggung jawab kewarganegaraan. Pemahaman terhadap perkembangan era digital, penerapan ilmu untuk kemaslahatan lokal hingga global, isu keberlanjutan (*sustainability*), kewarganegaraan global (*global citizenship*), serta prinsip pendidikan inklusif dan adaptif turut menjadi komponen kunci. CPL juga memperluas cakupan kompetensi melalui program Merdeka Belajar Kampus Merdeka (MBKM), yang memfasilitasi pencapaian pembelajaran di luar program studi [3].

2.4. Sistem Informasi

Sistem Informasi (SI) berfungsi sebagai fondasi pengelolaan dan pemanfaatan informasi secara efektif untuk mendukung pengambilan keputusan, pengendalian, dan operasi organisasi. Penerapannya dapat meningkatkan efisiensi, akurasi, dan daya saing organisasi serta mengikuti siklus hidup yang meliputi perencanaan, pengembangan, implementasi, operasi, dan pemeliharaan [4].

2.5. Back-End

Back-end merupakan komponen kunci dalam arsitektur sistem yang bertanggung jawab untuk memproses logika bisnis, mengelola data, dan menyediakan layanan yang diperlukan oleh *front-end*. *Back-end* juga berperan dalam memastikan keamanan, skalabilitas, dan kinerja sistem secara keseluruhan [5].

2.6. Framework Laravel

Dalam perkembangan awal web dinamis, pengembangan aplikasi web mengharuskan pengembang untuk menulis kode tidak hanya untuk logika bisnis utama, tetapi juga untuk berbagai komponen yang umum digunakan, seperti *authentication* pengguna, validasi input, akses basis data, dan *templating*. Namun, dengan berkembangnya teknologi, kini terdapat berbagai *framework* dan pustaka yang memudahkan proses pengembangan aplikasi. Laravel hadir sebagai salah satu *framework PHP* yang menawarkan solusi komprehensif dengan menyediakan berbagai fitur bawaan yang memungkinkan pengembang untuk

membangun aplikasi dengan lebih efisien. Dalam dunia pemrograman yang terus berkembang, di mana berbagai *framework* baru terus bermunculan, Laravel tetap menjadi pilihan utama karena fleksibilitas, kemudahan penggunaan, serta ekosistem dan komunitasnya yang kuat [6].

2.7. Laravel Sanctum

Laravel *Sanctum* menyediakan sistem *authentication* yang sangat ringan untuk SPAs (*single page applications*), *mobile applications*, dan API sederhana berbasis token. *Sanctum* memungkinkan setiap pengguna aplikasi anda untuk menghasilkan beberapa token *API* untuk akunnya. Token ini dapat diberikan *abilities/scopes* yang menentukan tindakan apa saja yang diizinkan untuk dilakukan oleh token tersebut [7].

2.8. Spatie Laravel-Permission

Spatie Laravel-Permission memungkinkan pengaitan pengguna dengan *roles* dan *permissions*. Paket ini memudahkan penambahan *permissions* atau *roles* ke pengguna dalam aplikasi Laravel anda. Jika anda menggunakan *multiple guards*, paket ini juga mendukungnya. Setiap *guard* memiliki set *permissions* dan *roles* tersendiri yang dapat diberikan kepada pengguna pada *guard* tersebut [8].

2.9. Software Development Life Cycle (SDLC)

SDLC (Software Development Life Cycle) adalah pendekatan sistematis dalam pengembangan perangkat lunak yang dapat membantu tim dalam mengatur tahapan pembuatan, pengembangan, pengujian, dan pemeliharaan perangkat lunak dengan struktur yang baik dan efisien, sehingga tim dapat menghasilkan perangkat lunak berkualitas tinggi yang memenuhi kebutuhan pengguna, tepat waktu, dan sesuai anggaran yang telah ditentukan [9].

2.10. Agile

Agile merupakan sekumpulan metode dan metodologi yang dirancang untuk meningkatkan efektivitas berpikir, efisiensi kerja, serta kualitas pengambilan keputusan dalam tim. Metode ini mencakup seluruh aspek rekayasa perangkat lunak tradisional, seperti manajemen proyek, desain, dan arsitektur perangkat lunak, serta peningkatan proses. Setiap metode dan metodologi dalam *Agile* terdiri atas serangkaian praktik

yang telah disederhanakan dan dioptimalkan untuk memudahkan adopsi dalam pengembangan sistem. Terdapat beberapa metodologi *Agile* yang populer diantaranya adalah *Scrum*, *Extreme Programming (XP)*, *Lean*, dan *Kanban* [10].

2.11. Kanban

Kanban adalah metode peningkatan proses yang digunakan oleh tim yang menggunakan metodologi *Agile* untuk memperbaiki alur kerja pengembangan perangkat lunak secara bertahap. Berbeda dengan pendekatan yang mengharuskan perubahan drastis, Kanban dimulai dengan memahami proses yang sedang berjalan, lalu melakukan penyempurnaan secara berkelanjutan [10].

2.12. Use Case Diagram

Use case adalah metode yang digunakan untuk menggambarkan interaksi antara pengguna dan sistem melalui kombinasi model grafis dan teks terstruktur. Secara dasar, *use case* mengidentifikasi dua komponen utama:

1. Aktor: Pihak yang terlibat dalam interaksi, baik manusia maupun sistem lain, direpresentasikan dengan simbol figur manusia (*stick figure*).
2. Interaksi: Jenis kegiatan yang dilakukan aktor terhadap sistem, direpresentasikan sebagai elips berlabel (*named ellipse*).

Hubungan antara aktor dan interaksi ditunjukkan dengan garis penghubung. Tanda panah (*arrowhead*) dapat ditambahkan pada garis untuk memperjelas arah inisiasi interaksi. Kumpulan *use case* dalam sebuah sistem menggambarkan seluruh interaksi yang akan menjadi dasar perumusan kebutuhan sistem (*system requirements*). Diagram *use case* tingkat tinggi (*high-level use case diagram*) digunakan sebagai alat visualisasi untuk memetakan hubungan ini secara menyeluruh [11].

2.13. Activity Diagram

Activity Diagram merupakan representasi visual dari berbagai tahapan dalam suatu proses atau pemrosesan data. Dalam pemodelan sistem menggunakan *Unified Modeling Language (UML)*, *activity diagram* sering digunakan untuk menggambarkan proses bisnis dan bagaimana sistem beroperasi di dalamnya. Diagram ini menunjukkan alur aktivitas dalam suatu proses serta bagaimana kontrol berpindah dari satu aktivitas ke aktivitas berikutnya.

Setiap proses dalam diagram ini diawali dengan node awal, yang biasanya direpresentasikan dengan lingkaran berisi warna penuh, dan berakhir dengan node akhir, yang digambarkan sebagai lingkaran dengan lingkaran lain di dalamnya.

Setiap aktivitas dalam diagram ini digambarkan dalam bentuk persegi panjang dengan sudut melengkung, yang menunjukkan subproses yang harus dijalankan dalam sistem. *Activity diagram* juga dapat mencakup berbagai objek yang digunakan dalam proses tertentu, seperti data yang diproses atau entitas yang berinteraksi dalam alur kerja. Dengan pemodelan seperti ini, *activity diagram* memberikan pemahaman yang lebih sistematis terkait bagaimana setiap tahapan berlangsung dan bagaimana keputusan atau aksi tertentu memengaruhi aliran kerja suatu sistem. Diagram ini sangat berguna dalam perancangan sistem karena membantu dalam mengidentifikasi langkah-langkah kritis serta memastikan bahwa setiap proses berjalan sesuai dengan yang diharapkan dalam pengembangan perangkat lunak [11].

2.14. PHPUnit Testing

Laravel menyediakan dukungan bawaan untuk pengujian perangkat lunak melalui integrasi dengan beberapa alat dan kerangka kerja. Laravel dilengkapi dengan *PHPUnit* (untuk pengujian unit), *Mockery* (untuk simulasi objek), dan *Faker* (untuk pembuatan data palsu pada seeding dan pengujian). Selain itu, Laravel memiliki alat pengujian aplikasi yang memungkinkan pengembang melakukan simulasi interaksi dengan sistem, seperti mengakses *URI*, mengirim formulir, memeriksa kode status *HTTP*, serta memvalidasi dan menguji struktur respons *JSON*. Kemudahan pengujian dalam Laravel juga tercermin dari penyediaan contoh tes aplikasi yang dapat langsung dijalankan saat proyek baru dibuat. Hal ini menghilangkan kebutuhan konfigurasi lingkungan pengujian yang rumit, sehingga mengurangi hambatan dalam penulisan tes aplikasi [6].

2.15. MySQL

MySQL merupakan sistem manajemen basis data relasional (*Relational Database Management System/RDBMS*) yang memungkinkan pengelolaan beberapa database secara terpusat dalam satu server. Setiap database terdiri dari tabel-tabel yang terstruktur

dalam bentuk baris (menyimpan entitas data) dan kolom (menyimpan atribut spesifik). Sebelum data dapat dimasukkan, struktur database (termasuk tabel dan kolom) harus ditentukan terlebih dahulu. Relasi antar tabel dibangun menggunakan kunci (*key*) yang menghubungkan data di kolom tertentu [12].

2.16. Postman

Mastering Postman sangat ideal bagi pengembang dan insinyur perangkat lunak yang ingin membangun *API* secara *end-to-end* dengan efisien dan efektif. Materi ini cocok untuk pemula yang baru dalam pengembangan *API* maupun pengembang berpengalaman yang ingin menguasai keterampilan mereka dalam pengembangan, pengujian, *debugging*, dan integrasi *API* [13].

2.17. REST Application Programming Interface (API)

Menyoroti perbedaan antara layanan web yang ideal menurut prinsip *REST API* dan layanan nyata yang sering ditemukan di web yang dapat diprogram. Banyak *API* populer bersifat hibrida dan menggunakan pola *REST* untuk pengambilan data pada saat memodifikasi data, sehingga praktik nyata sering menyimpang dari arsitektur Web yang murni. Untuk itu, mengalokasikan bagian desain yang memperlihatkan contoh *services* yang benar-benar *RESTful* dan berorientasi sumber daya, dengan tujuan memberikan pedoman desain yang konsisten dan selaras dengan prinsip *HTTP* dan model sumber daya Web [14].

2.18. Penelitian Terdahulu

Dalam rangka mendukung pelaksanaan penelitian serta menghindari plagiarisme, dilakukan studi literatur terhadap beberapa penelitian terdahulu yang relevan dengan topik evaluasi Capaian Pembelajaran Lulusan (CPL). Penelitian oleh Wijaya et al. (2023) berjudul *Analisa dan Perancangan Aplikasi Evaluasi Capaian Pembelajaran Lulusan*. Mengembangkan sistem menggunakan *framework CodeIgniter* dengan metode *waterfall*, mencakup fitur manajemen data CPL, penghitungan nilai berbasis akumulasi Capaian Pembelajaran Mata Kuliah (CPMK), serta visualisasi hasil evaluasi dalam bentuk grafik [15].

Selanjutnya, penelitian oleh Penelitian Zikry et al. (2024) berjudul *Implementasi Front-End Sistem Penilaian Capaian Pembelajaran*

Lulusan (CPL) dan Capaian Pembelajaran Matakuliah (CPMK) (Studi Kasus Pada Teknik Komputer Undip). mengembangkan sistem penilaian Capaian Pembelajaran Lulusan (CPL) dan Capaian Pembelajaran Mata Kuliah (CPMK) berbasis web untuk Program Studi Teknik Komputer Universitas Diponegoro dengan fokus pada implementasi *front-end*. Sistem tersebut dibangun menggunakan *framework* Laravel dan *Tailwind CSS*, menerapkan metodologi *waterfall* untuk pengembangan bertahap, serta menyajikan fitur visualisasi grafik rekap nilai dan integrasi data melalui impor *file Excel*. Pengujian kualitas antarmuka menggunakan *User Experience Questionnaire* (UEQ) menunjukkan hasil positif pada aspek daya tarik, efisiensi, dan kejelasan, sedangkan pengujian fungsionalitas melalui *black box testing* mengonfirmasi kelayakan operasional sistem [16].

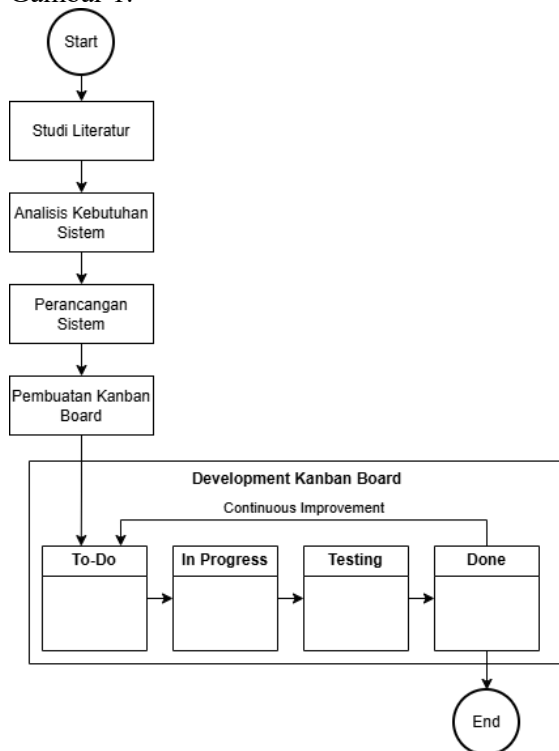
Penelitian oleh Prasimsha dan Putro (2024) berjudul *Manajemen Proyek Pengembangan Perangkat Lunak Kerja Sama Perguruan Tinggi dan Software House (Studi Kasus: Pengembangan Sistem Penilaian Capaian Pembelajaran Lulusan Program Studi Informatika Universitas Islam Indonesia dan PT. Javan Cipta Solusi)*. Mengembangkan sistem penilaian Capaian Pembelajaran Lulusan (CPL) berbasis web untuk Program Studi Informatika Universitas Islam Indonesia melalui kolaborasi dengan PT. Javan Cipta Solusi. Sistem ini dibangun menggunakan *framework* Laravel dengan metodologi *Agile Scrum*, mencakup fitur manajemen kurikulum, visualisasi grafik *bar chart*, dan integrasi *tools* manajemen proyek seperti *GitLab* dan *Collab* [17].

Selain itu, penelitian oleh Akbar dan Irianto (2023) berjudul *Usulan Sistem Penilaian Capaian Pembelajaran Mata Kuliah (CPMK) dalam Menunjang Penilaian Capaian Pembelajaran Lulusan (CPL) Program Studi*. Mengusulkan sistem penilaian terintegrasi Capaian Pembelajaran Mata Kuliah (CPMK) berbasis Microsoft Excel untuk mendukung evaluasi Capaian Pembelajaran Lulusan (CPL) dalam konteks akreditasi IABEE. Sistem ini memfasilitasi penghitungan nilai akhir mata kuliah dan status CPMK secara paralel melalui pemetaan instrumen penilaian ke rubrik berbobot, serta menyediakan visualisasi hasil dalam bentuk grafik. Keunggulan utama

penelitian terletak pada penyederhanaan proses penilaian ganda dan pengurangan beban kerja dosen [18].

3. METODE PENELITIAN

Penelitian ini menggunakan metode *Agile Kanban* untuk mengembangkan *back-end* sistem informasi Capaian Pembelajaran Lulusan (CPL), dilakukan dengan tujuan mencapai fleksibilitas dan pengembangan berkelanjutan. Metode ini dipilih karena kemampuannya dalam mengelola tugas-tugas pengembangan secara iteratif dan adaptif, terutama untuk proyek yang memerlukan penyesuaian dinamis seperti sistem berbasis kurikulum pendidikan. Adapun tahapan yang dilakukan pada penelitian ditunjukkan pada Gambar 1.



Gambar 1. Metodologi Penelitian

3.1. Studi Literatur

Tahapan ini dilaksanakan sebagai landasan teoritis dan empiris untuk merumuskan ruang lingkup sistem informasi Capaian Pembelajaran Lulusan. Kegiatan ini mencakup pembelajaran terhadap penelitian terdahulu, standar akademik terkait CPL dan CPMK, serta metode pemetaan yang relevan sehingga kebutuhan fungsional dan non-fungsional sistem dapat diturunkan

dari bukti dan praktik terbaik yang ada. Hasil studi literatur digunakan untuk menyusun definisi istilah, kerangka konseptual, dan kriteria evaluasi yang akan menjadi acuan dalam analisis dan perancangan. Dokumentasi literatur juga mencatat celah penelitian sehingga rancangan sistem dapat menargetkan perbaikan atau fitur yang belum tersedia pada solusi sebelumnya.

3.2. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk mengidentifikasi dan menentukan fitur, alur kerja, dan batasan yang diperlukan agar sistem mampu memenuhi kebutuhan pengguna dan pemangku kepentingan. Proses ini meliputi pengumpulan kebutuhan melalui wawancara, observasi, dan studi dokumen, pemodelan use case, serta memprioritaskan kebutuhan berdasarkan urgensi dan dampak terhadap pengelolaan CPL. Dari analisis dihasilkan kebutuhan fungsional dan spesifikasi non-fungsional yang akan menjadi dasar perancangan sistem. Semua kebutuhan didokumentasikan dalam bentuk daftar kebutuhan yang terstruktur.

3.3. Perancangan Sistem

Diagram sistem yang akan dikembangkan dirancang terlebih dahulu sesuai dengan kebutuhan sistem. Pada tahap ini dibuat diagram seperti use case diagram dan activity diagram untuk memastikan hubungan antar komponen dan aliran informasi tersusun secara logis dan efisien. Dokumen perancangan berfungsi sebagai panduan teknis bagi tim pengembang dan sebagai acuan verifikasi saat pengujian.

3.4. Kanban Board

Setelah itu proses pembuatan *kanban board*, dilakukan untuk mengorganisir pekerjaan pengembangan secara visual dan iteratif dengan prinsip aliran kerja yang jelas. Pada langkah ini ditentukan kolom-kolom utama (*To-Do*, *In Progress*, *Testing*, dan *Done*), aturan pembatasan pekerjaan berjalan (*WIP limits*), kriteria definisi selesai untuk setiap *card*, serta kategori *card* yang merepresentasikan tugas, fitur, dan perbaikan. *Kanban board* juga memuat prioritas dan kompleksitas sehingga dapat memantau kemajuan pembangunan sistem, mengidentifikasi hambatan, dan mengalokasikan sumber daya secara responsif. Hasilnya adalah alat manajemen kerja yang

mendukung efisiensi, akurasi, dan pengambilan keputusan selama siklus pengembangan sistem.

3.5. Development

Tahapan pelaksanaan pengembangan mengikuti alur *kanban board* yang telah disusun. Setiap card diproses dari *To-Do* ke *In Progress* hingga *Testing* dan *Done* sesuai definisi tugas. Pengembangan dilakukan secara runtut, dokumentasi perubahan, dan penulisan kode yang konsisten untuk menjaga kualitas kode. Setiap tugas yang selesai pada tahap *In Progress* langsung diuji pada tahap *Testing* menggunakan pengujian *PHPUnit testing* dari sisi *server* dan pengujian fungsional dengan menggunakan *postman* dari sisi *client* untuk memastikan fitur bekerja sesuai dengan kebutuhan. Hasil pengujian dicatat dan bila ditemukan cacat, *card* dikembalikan ke *To-Do* atau *In Progress* untuk perbaikan sebelum dinyatakan *Done*.

3.6. Continuous Improvement

Diterapkan sebagai siklus evaluasi berkelanjutan yang memanfaatkan umpan balik dari hasil pengujian untuk menyempurnakan sistem. Mekanisme ini mencakup evaluasi berkala, analisis permasalahan, dan penyesuaian proses atau desain yang diperlukan untuk meningkatkan efisiensi dan kualitas produk. Perbaikan dapat berupa penyempurnaan alur pemetaan CPL dan CPMK atau penambahan fitur baru berdasarkan kebutuhan dari hasil diskusi secara beriringan dan berkelanjutan bersama tim dan para *stakeholder*. Dengan menerapkan prinsip *continuous improvement*, sistem dapat disesuaikan dengan kebutuhan yang lebih tepat sesuai dengan hasil diskusi bersama.

4. HASIL DAN PEMBAHASAN

4.1. Studi Literatur

Studi literatur pada penelitian ini mengkaji referensi terkait sistem informasi Capaian Pembelajaran Lulusan (CPL), metode pemetaan CPL dan CPMK, serta mekanisme pengolahan data nilai mahasiswa untuk membangun dasar teoritis yang kuat. Hasil kajian digunakan sebagai acuan perancangan *back-end API* yang mendukung penghitungan, analisis, dan pelaporan CPL sehingga implementasi sistem dapat memenuhi kebutuhan akademik dan operasional program studi secara optimal.

4.2. Analisis Kebutuhan Sistem

Tahapan ini bertujuan untuk mengidentifikasi dan merumuskan kebutuhan sistem. Dengan melakukan analisis kebutuhan sistem, setiap fitur yang akan diimplementasikan dapat disesuaikan dengan kebutuhan aktor yang terlibat. Kebutuhan sistem dibagi menjadi dua bagian, yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Adapun setelah dilakukannya identifikasi terhadap tujuan pengembangan aplikasi didapatkan kebutuhan fungsional terdapat pada tabel 1:

Tabel 1. Kebutuhan Fungsional

Role	Kebutuhan
Admin Universitas	1. Dapat melakukan login. 2. Dapat mengelola data prodi. 3. Dapat mengelola akun admin universitas. 4. Dapat mengelola akun admin prodi. 5. Dapat melihat hasil perhitungan. 6. Dapat melakukan logout.
Admin Prodi	1. Dapat melakukan login. 2. Dapat mengelola akun kaprodi. 3. Dapat mengelola akun dosen. 4. Dapat mengelola data mahasiswa. 5. Dapat mengelola data mata kuliah. 6. Dapat mengelola data kelas. 7. Dapat mengelola data CPL. 8. Dapat mengelola data CPMK. 9. Dapat melakukan pemetaan CPL. 10. Dapat melakukan pemetaan CPMK. 11. Dapat mengelola sub penilaian. 12. Dapat melihat hasil perhitungan. 13. Dapat melakukan logout.
Kaprodi	1. Dapat melakukan login. 2. Dapat melihat hasil perhitungan. 3. Dapat melakukan logout.
Dosen	1. Dapat melakukan login. 2. Dapat mengelola sub penilaian. 3. Dapat mengelola data nilai mahasiswa. 4. Dapat melihat hasil perhitungan. 5. Dapat melakukan logout.

Lalu terdapat kebutuhan non-fungsional sistem yang dapat dilihat pada tabel 2:

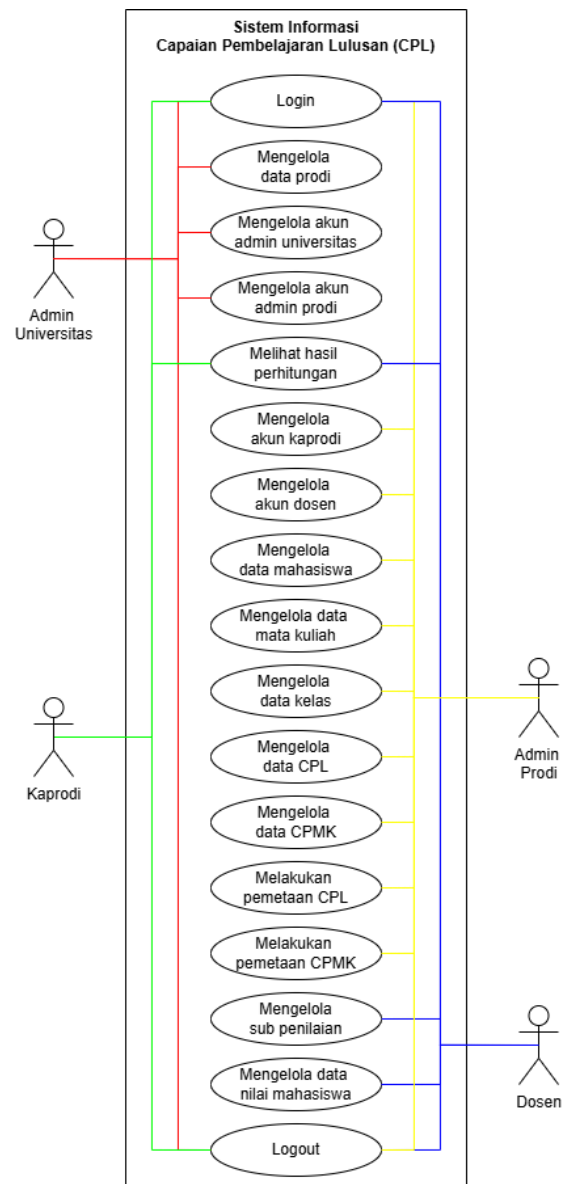
Tabel 2. Kebutuhan non-fungsional

Kategori	Kebutuhan
<i>Performance</i>	1. Setiap <i>endpoint API</i> harus memiliki <i>response time</i> kurang dari 1000 ms pada kondisi normal. 2. Waktu rata-rata yang diperlukan untuk <i>query database</i> yang kompleks seperti laporan CPL kurang dari 2000 ms.
<i>Security</i>	1. Menggunakan metode <i>authentication</i> yang aman (token dari <i>Laravel Sanctum</i>) dan menerapkan <i>authorization</i> berbasis <i>role</i> dan <i>permission</i> (menggunakan <i>Spatie Laravel-Permission</i>) agar hanya <i>user</i> dengan <i>role</i> tertentu yang dapat mengakses fungsi tertentu. 2. Data sensitif seperti <i>password</i> dan token harus dienkripsi atau di-hash dengan algoritma yang kuat serta penggunaan <i>HTTPS</i> wajib untuk mengamankan transmisi data.
<i>Reliability</i>	1. Sistem harus memastikan penanganan <i>error</i> yang konsisten (misalnya melalui <i>global exception handler</i>) dengan mengembalikan <i>status code</i> serta pesan <i>error</i> yang tepat agar mudah didiagnosis.
<i>Maintainability</i>	1. Penggunaan kerangka kerja dengan <i>layer</i> yang telah disesuaikan seperti <i>model</i>, <i>controller</i>, <i>service</i>, dan <i>layer</i> lainnya digunakan pada saat penulisan kode akan memudahkan pengembangan serta pemeliharaan di masa yang akan datang.
<i>Compatibility</i>	1. <i>API</i> menggunakan format data <i>JSON</i> untuk mempermudah integrasi dengan sistem eksternal.
<i>Testability</i>	1. Minimal 85% cakupan kode (<i>code coverage</i>) pada pengujian unit dan integrasi menggunakan <i>PHPUnit</i>. 2. Pengujian <i>endpoint API</i> menggunakan <i>postman</i> untuk memastikan semua <i>request</i> yang dilakukan oleh <i>user</i> menghasilkan <i>response</i> yang sesuai.
<i>Usability</i>	1. Penyediaan dokumentasi <i>API</i> yang lengkap dan jelas agar <i>developer</i> yang menggunakan <i>API</i> dapat melakukan integrasi dengan mudah. 2. Struktur <i>endpoint</i> dan <i>response JSON</i> yang konsisten sehingga <i>developer</i> yang menggunakan <i>API</i> dapat dengan mudah memahaminya.

4.3. Perancangan Sistem

Dalam tahap ini dilakukan pembuatan rancangan visual dan teknis yang meliputi diagram *use case diagram* dan *activity diagram* yang akan menjadi acuan dalam pengembangan sistem, sebagai berikut:

1. Use Case Diagram



Gambar 2. Use Case Diagram Sistem Informasi CPL

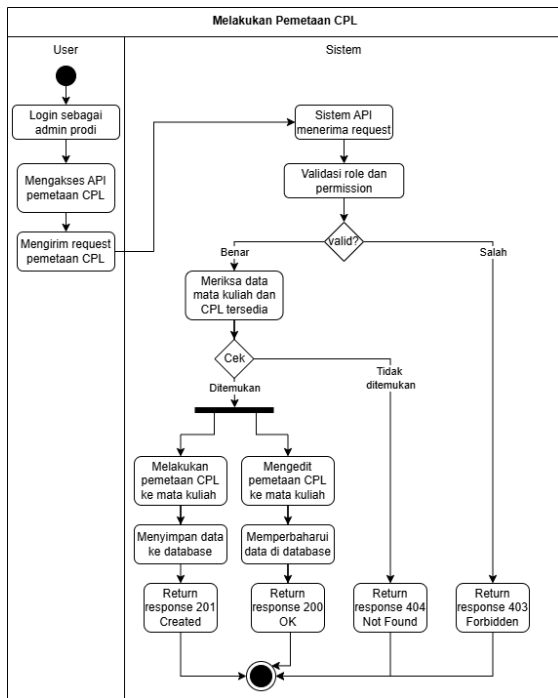
Gambar 2 merupakan diagram yang menggambarkan hubungan antara para aktor dan fungsi-fungsi utama dalam Sistem Capaian Pembelajaran Lulusan.

2. Activity Diagram

Beberapa diagram aktivitas disertakan untuk memvisualisasikan mekanisme kerja inti sistem dan hubungan antar aktor dengan sistem:

1. Activity Diagram Melakukan Pemetaan CPL

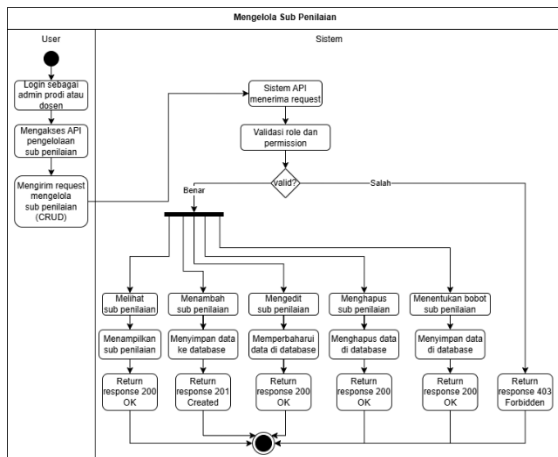
Gambar 3 memperlihatkan alur aktivitas melakukan pemetaan CPL yang dilakukan oleh *role* admin prodi.



Gambar 3. Activity Diagram Melakukan Pemetaan CPL

2. Activity Diagram Mengelola Sub Penilaian

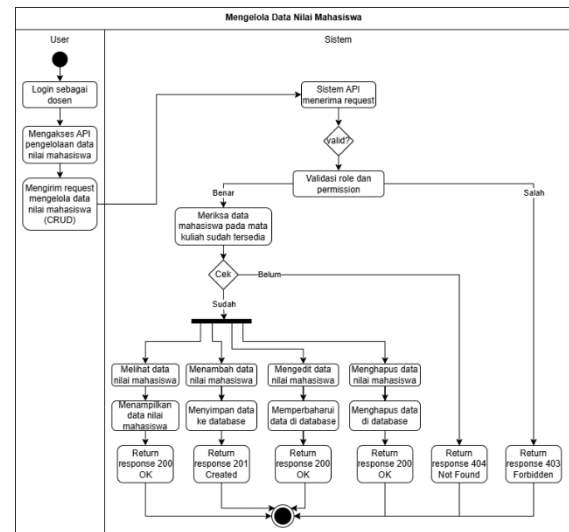
Gambar 4 memperlihatkan alur aktivitas untuk mengelola sub penilaian yang dilakukan oleh *role* admin prodi atau dosen.



Gambar 4. Activity Diagram Mengelola Sub Penilaian

3. Activity Diagram Mengelola Data Nilai Mahasiswa

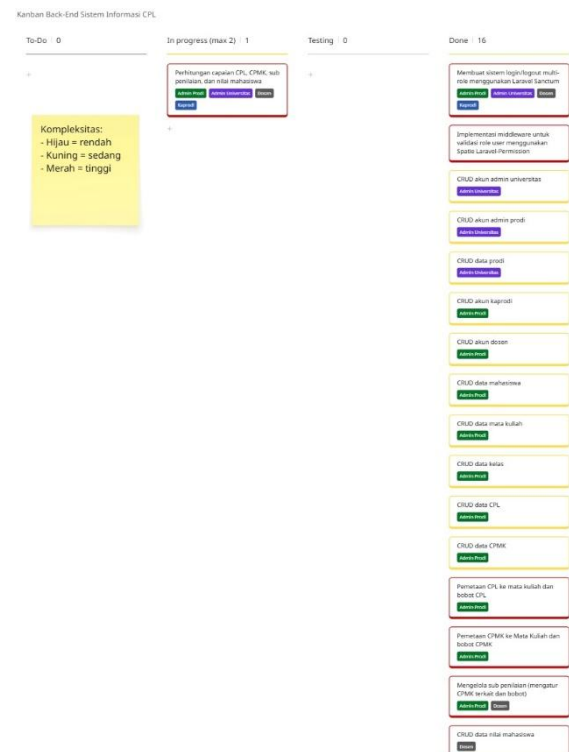
Gambar 5 memperlihatkan alur aktivitas untuk mengelola data nilai mahasiswa yang dilakukan oleh *role* dosen.



Gambar 5. Activity Diagram Mengelola Data Nilai Mahasiswa

4.4. Pembuatan Kanban Board

Gambar 6 merupakan hasil dari pembuatan *kanban board* menggunakan *mernggunakan* Miro.



Gambar 6. Kanban Board Back-End Sistem Informasi CPL

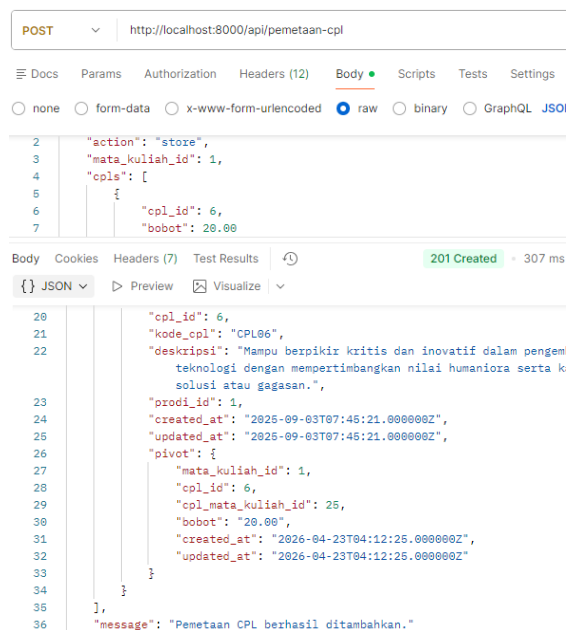
4.5. Development

Pada tahapan ini, proses development sistem *back-end* dilakukan, mulai dari penyiapan lingkungan pengembangan, perancangan

struktur basis data beserta relasi, implementasi logika bisnis dalam bentuk *service* dan *controller*, hingga pembuatan dan pengujian *API* untuk komunikasi dengan antarmuka pengguna. *Endpoint API* ditentukan berdasarkan desain ui/ux yang telah dibuat oleh rekan tim. Berikut beberapa hasil *API* yang telah diimplementasikan dan dapat dilihat menggunakan *postman*:

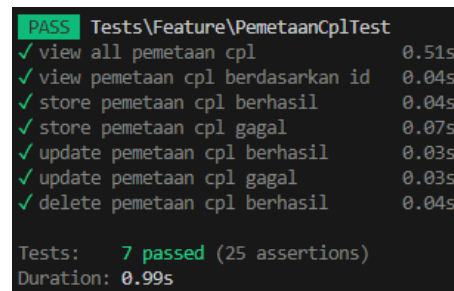
1. Endpoint Create Pemetaan CPL

Gambar 7 merupakan *endpoint create* untuk pemetaan cpl yang diakses melalui metode *HTTP POST* pada *endpoint* `/api/pemetaan-cpl`. Permintaan dikirimkan dalam format *JSON* yang memuat parameter *action* bernilai *store*, *mata_kuliah_id*, dan *cpls* yang berisi *cpl_id* yang akan dikaitkan beserta bobotnya. Setelah diterima, *server* melakukan validasi data cpl yang tersedia di program studi mata kuliah tersebut. Jika validasi berhasil, entitas data pemetaan cpl baru disimpan ke dalam basis data. Respons dengan status *HTTP 201 Created* kemudian dikembalikan beserta objek *JSON* yang memuat atribut *cpl_id*, *kode_cpl*, *deskripsi*, *prodi_id*, waktu pembuatan dan pembaruan, dan juga pivot data yang berisi data cpl yang dikaitkan ke mata kuliah beserta dengan bobotnya sehingga memastikan integritas dan konsistensi data pemetaan cpl.



Gambar 7. Endpoint Create Pemetaan CPL

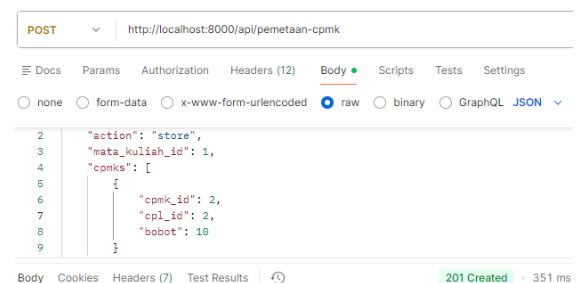
Gambar 8 merupakan pengujian untuk pemetaan cpl, meliputi tampilan daftar pemetaan cpl, detail berdasarkan id, pembuatan (berhasil dan gagal validasi), pembaruan (berhasil dan gagal validasi), serta penghapusan. Terdapat 7 skenario yang diuji dan lulus, dengan total 25 pemeriksaan (*assertions*) dan durasi eksekusi 0,99 detik.



Gambar 8. Testing Pemetaan CPL

2. Endpoint Create Pemetaan CPMK

Gambar 9 dan 10 merupakan *endpoint create* untuk pemetaan cpmk yang diakses melalui metode *HTTP POST* pada *endpoint* `/api/pemetaan-cpmk`. Permintaan dikirimkan dalam format *JSON* yang memuat parameter *action* bernilai *store*, *mata_kuliah_id*, dan *cpmks* yang berisi *cpmk_id*, *cpl_id* yang akan dikaitkan beserta bobotnya. Setelah diterima, *server* melakukan validasi data cpmk yang tersedia pada mata kuliah tersebut. Jika validasi berhasil, entitas data pemetaan cpmk baru disimpan ke dalam basis data. Respons dengan status *HTTP 201 Created* kemudian dikembalikan beserta objek *JSON* yang memuat atribut *cpmk_mata_kuliah_id*, *mata_kuliah_id*, *cpmk_id*, *cpl_id*, dan bobot yang dikaitkan ke mata kuliah tersebut sehingga memastikan integritas dan konsistensi data pemetaan cpmk.



Gambar 9. Endpoint Create Pemetaan CPMK

```
{ } JSON ▾ ▸ Preview ▢ Visualize ▾
1 {
2   "data": [
3     {
4       "cpmk_mata_kuliah_id": 2,
5       "mata_kuliah_id": 1,
6       "cpmk_id": 2,
7       "cpl_id": 2,
8       "bobot": "20.00"
9     },
10    {
11      "cpmk_mata_kuliah_id": 12,
12      "mata_kuliah_id": 1,
13      "cpmk_id": 2,
14      "cpl_id": 2,
15      "bobot": "10.00"
16    }
17  ]
18 }
```

Gambar 10. *Endpoint Create Pemetaan CPMK Lanjutan*

Gambar 11 merupakan pengujian untuk pemetaan cpmk, meliputi tampilan daftar pemetaan cpmk, detail berdasarkan id, pembuatan (berhasil dan gagal validasi), pembaruan (berhasil dan gagal validasi), serta penghapusan. Terdapat 7 skenario yang diuji dan lulus, dengan total 52 pemeriksaan (*assertions*) dan durasi eksekusi 1,30 detik.

```
PASS Tests\Feature\PemetaanCpmkTest
✓ view all pemetaan cpmk 0.80s
✓ view pemetaan cpmk berdasarkan id 0.05s
✓ store pemetaan cpmk berhasil 0.04s
✓ store pemetaan cpmk gagal 0.04s
✓ update pemetaan cpmk berhasil 0.04s
✓ update pemetaan cpmk gagal 0.03s
✓ delete pemetaan cpmk berhasil 0.04s

Tests: 7 passed (52 assertions)
Duration: 1.30s
```

Gambar 11. *Testing Pemetaan CPMK*

3. *Endpoint Create Mengelola Sub Penilaian*

Gambar 12 merupakan *endpoint create* untuk pembuatan sub penilaian yang diakses melalui metode *HTTP POST* pada *endpoint* `/api/kelola-sub-penilaian-oleh-admin-prodi`. Permintaan dikirimkan dalam format *JSON* yang memuat parameter *action* bernilai *store*, *penilaian_id*, *kelas_id*, dan *nama_sub_penilaian*. Setelah diterima, *server* melakukan validasi *penilaian_id* yang tersedia pada program studi. Jika validasi berhasil, entitas sub penilaian baru disimpan ke dalam basis data. Respons dengan status *HTTP 201 Created* kemudian dikembalikan beserta objek *JSON* yang memuat atribut *penilaian_id*, *kelas_id*, *nama_sub_penilaian*, *sub_penilaian_id*, serta waktu pembuatan dan pembaruan, sehingga memastikan integritas dan konsistensi data sub penilaian.

```
POST http://localhost:8000/api/kelola-sub-penilaian-oleh-admin-prodi
Docs Params Authorization Headers (12) Body Scripts Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON ▾
2 {
3   "action": "store",
4   "penilaian_id": 1,
5   "kelas_id": 2,
6   "nama_sub_penilaian": "UTS 1"
7 }
Body Cookies Headers (7) Test Results 201 Created 362 ms
{ } JSON ▾ ▸ Preview ▢ Visualize ▾
1 {
2   "data": {
3     "penilaian_id": 1,
4     "kelas_id": 2,
5     "nama_sub_penilaian": "UTS 1",
6     "updated_at": "2026-05-07T03:35:55.000000Z",
7     "created_at": "2026-05-07T03:35:55.000000Z",
8     "sub_penilaian_id": 40
9   },
10  "message": "Sub-penilaian berhasil ditambahkan."
11 }
```

Gambar 12. *Endpoint Create Mengelola Sub Penilaian*

Gambar 13 merupakan pengujian untuk mengelola sub penilaian, meliputi tampilan daftar sub penilaian, detail berdasarkan id, pembuatan (berhasil dan gagal validasi), pembaruan (berhasil dan gagal validasi), serta penghapusan. Terdapat 7 skenario yang diuji dan lulus, dengan total 49 pemeriksaan (*assertions*) dan durasi eksekusi 1,41 detik.

```
PASS Tests\Feature\SubPenilaianTest
✓ view all sub penilaian 0.63s
✓ view detail sub penilaian berdasarkan id 0.05s
✓ store sub penilaian berhasil 0.05s
✓ store sub penilaian validasi gagal 0.06s
✓ delete sub penilaian berhasil 0.07s
✓ store bobot sub penilaian berhasil 0.08s
✓ store bobot sub penilaian validasi gagal 0.20s

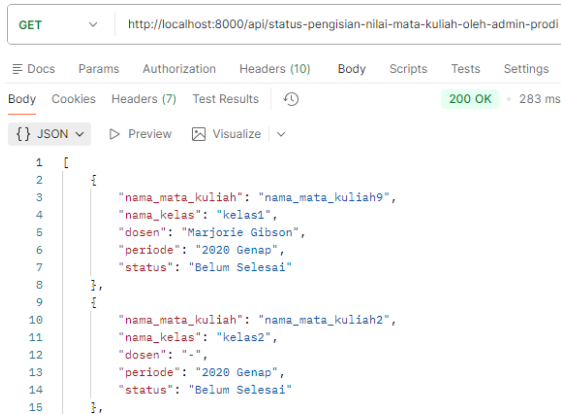
Tests: 7 passed (49 assertions)
Duration: 1.41s
```

Gambar 13. *Testing Mengelola Sub Penilaian*

4. *Endpoint Read Status Pengisian Nilai Mata Kuliah*

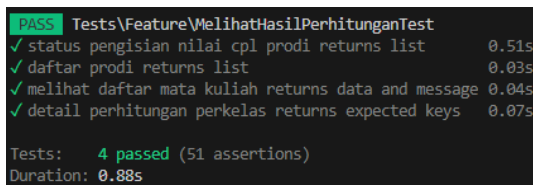
Gambar 14 merupakan *endpoint read* untuk pengambilan data status pengisian nilai mata kuliah di semua kelas yang ada pada program studi admin prodi tersebut. Diakses melalui metode *HTTP GET* pada *endpoint* `/api/status-pengisian-nilai-mata-kuliah-oleh-admin-prodi`. Permintaan tersebut diproses untuk mengambil seluruh entitas status pengisian nilai mata kuliah beserta atribut terkait, termasuk *nama_mata_kuliah*, *nama_kelas*, *dosen*, *periode* dan *status* dari basis data. Setiap objek status pengisian nilai mata kuliah dikembalikan dalam format *JSON* yang terstruktur, sehingga antarmuka pengguna dapat menampilkan daftar status pengisian nilai mata kuliah di semua kelas yang ada pada program studi admin prodi

tersebut akan digunakan untuk memantau seluruh aktivitas sistem cpl pada masing-masing program studi.



Gambar 14. *Endpoint Read Status Pengisian Nilai Mata Kuliah*

Gambar 15 merupakan pengujian untuk melihat hasil perhitungan, meliputi status pengisian nilai cpl, daftar prodi, daftar mata kuliah, serta detail perhitungan perkelas. Terdapat 4 skenario yang diuji dan lulus, dengan total 51 pemeriksaan (*assertions*) dan durasi eksekusi 0,88 detik.



Gambar 15. *Testing Melihat Hasil Perhitungan*

4.6. Progres *Development* Kebutuhan Fungsional

Sistem belum dapat beroperasi secara penuh karena pengembangan *API back-end* sistem CPL yang telah diimplementasikan belum sepenuhnya diselesaikan. Terdapat Hasil pengembangan 90 endpoint dari total 114 endpoint yang telah diimplementasikan, maka dapat disimpulkan persentase ketercapaian dalam pengembangan back-end API sistem CPL yang dapat dilihat pada tabel 3 berikut ini:

Tabel 3. Persentase Ketercapaian Pengembangan *Back-End API* Sistem CPL

Aktor	Jumlah Endpoint Selesai	Jumlah Endpoint Belum Selesai	Total Endpoint
Authentication	2	0	2
Admin Universitas	17	0	17
Admin Prodi	50	9	59
Kaprodi	7	9	16
Dosen	14	6	20
Total	90	24	114

$$\text{Persentase Ketercapaian} = \frac{90}{114} \times 100\% = 78,94\%$$

5. KESIMPULAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa *back-end API* Sistem Informasi Capaian Pembelajaran Lulusan (CPL) berhasil dianalisis, dirancang, dan dikembangkan dengan struktur yang terorganisasi sehingga mendukung pengelolaan data CPL, CPMK, kelas, mahasiswa, dan akun sesuai kebutuhan fungsional dan non-fungsional. Pengembangan dilakukan secara iteratif menggunakan pendekatan *Agile Kanban* untuk prioritas tugas dan pemantauan kemajuan, sehingga proses implementasi lebih adaptif. Analisis kebutuhan fungsional dan non-fungsional menjadi dasar pengembangan, dan pengujian fungsional menunjukkan bahwa *endpoint* yang telah diimplementasikan beroperasi sesuai spesifikasi. Hingga saat ini *endpoint API* yang telah diterapkan tercapai sebesar 78,94% dari 114 endpoint yang direncanakan, sedangkan 21,06% masih memerlukan pengembangan lanjutan dan integrasi lebih mendalam dengan data nyata.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada para narasumber dari Program Studi Informatika yang telah memberikan informasi dan masukan dalam proses pengembangan Sistem Capaian Pembelajaran Lulusan (CPL).

DAFTAR PUSTAKA

- [1] Ditjen Diktiristek Kemendikbudristek, *Panduan Penyusunan Kurikulum Pendidikan Tinggi Mendukung Merdeka Belajar-Kampus*

- Merdeka Menuju Indonesia Emas*, 5th ed. Jakarta, 2024.
- [2] Indonesia, *Peraturan Presiden Republik Indonesia Nomor 8 Tahun 2012 Tentang Kerangka Kualifikasi Nasional Indonesia*. Indonesia, 2012.
- [3] Indonesia, *Peraturan Menteri Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia Nomor 3 Tahun 2024 tentang Statuta Universitas Tadulako*. Indonesia, 2024.
- [4] E. Widarti *et al.*, *Buku Ajar Pengantar Sistem Informasi*, 1st ed. Surakarta: PT. Sonpedia Publishing Indonesia, 2024. [Online]. Available: <https://www.researchgate.net/publication/377153671>
- [5] S. Newman, *Building Microservices Designing Fine-Grained Systems*, 1st ed. Sebastopol: O'Reilly Media, 2015. [Online]. Available: <http://safaribooksonline.com>
- [6] M. Stauffer, *Laravel Up & Running A Framework for Building Modern PHP Apps*, 2nd ed. Sebastopol: O'Reilly Media, 2019. [Online]. Available: www.EBooksWorld.ir
- [7] Laravel LLC, "Laravel Sanctum," Laravel Documentation. Accessed: Jun. 08, 2026. [Online]. Available: <https://spatie.be/docs/laravel-permission/v6/introduction>
- [8] Spatie, "Spatie Laravel-Permission," Spatie Laravel-Permission Documentation. Accessed: Jun. 08, 2026. [Online]. Available: <https://spatie.be/docs/laravel-permission/v6/introduction>
- [9] A. A. Permana *et al.*, *Memahami Software Development Life Cycle*, 1st ed. Purbalingga: Eureka Media Aksara, 2023.
- [10] Andrew. Stellman and Jennifer. Greene, *Learning Agile*, 1st ed. Sebastopol: O'Reilly Media, 2014.
- [11] I. Sommerville, *Software engineering*, 10th ed. Harlow: Pearson Education Limited, 2016.
- [12] F. D. Silalahi, *MySQL (Structured Query Language) Manajemen Database*. Semarang: Yayasan Prima Agus Teknik, 2022.
- [13] O. James, *Mastering Postman: A Comprehensive Guide to Building End-to-End APIs with Testing, Integration and Automation PDF*. GitforGits, 2023.
- [14] L. Richardson and S. Ruby, *RESTful Web Services*. Sebastopol: O'Reilly, 2007.
- [15] D. Tanu Wijaya, I. Sumadikarta, and B. Panjaitan, "Analisa dan Perancangan Aplikasi Evaluasi Capaian Pembelajaran Lulusan," *Prosiding Seminar Nasional Inovasi Teknologi*, vol. 4, pp. 137–147, Jun. 2023, doi: 10.59134/prosidng.v4i.558.
- [16] M. H. Zikry, A. B. Prasetyo, and R. Septiana, "Implementasi Front-end Sistem Penilaian Capaian Pembelajaran Lulusan (CPL) dan Capaian Pembelajaran Mata Kuliah (CPMK) (Studi Kasus pada Teknik Komputer Undip)," *Jurnal Teknik Komputer*, vol. 3, no. 1, pp. 38–47, May 2024, doi: <https://doi.org/10.14710/jtk.v3i1.44282>.
- [17] O. Prasimsha and H. P. Putro, "Manajemen Proyek Pengembangan Perangkat Lunak Kerja Sama Perguruan Tinggi dan Software House (Studi Kasus: Pengembangan Sistem Penilaian Capaian Pembelajaran Lulusan Program Studi Informatika Universitas Islam Indonesia dan PT. Javan Cipta Solusi)," *journal.uui.ac.id/AUTOMATA*, vol. 1, no. 2, Jun. 2020.
- [18] M. Akbar and D. Irianto, "Usulan Sistem Penilaian Capaian Pembelajaran Mata Kuliah (CPMK) dalam Menunjang Penilaian Capaian Pembelajaran Lulusan (CPL) Program Studi," *Prosiding Seminar Nasional Teknik Industri*, vol. 1, pp. 974–984, Oct. 2023.