Pengembangan Aplikasi Penyusun RPS Berbasis OBE dengan LLM Metode RAD

A. Labib Fardany Faisal[1*], Yoza Wiratama[2], Mohammad Abdullah[3], Dimas Panji Eka Jala Putra[4]
[1,4] Teknologi Rekayasa Informatika Industri, Jurusan Teknologi Elektro, Politeknik Negeri Madura
[2] Peneliti Independen, Jakarta, Indonesia
[3] Teknik Bangunan Kapal, Jurusan Teknologi Kemaritiman, Politeknik Negeri Madura
[1]a.labib@poltera.ac.id, [2]yoza.wiratama@gmail.com, [3]fisabduh04@poltera.ac.id, [4]dimas.eka@poltera.ac.id
*Penulis Korespondensi
(A. Labib Fardany Faisal, Teknologi Rekayasa Informatika Industri, a.labib@poltera.ac.id)

ABSTRAK

Penyusunan Rencana Pembelajaran Semester (RPS) berbasis *Outcome-Based Education* (OBE) menuntut dosen menjaga keselarasan capaian pembelajaran lulusan (CPL), capaian pembelajaran mata kuliah (CPMK), sub-CPMK, rencana mingguan, dan penilaian, sehingga memakan waktu dan rentan tidak konsisten. Penelitian ini bertujuan mengembangkan aplikasi web penyusun RPS berbasis OBE yang memanfaatkan *large language model* (LLM) dan menguji fungsinya. Pengembangan memakai metode *Rapid Application Development* (RAD) empat fase, perancangan memakai *Unified Modeling Language* (UML) berupa *use case*, *activity*, dan *class diagram*, sedangkan pengujian memakai *black box testing* dengan 18 kasus uji terhadap tiga RPS hasil generate sebagai data uji. Aplikasi yang dihasilkan menerima data mata kuliah dan CPL, menyusun CPMK, sub-CPMK berformat ABCD (*audience*, *behavior*, *condition*, *degree*), rencana 16 pertemuan, dan sistem penilaian; menyediakan tiga model dengan lapisan pemeriksaan bertingkat; mengelola kuota; serta mengekspor hasil ke Excel. Dari 54 hasil uji, 46 sesuai. Seluruh 42 hasil uji fungsional sesuai, sedangkan delapan temuan berada pada kualitas isi: level *taxonomy bloom* yang tidak sesuai kata kerja, sub-CPMK kembar, capaian sikap yang diberi level kognitif, dan pengulangan kemampuan akhir pada rencana mingguan. Aplikasi memenuhi kebutuhan fungsional yang diuji, tetapi validasi aturan perlu diperluas ke pemeriksaan semantik dan hasil tetap memerlukan tinjauan dosen. Pengujian terbatas pada tiga RPS tanpa penilaian pengguna.

Kata kunci: RPS, OBE, large language model, RAD, black box testing

ABSTRACT

Preparing an Outcome-Based Education (OBE) semester learning plan (RPS) requires lecturers to keep graduate learning outcomes, course learning outcomes, sub-outcomes, the weekly plan, and assessment aligned, which is time-consuming and prone to inconsistency. This study develops a web application that drafts OBE-based RPS with a large language model (LLM) and tests its functions. Development followed the four-phase Rapid Application Development (RAD) method, design used the Unified Modeling Language (UML) with use case, activity, and class diagrams, and testing used black box testing with 18 test cases on three generated RPS as test data. The application accepts course data and graduate outcomes, drafts course outcomes, ABCD-format (audience, behavior, condition, degree) sub-outcomes, a 16-meeting plan, and an assessment scheme; offers three models with tiered checking layers; manages a usage quota; and exports results to Excel. Of 54 test results, 46 conformed. All 42 functional results conformed, while eight findings concerned content quality: Bloom taxonomy levels that do not match the action verb, duplicate sub-outcomes, attitude outcomes given cognitive levels, and repeated final abilities in the weekly plan. The application meets the functional requirements tested, but its rule validation should be extended to semantic checks and results still need lecturer review. Testing was limited to three RPS without user evaluation.

Keywords: *semester learning plan, OBE, large language model, RAD, black box testing*

1. PENDAHULUAN

Rencana Pembelajaran Semester (RPS) adalah dokumen perencanaan perkuliahan yang menjadi rujukan dosen dan mahasiswa selama satu semester. Pada pendidikan tinggi berbasis *Outcome-Based Education* (OBE), RPS harus menurunkan capaian pembelajaran lulusan (CPL) menjadi capaian pembelajaran mata kuliah (CPMK) dan sub-CPMK, lalu menyelaraskannya dengan kegiatan belajar dan penilaian sesuai prinsip *constructive alignment* [1]. Penerapan OBE tidak sederhana karena menuntut penataan ulang capaian, teknik penilaian, dan peran dosen [2].

Kesulitan itu tampak pada praktik. Yusnaldi dkk. melaporkan bahwa RPS di sebuah universitas belum berbasis OBE meskipun sosialisasi dan lokakarya bagi dosen telah dilakukan pada 2023, sehingga perangkat

pembelajarannya dikembangkan ulang dengan model 4-D [3]. Rahmaniati dkk. menempuh jalur pelatihan penyusunan RPS OBE bagi dosen [4]. Penyusunan yang baik juga menuntut rumusan capaian yang terukur dan berjenjang menurut level kognitif *taxonomy bloom* revisi [5], serta memuat capaian nonkognitif seperti komunikasi dan kerja sama yang dalam kajian sistematis dilatihkan melalui intervensi pada kurikulum [6].

Large language model (LLM) membuka peluang mengotomatiskan sebagian pekerjaan tersebut. Survei terbaru merangkum perkembangan LLM pada tahap pralatih, pascalatih, dan pemanfaatan melalui rekayasa instruksi (*prompt engineering*) [7]. Kasneci dkk. menguraikan peluang LLM bagi pendidikan sekaligus risiko yang menuntut pengawasan manusia [8]. Pada perancangan kurikulum, Sridhar dkk. mengevaluasi 127 tujuan pembelajaran yang dibangkitkan GPT-4 dan menemukan bahwa rumusannya sebagian besar bermakna, diawali kata kerja aksi, dan berada pada level *taxonomy bloom* yang sesuai [9]. Almatrafi dan Johri membandingkan strategi instruksi pada GPT-4 dengan BERT untuk mengklasifikasikan tujuan pembelajaran mata kuliah ke level Bloom, dan BERT akhirnya lebih akurat [5]. Pendekatan berbasis aturan juga telah dipakai untuk membangkitkan rencana studi secara otomatis [10].

Dari literatur yang ditelaah, belum ditemukan laporan rancang bangun aplikasi yang menyusun RPS OBE utuh, dari CPMK, sub-CPMK, rencana 16 pertemuan, hingga sistem penilaian, dengan pemeriksaan berlapis dan bukti pengujian fungsinya. Studi [9] menilai rumusan tujuan belajar, bukan dokumen RPS lengkap, dan studi [10] menyusun rencana studi mahasiswa, bukan RPS mata kuliah. Artikel ini melaporkan (1) pengembangan aplikasi web penyusun RPS berbasis OBE dengan LLM menggunakan metode RAD, (2) model perancangannya dengan UML, dan (3) pengujian *black box* terhadap tiga RPS hasil generate sebagai data uji.

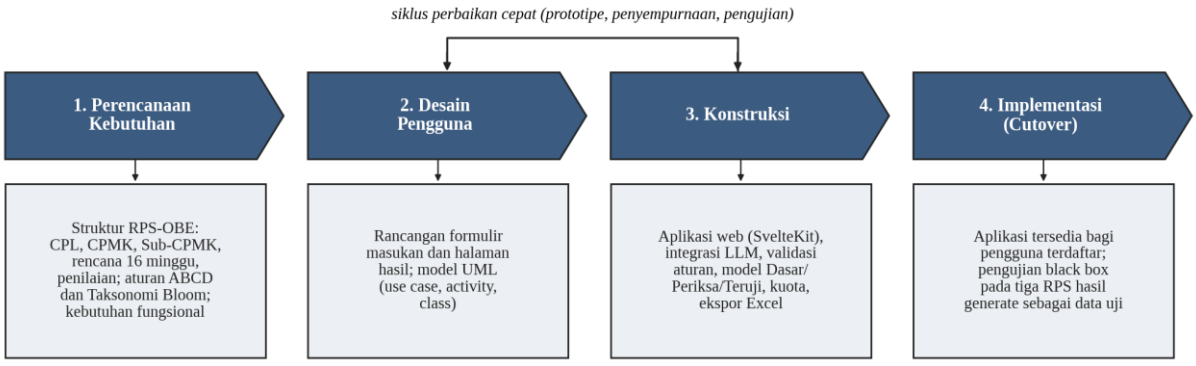
Kebaruan dari sistem ini terdiri dari 2 aspek yaitu metode RAD dan lapisan pemeriksaan bertingkat (Dasar, Periksa, dan Teruji). Dengan RAD, alur pemikiran LLM dapat diarahkan dengan jelas di awal sehingga meminimalisir pemikiran AI yang liar. Dengan menetapkan alur yang *rigid*, hasil akan lebih sejalan dengan struktur baku RPS yang telah ditentukan oleh SN-DIKTI, sebagai contoh aturan ABCD, bentuk dan metode pembelajaran, indikator dan kriteria penilaian, serta pembobotan nilai. Hal ini memberikan kebaruan karena melakukan *prompting* yang berisi kaidah SN Dikti pada layanan AI publik seperti ChatGPT dan Gemini akan sangat panjang dan tidak *user friendly*. Adapun lapisan pemeriksaan bertingkat (Dasar, Periksa, dan Teruji) berguna sebagai *quality control* otomatis untuk validasi output LLM yang masih menyimpang dari kaidah kaidah yang ditentukan. User dapat memilih salah satu dari 3 pemeriksaan ini sesuai dengan biaya yang dimiliki. Kebaruan yang diberikan di sini adalah meminimalisir revisi yang memaksa user untuk mengetikkan prompt tambahan untuk mengoreksi hasil dari LLM.

2. METODOLOGI PENELITIAN

Penelitian ini merupakan penelitian pengembangan perangkat lunak dengan tiga komponen: metode pengembangan RAD, pemodelan UML, dan pengujian *black box*.

2.1 Metode pengembangan

RAD adalah model pengembangan yang menekankan siklus pendek dan prototipe [11][12]. Empat fase baku RAD dipetakan pada penelitian ini seperti pada Gambar 1: (1) perencanaan kebutuhan, yaitu menetapkan struktur RPS-OBE, aturan rumusan sub-CPMK format ABCD, dan kebutuhan fungsional; (2) desain pengguna, yaitu merancang formulir masukan, halaman hasil, dan model UML; (3) konstruksi, yaitu membangun aplikasi web, mengintegrasikan LLM melalui API, dan menyusun validasi aturan; serta (4) implementasi (*cutover*), yaitu menyediakan aplikasi bagi pengguna terdaftar dan mengujinya. Panah umpan balik pada Gambar 1 menandai siklus perbaikan antara fase 2 dan 3. Artikel ini tidak melaporkan jumlah atau durasi iterasi karena berfokus pada hasil rancangan dan pengujian.



Gambar 1. Pemetaan fase RAD pada pengembangan aplikasi

Validasi aturan dibangun sebagai modul pemeriksa berbasis aturan (*rule-based*) di dalam mesin penyusun RPS pada aplikasi web dengan framework SvelteKit (Gambar 2). Aturan yang diperiksa bersumber dari dua rujukan. Rujukan pertama adalah SN Dikti [16], yang menetapkan bahwa CPL disusun ke dalam mata kuliah dan CPMK berkontribusi pada CPL. Dari ketentuan ini diturunkan aturan integritas rantai capaian, yaitu setiap sub-CPMK harus menunjuk CPMK dan CPL yang ada. Rujukan kedua adalah konvensi penyusunan RPS berbasis OBE, yaitu format rumusan ABCD, pemilihan kata-kata baku untuk bentuk dan metode pembelajaran, serta indikator dan kriteria penilaian, juga konsistensi aritmetika berupa total bobot 100%. Modul ini berupa fungsi pemeriksa yang dieksekusi oleh kode aplikasi dan dijalankan pada semua model, setelah keluaran LLM diterima dan sebelum RPS disimpan sebagai versi baru (Gambar 4). Pemeriksaannya bersifat struktural, yaitu keberadaan, kelengkapan, dan konsistensi angka.

2.2 Pemodelan UML

Model disusun dengan tiga diagram UML [13]: *use case diagram* untuk kebutuhan fungsional, *activity diagram* untuk alur pembuatan RPS, dan *class diagram* untuk struktur data. Ketiganya digambar dari fitur aplikasi yang sudah berjalan.

2.3 Pengujian black box dan data uji

Pengujian *black box* memeriksa fungsi berdasarkan masukan dan keluaran tanpa melihat kode program [14][15]. Kasus uji diturunkan dari spesifikasi fungsional dan aturan OBE dengan partisi ekuivalensi untuk kelas keadaan (misalnya kuota cukup atau tidak) dan pemeriksaan nilai batas untuk bobot dan jumlah pertemuan [14]. Kasus dibagi menjadi uji fungsional (T01–T14) dan uji kualitas isi (T15–T18). Data uji adalah tiga RPS yang sudah dibangkitkan aplikasi (Tabel 1).

Tabel 1. Profil data uji (RPS hasil generate)

Kode	Mata kuliah	Program studi (jenjang)	SKS	Smt	CPL	CPMK	Sub-CPMK	Komp. nilai	Bobot mingguan
D1	Pemrograman Terapan	Teknik Informatika (S1)	4	3	5	6	13	5	100%
D2	Pemrograman Dasar	Teknologi Rekayasa Otomasi (D4)	2	2	1	6	12	5	100%
D3	Elektronika Digital dan Mikrokontroler	Teknik Elektro (S1)	4	2	5	6	12	6	100%

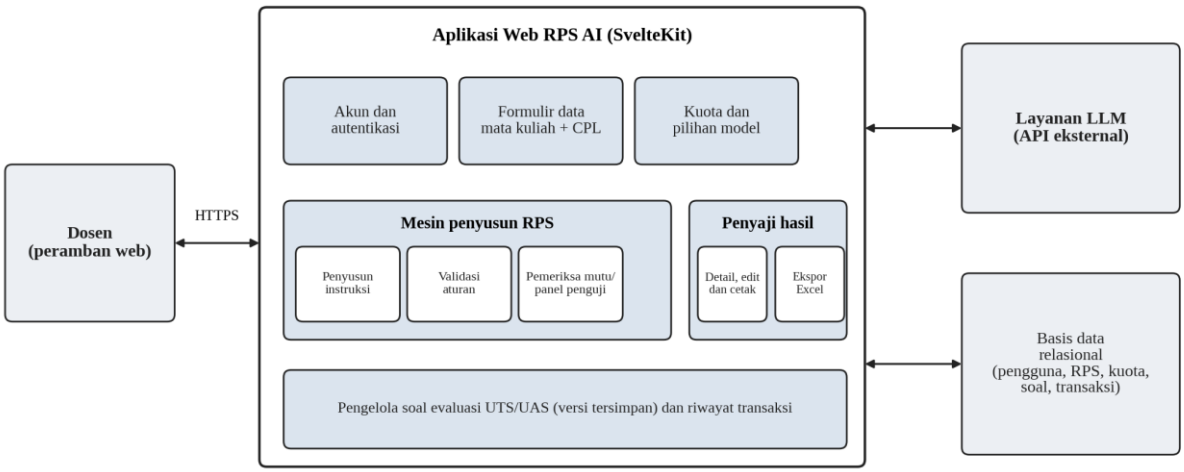
Pengujian bersifat observasional. Penguji membuka halaman formulir dan halaman detail setiap RPS, lalu membandingkan tampilan dan isinya dengan hasil yang diharapkan. Tidak ada RPS baru yang dibangkitkan dan tidak ada data yang diubah. Akibatnya, fungsi yang memakai kuota (generate ulang dan pembuatan draf soal) serta fungsi unduh tidak dieksekusi; yang diuji adalah keadaan antarmuka serta kelengkapan dan konsistensi keluaran. Pada T15, level Bloom dinilai dengan kata kerja berlevel baku pada *taxonomy bloom* revisi [5], yaitu menjelaskan dan menginterpretasikan (C2), menerapkan dan mengimplementasikan (C3), menganalisis (C4), serta mengevaluasi (C5); kata kerja lain tidak dinilai. Hasil setiap kasus dicatat sebagai Sesuai (S) atau Temuan (T).

3. HASIL DAN PEMBAHASAN

3.1 Perencanaan kebutuhan dan arsitektur

Kebutuhan fungsional yang ditetapkan pada fase pertama adalah: (a) registrasi dan autentikasi pengguna; (b) pengelolaan data mata kuliah dan CPL; (c) pemilihan model dan pembangkitan RPS; (d) validasi aturan atas hasil pembangkitan; (e) penayangan, penyuntingan, pencetakan, dan ekspor RPS ke Excel; (f) pembuatan draf soal evaluasi UTS/UAS yang tersimpan per versi; serta (g) pengelolaan kuota dan riwayat transaksi. Struktur RPS-OBE yang dituju berjenjang dari CPL ke CPMK, sub-CPMK, rencana 16 pertemuan, dan sistem penilaian; sub-CPMK dirumuskan dengan format ABCD (*audience*, *behavior*, *condition*, *degree*) dan diberi level Bloom C1–C6 [5].

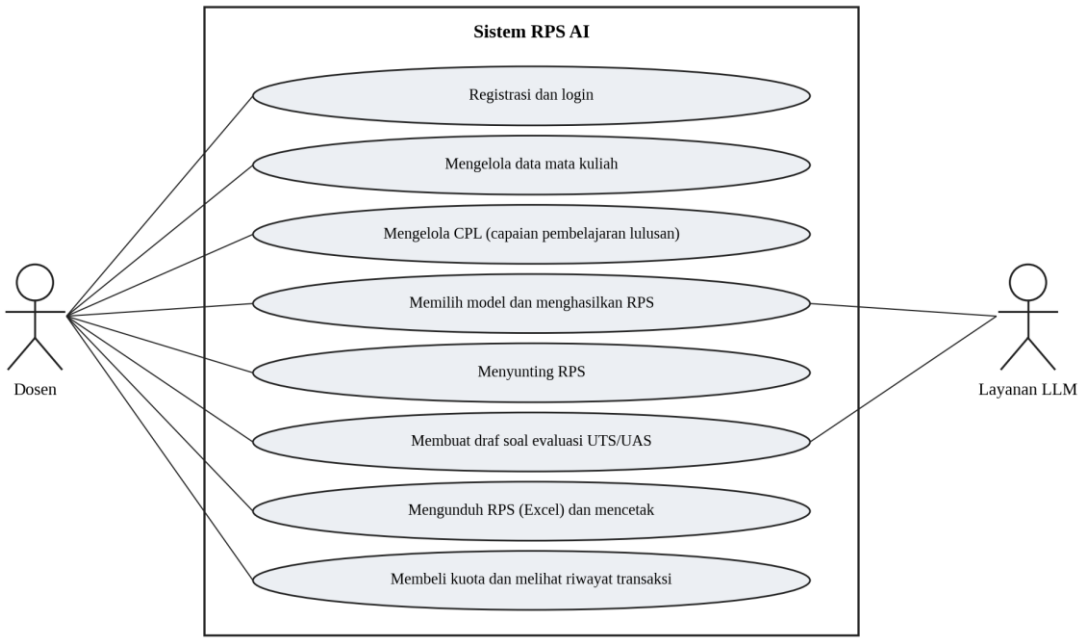
Arsitektur aplikasi terdiri atas tiga lapis (Gambar 2). Dosen mengakses aplikasi web yang dibangun dengan kerangka kerja SvelteKit melalui peramban. Aplikasi memuat modul akun, formulir data, kuota dan pilihan model, mesin penyusun RPS (penyusun instruksi, validasi aturan, pemeriksa mutu atau panel penguji), penyaji hasil, dan pengelola soal. Aplikasi memanggil layanan LLM eksternal melalui API dan menyimpan pengguna, RPS, kuota, soal, dan transaksi pada basis data relasional.



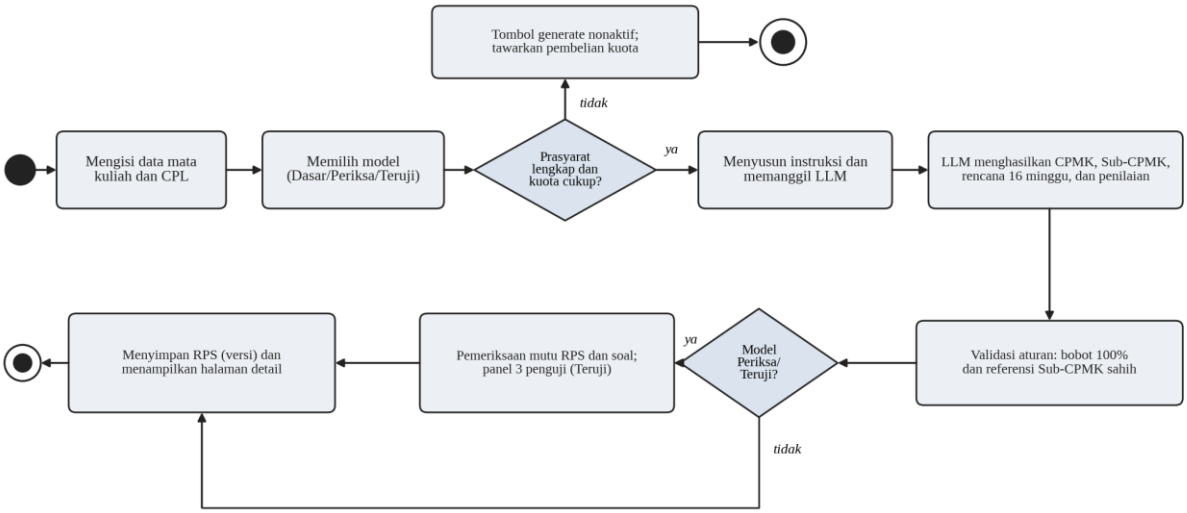
Gambar 2. Arsitektur aplikasi web penyusun RPS

3.2 Desain pengguna dan model UML

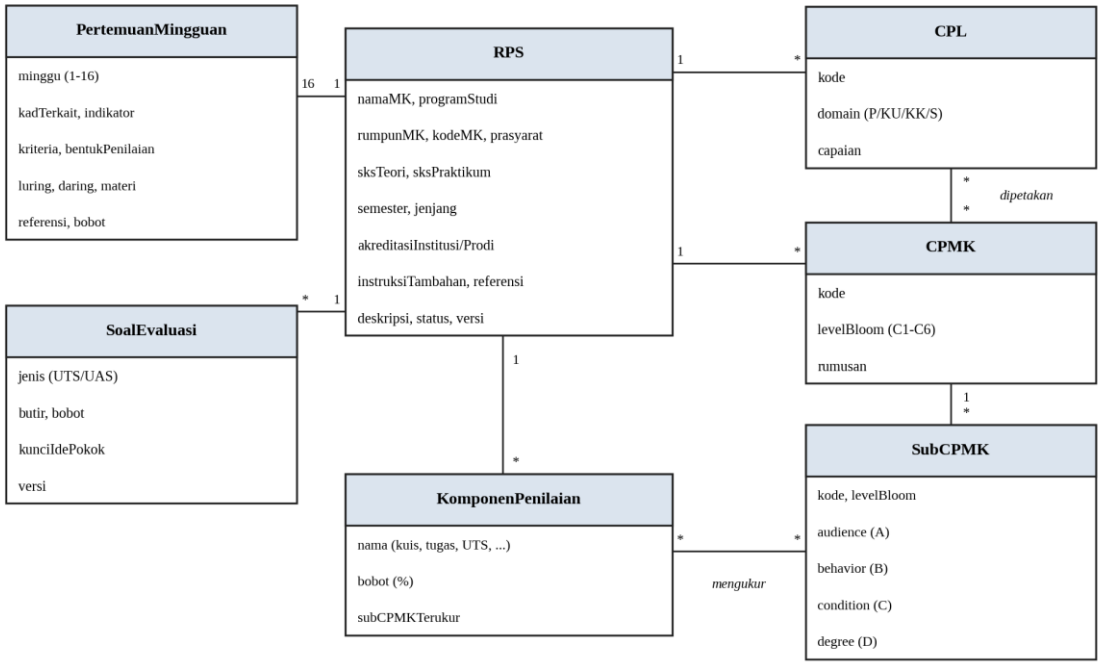
Use case diagram (Gambar 3) memuat delapan *use case*. Aktor utama adalah dosen; layanan LLM menjadi aktor sekunder pada *use case* pembangkitan RPS dan pembuatan draf soal. *Activity diagram* (Gambar 4) menggambarkan alur pembuatan RPS: setelah data mata kuliah dan CPL diisi dan model dipilih, sistem memeriksa kelengkapan prasyarat dan kecukupan kuota; jika belum terpenuhi, tombol generate dinonaktifkan dan pembelian kuota ditawarkan. Jika terpenuhi, sistem menyusun instruksi, memanggil LLM, memvalidasi aturan (bobot 100% dan referensi sub-CPMK yang sah), menjalankan pemeriksaan mutu atau panel penguji untuk model Periksa dan Teruji, lalu menyimpan RPS sebagai versi baru. *Class diagram* (Gambar 5) menunjukkan satu RPS yang berelasi dengan banyak CPL, CPMK, komponen penilaian, dan soal evaluasi, serta 16 pertemuan mingguan; CPL dan CPMK berelasi banyak-ke-banyak, sedangkan satu CPMK memiliki banyak sub-CPMK.



Gambar 3. Use case diagram sistem RPS AI



Gambar 4. Activity diagram pembuatan RPS



Gambar 5. Class diagram struktur data RPS

3.3 Konstruksi: model dan lapisan pemeriksaan

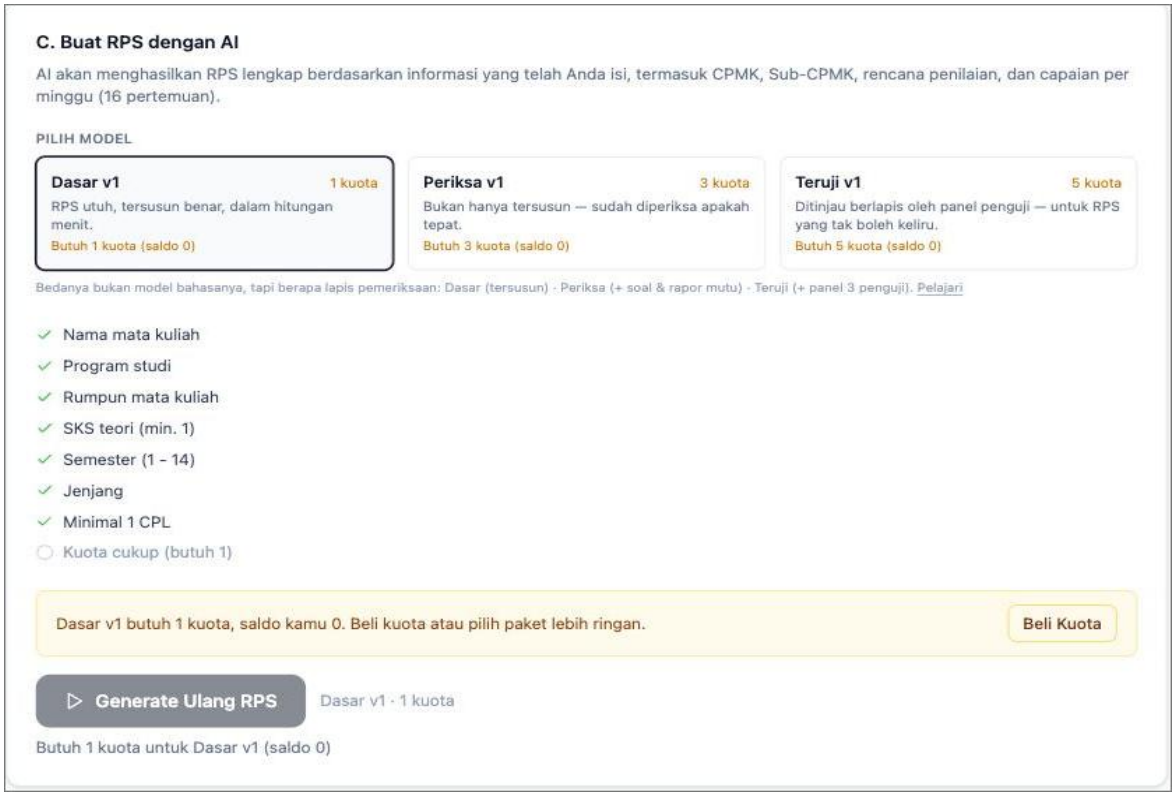
Aplikasi menyediakan tiga model yang memakai mesin pembangkitan yang sama tetapi berbeda pada jumlah lapisan pemeriksaan (Tabel 2). Pengguna membayar dengan kuota: 1 kuota untuk Dasar, 3 untuk Periksa, dan 5 untuk Teruji. Pada model Periksa ditambahkan pemeriksaan soal dan rapor mutu, dan pada Teruji ditambahkan panel tiga penguji. Validasi aturan berjalan pada semua model dan memeriksa bahwa total bobot 100% serta referensi sub-CPMK sahih.

Tabel 2. Model pada aplikasi dan lapisan pemeriksaannya

Model	Kuota	Lapisan pemeriksaan	Keterangan pada aplikasi
Dasar v1	1	Tersusun	RPS utuh, tersusun benar
Periksa v1	3	Dasar + soal dan rapor mutu	RPS diperiksa ketepatannya
Teruji v1	5	Periksa + panel 3 penguji	RPS ditinjau berlapis oleh panel penguji

3.4 Implementasi: antarmuka

Formulir masukan terdiri atas tiga bagian. Bagian A memuat informasi dasar: nama mata kuliah, program studi atau bidang ilmu, rumpun, SKS teori, semester, dan jenjang (wajib), serta kode, SKS praktikum, akreditasi, prasyarat, instruksi tambahan untuk AI, dan referensi (opsional). Bagian B memuat CPL dengan kode, domain (P, KU, KK, S), dan rumusan capaian; minimal satu CPL diperlukan. Bagian C memuat pilihan model dan daftar periksa prasyarat (Gambar 6). Halaman detail menampilkan CPL, CPMK beserta pemetaannya, sub-CPMK berformat ABCD, rencana mingguan, sistem penilaian, dan bagian soal evaluasi (Gambar 7).



Gambar 6. Bagian pembuatan RPS pada formulir: pilihan model dan daftar periksa prasyarat



Gambar 7. Contoh keluaran D1: sub-CPMK format ABCD dan baris rencana mingguan

3.5 Hasil pengujian black box

Tabel 3 merangkum 18 kasus uji pada tiga data uji, sehingga terdapat 54 hasil. Sebanyak 46 hasil (85,2%) sesuai dan delapan (14,8%) berupa temuan. Seluruh 42 hasil uji fungsional (T01–T14) sesuai. Delapan temuan seluruhnya berada pada uji kualitas isi (T15–T18); dari 12 hasil pada kelompok ini, empat sesuai.

Tabel 3. Hasil pengujian black box pada tiga RPS (S = sesuai, T = temuan)

Kode	Kasus uji dan hasil yang diharapkan	D1	D2	D3
T01	Formulir memuat bagian A (informasi dasar), B (CPL), dan C (pembuatan RPS) dengan penanda bidang wajib	S	S	S
T02	Pilihan model menampilkan Dasar, Periksa, dan Teruji dengan biaya 1, 3, dan 5 kuota serta saldo pengguna	S	S	S
T03	Daftar periksa prasyarat memuat delapan butir (nama mata kuliah, program studi, rumpun, SKS teori, semester 1–14, jenjang, minimal 1 CPL, kuota)	S	S	S
T04	Tombol generate nonaktif saat kuota kurang (saldo 0), disertai pesan dan tombol pembelian kuota	S	S	S
T05	Informasi dasar (nama, program studi, kelompok, SKS, semester, jenjang) tampil sesuai profil data uji	S	S	S
T06	Setiap CPMK berlevel Bloom dan dipetakan ke CPL; setiap sub-CPMK terhubung ke CPMK dan CPL	S	S	S
T07	Setiap sub-CPMK berformat ABCD lengkap dengan derajat berupa persentase	S	S	S
T08	Rencana memuat 16 pertemuan: 14 baris mingguan, UTS pada minggu ke-8, UAS pada minggu ke-16	S	S	S
T09	Jumlah bobot mingguan sama dengan 100%	S	S	S
T10	Setiap KAD (13, 12, dan 12 buah) muncul pada minimal satu minggu	S	S	S
T11	Rumus alokasi waktu TM dan PT+BM konsisten dengan jumlah SKS (4, 2, dan 4 SKS)	S	S	S
T12	Bobot komponen penilaian berjumlah 100% dan setiap sub-CPMK diukur oleh minimal satu komponen	S	S	S
T13	Halaman detail menyediakan kontrol Edit, Download Excel, Cetak, dan pembuatan soal (keberadaan kontrol)	S	S	S
T14	Bagian soal UTS/UAS menampilkan status kosong yang konsisten (“Belum ada soal”, total 0% dari 100%)	S	S	S
T15	Level Bloom sesuai kata kerja berlevel baku (jumlah yang sesuai dari yang dinilai)	S 8/8	T 3/6	T 3/6
T16	Setiap sub-CPMK memiliki perilaku (B) yang berbeda	S	T 2 pasang	S
T17	Capaian kerja sama, komunikasi, dan sikap tidak diberi level kognitif (jumlah yang diberi level)	T 4/13	S 0/12	T 4/12
T18	Rencana mingguan tidak didominasi pengulangan KAD (penempatan yang mengulang)	T 15/28	T 14/26	T 16/28

Uji fungsional menunjukkan bahwa aplikasi memenuhi kebutuhan yang diuji. Ketiga RPS memuat 16 pertemuan dengan total bobot mingguan 100% (misalnya D1: 7, 7, 9, 7, 7, 7, 7, lalu tujuh minggu masing-masing 7%), seluruh KAD dan sub-CPMK tercakup, dan alokasi waktu mengikuti SKS. Hasil ini konsisten dengan adanya validasi aturan pada alur pembangkitan (Gambar 4), tetapi pengujian *black box* tidak dapat membuktikan bahwa validator itulah penyebabnya. Prinsip serupa, yaitu aturan yang menjaga keluaran otomatis tetap sah, dipakai pada pembangkit rencana studi berbasis aturan [10].

Temuan berada pada kualitas isi, yang tidak tertangkap oleh validasi struktural; rinciannya untuk T15 sampai T17 disajikan pada Tabel 4.

Tabel 4. Rincian temuan T15–T17 (SC = sub-CPMK)

Kode	Data-SC	Perilaku (B) pada sub-CPMK	Level di aplikasi	Keterangan
T15	D2-SC1	menganalisis konsep dasar pemrograman	C2	kata kerja setara C4
T15	D2-SC7	mengevaluasi solusi pemrograman	C4	kata kerja setara C5

Kode	Data-SC	Perilaku (B) pada sub-CPMK	Level di aplikasi	Keterangan
T15	D2-SC8	mengevaluasi solusi pemrograman	C4	kata kerja setara C5
T15	D3-SC3	menganalisis rangkaian kombinasi	C3	kata kerja setara C4
T15	D3-SC4	menganalisis rangkaian sekuensial	C3	kata kerja setara C4
T15	D3-SC12	menerapkan etika dalam proyek	C4	kata kerja setara C3
T16	D2-SC7, SC8	mengevaluasi solusi pemrograman	C4, C4	perilaku sama pada dua sub-CPMK
T16	D2-SC11, SC12	membandingkan pendekatan pemrograman	C3, C3	perilaku sama pada dua sub-CPMK
T17	D1-SC6	berkolaborasi dalam tim	C2	kerja sama
T17	D1-SC7	mengomunikasikan ide dan solusi	C2	komunikasi
T17	D1-SC10	mendemonstrasikan etika pemrograman	C3	sikap
T17	D1-SC11	mendemonstrasikan tanggung jawab sosial	C3	sikap
T17	D3-SC7	bekerja sama dalam tim	C3	kerja sama
T17	D3-SC8	mengelola konflik dalam tim	C3	kerja sama
T17	D3-SC11	menunjukkan sikap profesional	C4	sikap
T17	D3-SC12	menerapkan etika dalam proyek	C4	sikap

Pada T15, enam dari 20 sub-CPMK yang dinilai (30%) berlevel tidak sesuai kata kerjanya: pada D2, “menganalisis konsep dasar pemrograman” berlabel C2 serta dua sub-CPMK “mengevaluasi solusi pemrograman” berlabel C4; pada D3, dua sub-CPMK “menganalisis rangkaian” berlabel C3 dan “menerapkan etika dalam proyek” berlabel C4. Angka ini tidak dapat dibandingkan langsung dengan hasil Sridhar dkk. [9] yang menilai tujuan pembelajaran mata kuliah dengan instruksi khusus pada GPT-4, dan sampel penelitian ini hanya 37 sub-CPMK.

Taxonomy bloom revisi yang dipakai bersifat kognitif [5], sedangkan komunikasi dan kerja sama termasuk keterampilan lunak yang dilatihkan melalui kurikulum [6]; dengan demikian aplikasi perlu memperlakukan domain sikap dan keterampilan secara terpisah. Pada T18, lebih dari separuh penempatan KAD (54% sampai 57%; 14 sampai 16 dari 26 sampai 28 penempatan) mengulang KAD yang sudah muncul pada minggu sebelumnya; pada D1 dan D3 seluruh KAD sudah tersaji pada paruh pertama sehingga semua penempatan setelah UTS bersifat pengulangan. Pengulangan dapat dimaksudkan sebagai penguatan, tetapi aturan “semua KAD tercakup” terpenuhi tanpa jaminan adanya progresi materi.

3.6 Tindak Lanjut Hasil Pengujian

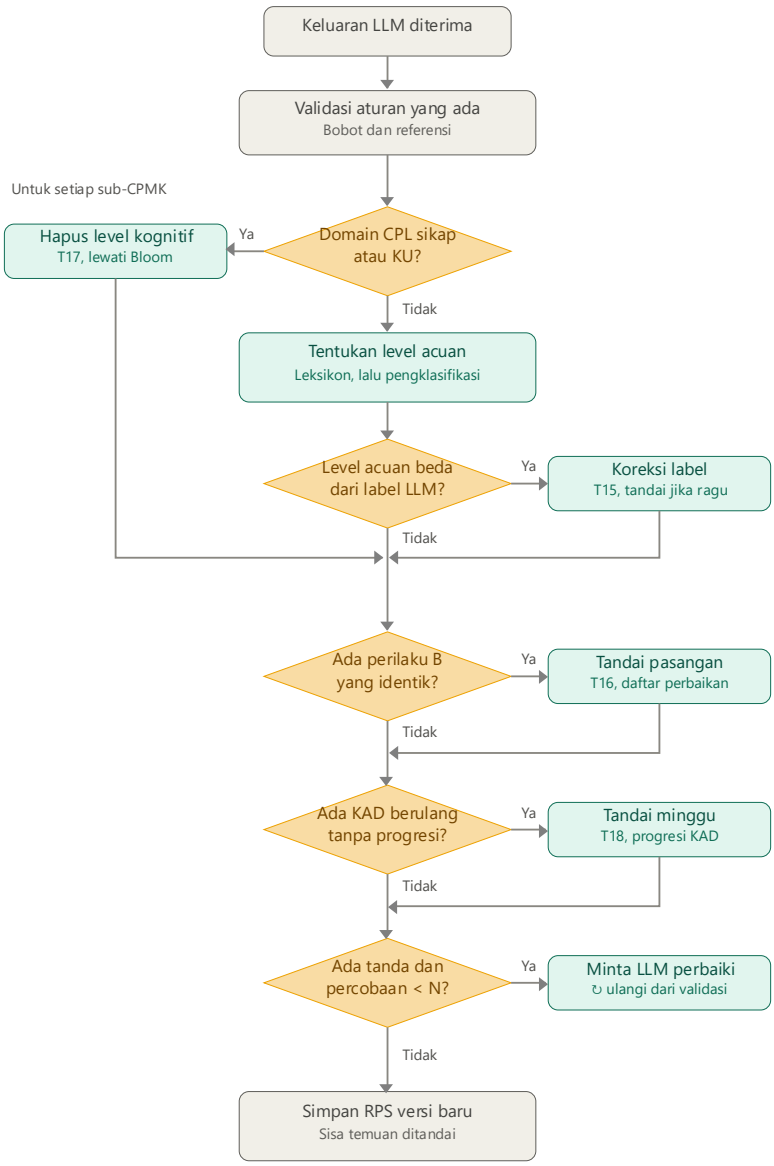
Hasil pengujian memperlihatkan jarak antara kesahihan struktur dan kualitas isi. Seluruh 42 hasil uji fungsional sesuai, tetapi delapan dari 12 hasil uji kualitas isi berupa temuan. Hal ini terjadi karena validasi aturan yang ada baru memeriksa total bobot dan keabsahan referensi sub-CPMK. Pemeriksaan tersebut bersifat struktural, sehingga RPS yang strukturnya benar tetap dapat memuat rumusan yang keliru secara makna. Pola ini sejalan dengan peringatan Kasneci dkk. [8] bahwa keluaran LLM memerlukan pengawasan manusia. Pada aplikasi ini, pengawasan tersebut baru diwujudkan melalui fitur penyuntingan RPS, sehingga beban pemeriksaan isi masih sepenuhnya berada pada dosen.

Kesalahan semantik taksonomi Bloom (T15) paling mendesak ditangani. Dari enam sub-CPMK yang tidak sesuai, lima berlabel lebih rendah daripada level kata kerjanya (“menganalisis” berlabel C2 atau C3, dan “mengevaluasi” berlabel C4), dan satu berlabel lebih tinggi (“menerapkan” berlabel C4). Pada sampel ini, label level tampak tidak terikat pada kata kerja yang ditulis. Usulan perbaikannya adalah validator semantik yang dijalankan setelah validasi aturan yang ada (Tabel 5). Lapis pertama berupa leksikon kata kerja operasional per level Bloom. Perilaku (B) setiap sub-CPMK dicocokkan dengan leksikon, lalu levelnya dibandingkan dengan label dari LLM. Bila berbeda, sistem mengoreksi label dan tidak mengubah kata kerja, karena kata kerja membawa makna capaian yang dimaksud dosen. Lapis kedua berupa pengklasifikasi level untuk kata kerja di luar leksikon. Almatrafi dan Johri [5] melaporkan bahwa BERT lebih akurat daripada strategi instruksi pada GPT-4 untuk tugas ini, sehingga pengklasifikasi terpisah lebih layak daripada meminta LLM memeriksa keluarannya sendiri. Hasil klasifikasi dengan keyakinan rendah tidak diubah otomatis, tetapi ditandai agar ditinjau dosen. Lapis ketiga berupa permintaan ulang terbatas, yaitu sub-CPMK yang ditandai dikembalikan ke LLM beserta pesan kesalahannya, dengan jumlah percobaan dibatasi.

Tabel 5. Usulan aturan validasi semantik berdasarkan temuan T15–T18

Temuan	Aturan usulan	Tindakan sistem
T15 – Level tidak sesuai kata kerja	Leksikon kata kerja operasional per level Bloom, dan pengklasifikasi level untuk kata kerja di luar leksikon	Koreksi label, atau tandai bagi dosen bila keyakinan rendah
T16 – Sub-CPMK kembar	Pencocokan perilaku (B) antarsub-CPMK setelah normalisasi teks	Minta LLM membedakan cakupan sub-CPMK yang ditandai
T17 – Capaian sikap berlevel kognitif	Pemeriksaan domain CPL (S dan KU) dan daftar kata kerja keterampilan lunak	Hapus level kognitif dan lewati pemeriksaan Bloom
T18 – Pengulangan KAD	Aturan progresi dan penyebaran kemunculan pertama KAD	Susun ulang minggu yang ditandai

Tiga temuan lain memiliki penanganan yang lebih sederhana. Untuk T17, domain CPL yang sudah dimasukkan dosen (P, KU, KK, S) dapat dipakai. Sub-CPMK yang dipetakan ke CPL berdomain sikap atau keterampilan umum tidak diberi level kognitif C1 sampai C6 dan dilewati pada pemeriksaan Bloom. Daftar kata kerja *soft-skill*, seperti berkolaborasi dan mengomunikasikan, menjadi pengaman tambahan bila pemetaan domain keliru. Langkah ini mengikuti pemisahan domain kognitif dan keterampilan lunak [6]. Untuk T16, kedua pasangan sub-CPMK kembar pada D2 memiliki frasa perilaku yang identik, sehingga pencocokan teks setelah normalisasi sudah cukup menangkapnya pada sampel ini. Ambang kemiripan untuk frasa yang hampir sama perlu dikalibrasi pada penelitian lanjutan. Untuk T18, diperlukan aturan progresi, yaitu setiap penempatan KAD setelah kemunculan pertamanya harus menaikkan level Bloom atau memperluas materi. Penempatan yang tidak memenuhinya ditandai sebagai pengulangan murni. Aturan penyebaran juga diperlukan, yaitu kemunculan pertama KAD tidak boleh seluruhnya terkumpul pada paruh pertama, seperti yang terjadi pada D1 dan D3. Berikut ini flowchart rancangan validator semantik yang dijalankan berdasarkan usulan pada tabel 5.



Gambar 8. Usulan alur validator semantik

4. KESIMPULAN

Aplikasi web penyusun RPS berbasis OBE dengan LLM telah dikembangkan dengan metode RAD dan dimodelkan dengan UML. Aplikasi menghasilkan CPMK, sub-CPMK berformat ABCD, rencana 16 pertemuan, dan sistem penilaian, menyediakan tiga model dengan lapisan pemeriksaan bertingkat, serta mengelola kuota dan ekspor Excel. Pengujian black box pada tiga RPS hasil generate menunjukkan 46 dari 54 hasil sesuai. Seluruh uji fungsional sesuai, sedangkan delapan temuan berada pada kualitas isi, yaitu ketidaksesuaian level Bloom, sub-CPMK kembar, capaian sikap yang berlevel kognitif, dan pengulangan KAD. Keterbatasan penelitian ini adalah pengujian hanya pada tiga RPS, bersifat observasional tanpa mengeksekusi fungsi berkuota dan unduh, serta tanpa penilaian pengguna.

Saran untuk penelitian selanjutnya mencakup empat hal. (1) Mengimplementasikan validator semantik pada Tabel 5, lalu mengulang kasus uji T15 sampai T18 pada tiga RPS yang sama dan RPS dari program studi lain, dengan membandingkan jumlah temuan sebelum dan sesudah validator. (2) Menyusun leksikon kata kerja operasional Bahasa Indonesia yang lebih lengkap per level Bloom dan menguji pengklasifikasi level terhadap sampel yang dilabeli pakar. (3) Menjalankan fungsi yang belum dieksekusi, yaitu generate ulang, pembuatan draf soal, dan unduh, serta membandingkan ketiga model pada data uji yang sama dengan model yang tercatat. (4) Melibatkan dosen dari beberapa program studi untuk mengukur efisiensi waktu penyusunan dan penerimaan aplikasi, disertai penilaian pakar atas mutu isi RPS

DAFTAR PUSTAKA

[1] E. D. Spangenberg, "Constructive alignment of a mathematics methodology module," *Instructional Science*, vol. 53, no. 4, pp. 729–760, 2025, doi: 10.1007/s11251-025-09712-z.

[2] M. I. Muzakir and Susanto, "Implementasi kurikulum Outcome Based Education (OBE) dalam sistem pendidikan tinggi di era Revolusi Industri 4.0," *Edukasiana: Journal of Islamic Education*, vol. 2, no. 1, pp. 118–139, 2023, doi: 10.61159/edukasiana.v2i1.86.

[3] E. Yusnaldi, Zunidar, N. Siregar, and A. Yumni, "Pengembangan perangkat pembelajaran berbasis Outcome Based Education pada Program Studi Pendidikan Guru Madrasah Ibtidaiyah," *Research and Development Journal of Education*, vol. 10, no. 2, pp. 1321–1328, 2024, doi: 10.30998/rdje.v10i2.25926.

[4] R. Rahmaniati, P. Rosawanti, Mariaty, A. Purnama, and Hariyadi, "Pelatihan pembuatan RPS berbasis OBE bagi dosen Fakultas Pertanian dan Kehutanan Universitas Muhammadiyah Palangkaraya," *PengabdianMu: Jurnal Ilmiah Pengabdian kepada Masyarakat*, vol. 9, no. 8, pp. 1458–1463, 2024, doi: 10.33084/pengabdianmu.v9i8.7541.

[5] O. Almatrafi and A. Johri, "Leveraging generative AI for course learning outcome categorization using Bloom's taxonomy," *Computers and Education: Artificial Intelligence*, vol. 8, Art. no. 100404, 2025, doi: 10.1016/j.caeai.2025.100404.

[6] D. Orih, M. Heyeres, R. Morgan, H. Udah, and K. Tsey, "A systematic review of soft skills interventions within curricula from school to university level," *Frontiers in Education*, vol. 9, Art. no. 1383297, 2024, doi: 10.3389/feduc.2024.1383297.

[7] W. X. Zhao *et al.*, "A survey of large language models," *Frontiers of Computer Science*, vol. 20, no. 12, Art. no. 2012627, 2026, doi: 10.1007/s11704-026-60308-3.

[8] E. Kasneci *et al.*, "ChatGPT for good? On opportunities and challenges of large language models for education," *Learning and Individual Differences*, vol. 103, Art. no. 102274, 2023, doi: 10.1016/j.lindif.2023.102274.

[9] P. Sridhar, A. Doyle, A. Agarwal, C. Bogart, J. Savelka, and M. Sakr, "Harnessing LLMs in curricular design: Using GPT-4 to support authoring of learning objectives," *arXiv preprint* arXiv:2306.17459, 2023, doi: 10.48550/arXiv.2306.17459.

[10] M. A. Mustapa, L. Salahuddin, and U. R. Hashim, "Automated study plan generator using rule-based and knapsack problem," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 8, Art. no. 47, 2022, doi: 10.14569/IJACSA.2022.0130847.

[11] R. Parlika, M. Afifudin, I. A. Pradana, Y. D. W. Wiratama, and M. N. Holis, "Literature study on the efficiency of the Rapid Application Development model in software development," *POSITIF: Jurnal Sistem dan Teknologi Informasi*, vol. 8, no. 2, pp. 64–73, 2022, doi: 10.31961/positif.v8i2.1329.

[12] A. Profita, A. N. Ifan, and A. E. Burhandenny, "Penerapan metode Rapid Application Development (RAD) untuk digitalisasi UKM industri busana muslim," *Jurnal Rekayasa Teknologi Informasi (JURTI)*, vol. 6, no. 2, pp. 171–178, 2022, doi: 10.30872/jurti.v6i2.8096.

[13] B. Alturas, "Connection between UML use case diagrams and UML class diagrams: A matrix proposal," *International Journal of Computer Applications in Technology*, vol. 72, no. 3, pp. 161–168, 2023, doi: 10.1504/IJCAT.2023.133294.

[14] P. A. D. A. Santi, R. Afwani, M. A. Albar, S. E. Anjarwani, and A. Z. Mardiansyah, "Black box testing with equivalence partitioning and boundary value analysis methods (study case: Academic Information System of Mataram University)," in *Proc. First Mandalika Int. Multi-Conf. Sci. Eng. 2022, MIMSE 2022 (Informatics and Computer Science)*, Adv. Comput. Sci. Res., vol. 102, 2022, pp. 207–219, doi: 10.2991/978-94-6463-084-8_19.

[15] A. Tyanafisya *et al.*, "Black box testing using equivalence partitioning technique on Bakkar website," *Jurnal INSTEK (Informatika Sains dan Teknologi)*, vol. 10, no. 1, pp. 230–238, 2025, doi: 10.24252/instek.v10i1.52951.

[16] Kementerian Pendidikan Tinggi, Sains, dan Teknologi, "Peraturan Menteri Pendidikan Tinggi, Sains, dan Teknologi Republik Indonesia Nomor 39 Tahun 2025 tentang Penjaminan Mutu Pendidikan Tinggi," Jakarta, 2025. [Daring]. Tersedia: https://peraturan.go.id/files/permendiktisaintek-no-39-tahun-2025.pdf