



Artikel Penelitian

Pembangunan Sistem Informasi Pengelolaan Rencana Pembelajaran Semester (RPS)

Aina Hubby Aziira ^{a,*}, Husnil Kamil ^b, Afriyanti Dwi Kartika ^c

^{a,b,c} Departemen Sistem Informasi, Fakultas Teknologi Informasi, Universitas Andalas, Limau Manis, Kec. Pauh, Kota Padang, Sumatera Barat 25175, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 02 Januari 2023

Revisi Akhir: 09 Juni 2023

Diterbitkan Online: 31 Agustus 2023

KATA KUNCI

Sistem Pengelolaan,

Pembelajaran,

RPS,

Sistem Informasi,

JavaScript.

KORESPONDENSI

E-mail: aina.hubbyaziira@it.unand.ac.id

A B S T R A C T

Rencana Pembelajaran Semester (RPS) merupakan wujud pengembangan kurikulum yang dibuat berdasarkan Dokumen Kurikulum. Pengelolaan RPS yang baik adalah salah satu kunci untuk memaksimalkan proses pengawasan dan pengembangan pembelajaran. Sistem pengelolaan RPS di Program Studi Sistem Informasi Universitas Andalas masih dilakukan secara konvensional. RPS belum terstruktur dengan baik sesuai standar penyusunan RPS sehingga akan menyulitkan dosen dalam melakukan pengawasan, evaluasi, dan pengembangan rencana pembelajaran, yang selanjutnya juga akan mempengaruhi kualitas lulusan serta akan menyulitkan proses penilaian pada saat Re-Akreditasi program studi. Oleh karena itu, pengelolaan RPS berupa penulisan yang terstruktur dan adanya wadah untuk mengarsipkan perubahan RPS yang dilakukan selama proses pengembangan, penting dilakukan. Hal ini dapat dicapai dengan menggunakan sebuah sistem informasi sehingga dirasa perlu untuk membangun sistem informasi pengelolaan RPS. Sistem Informasi Pengelolaan RPS dibangun berdasarkan metode *Waterfall* yang terdiri dari tahap analisis kebutuhan, perancangan sistem, implementasi sistem dengan memanfaatkan arsitektur MERN (MongoDB, Express.js, React.js, dan Node.js) sebagai empat teknologi utama, yang dilanjutkan dengan tahap pengujian. Dengan sistem yang dikembangkan, dosen dapat membuat RPS secara terstruktur serta memiliki arsip RPS yang pernah dikembangkan untuk nantinya dijadikan bahan evaluasi pembelajaran. Pengujian fungsional yang dilakukan menunjukkan bahwa sistem informasi pengelolaan RPS sudah berjalan sebagaimana yang direncanakan.

1. PENDAHULUAN

Kurikulum adalah bagian penting dalam proses pendidikan. Kurikulum dirancang dan dievaluasi secara berkala sebagai upaya peningkatan kualitas pendidikan serta pengembangan keunggulan kompetitif program studi dan lulusan. Rencana Pembelajaran Semester (RPS) merupakan wujud pengembangan kurikulum yang dibuat berdasarkan Dokumen Kurikulum.

Menurut Permenristekdikti No. 44 Tahun 2015, RPS hendaknya mengikat dosen dan mahasiswa ke dalam kontrak belajar/kuliah. RPS menjadi acuan dalam pelaksanaan proses pembelajaran sebuah mata kuliah dalam satu semester, sehingga perlu dilakukan pengawasan dan pemeriksaan terhadap kualitas RPS yang dibuat

oleh dosen. Perkuliahan yang dilaksanakan tanpa RPS bisa dikatakan merupakan perkuliahan yang tidak terencana.

Pengelolaan RPS yang baik adalah salah satu kunci untuk memaksimalkan proses pengawasan dan pengembangan pembelajaran. Oleh karena itu, pengelolaan RPS berupa penulisan yang terstruktur dan adanya wadah untuk mengarsipkan perubahan RPS yang dilakukan selama proses pengembangan, penting dilakukan.

Sistem pengelolaan RPS di Program Studi Sistem Informasi Universitas Andalas masih dilakukan secara konvensional. RPS belum terstruktur dengan baik sesuai standar penyusunan RPS sehingga menimbulkan beberapa permasalahan. Oleh karena RPS disusun dengan menggunakan aplikasi pengolah kata (*word processing*) sehingga tidak ada batasan (*constraint*) dan *layout*

yang bisa diterapkan pada aplikasi tersebut. Jika permasalahan ini tidak diselesaikan maka akan menyulitkan dosen dalam melakukan pengawasan, evaluasi, dan pengembangan rencana pembelajaran, yang selanjutnya juga akan mempengaruhi kualitas lulusan serta akan menyulitkan proses *assessment* pada saat Re-Akreditasi program studi.

Salah satu solusi dari permasalahan diatas adalah dengan memanfaatkan teknologi informasi dalam pengelolaan RPS. Penggunaan sistem informasi dapat mempermudah dosen dalam menyusun RPS sesuai ketentuan. Data RPS yang ada dalam sistem informasi juga dapat mempermudah akses mahasiswa terhadap RPS tersebut [1], [2]. Sistem Informasi juga dapat menyeragamkan RPS dan dapat dimanfaatkan lebih lanjut seperti untuk mendukung pencatatan kehadiran dosen [3]. Penggunaan sistem informasi juga akan mempermudah pihak terkait dalam membuat laporan rekapitulasi dan monitoring RPS [4]. Berdasarkan permasalahan di atas, maka perlu dilakukan penelitian untuk mengembangkan sistem informasi pengelolaan RPS. Akan tetapi, kemudahan dalam pengembangan sistem kedepannya untuk memenuhi kebutuhan yang cenderung terus bertambah, terkadang luput dari perhatian. Oleh karena itu, berbeda dengan penelitian sebelumnya, sistem pengelolaan RPS pada penelitian ini akan dirancang dengan memisahkan *back-end* dan *front-end* dalam proses pengembangan sistem. Selain untuk mengoptimalkan proses kerja sehingga menjadi lebih cepat dan tepat, hal ini juga dilakukan untuk memudahkan pengembangan sistem kedepannya. Sistem informasi ini akan memanfaatkan Node.js dengan Bahasa pemrograman JavaScript yang memiliki performa yang baik dalam pembuatan *website* serta memanfaatkan React.js untuk membuat User Interface (UI) *website* yang menarik. Penelitian ini juga memfokuskan pada keterkaitan setiap komponen dalam pengisian RPS sehingga diharapkan dapat menyelesaikan permasalahan yang ada di Program Studi Sistem Informasi terkait pengelolaan RPS.

Beberapa penelitian terkait sistem pengelolaan Rencana Pembelajaran Semester (RPS) telah dilakukan sebelumnya. Penelitian Budi dan Rika [1] serta penelitian yang dilakukan oleh Nada, Lillyan, dan Rampi [2], memanfaatkan *website* sebagai basis pengembangan sistem informasi pengelolaan RPS. Penelitian ini berhasil mengembangkan sebuah sistem informasi pengelolaan RPS yang dapat memudahkan dosen dalam mengisi data RPS dan menjadi wadah penyimpanan rekapitulasi RPS yang pernah dibuat. Gita dan Agus [4] juga melakukan penelitian untuk merancang aplikasi RPS sebagai upaya meningkatkan pencapaian pembelajaran bagi dosen. Penelitian ini [4] menyimpulkan bahwa pengembangan sistem terintegrasi sebagai pengganti sistem konvensional perlu dilakukan dan dipandang mampu meningkatkan pencapaian dosen.

Penelitian yang dilakukan oleh Badrudin dan Rayan Nurudin pada tahun 2019 dengan topik “SIM (Sistem Informasi Manajemen) Kurikulum Perguruan Tinggi Keagamaan Islam Berbasis CMS Wordpress” membahas sistem penyusunan Rencana Pembelajaran Semester pada Perguruan Tinggi Keagamaan Islam

atau PTKI yang masih bervariasi. Sering ditemukan perbedaan format dalam penyusunan RPS antar dosen, proses pelaksanaan juga terkesan memakan waktu yang lama, pendistribusian RPS yang telah disusun juga tidak merata sehingga informasi tidak mencapai seluruh mahasiswa. Penelitian ini menunjukkan bahwa sistem informasi yang diusulkan dapat meminimalisir perbedaan format RPS yang disusun oleh masing-masing dosen serta alur distribusi informasi mengenai RPS berjalan baik meski hanya menggunakan Wordpress sebagai sistem informasi manajemen [5].

Penelitian selanjutnya berjudul “Perencanaan Sistem Informasi Instrumen Kurikulum dan Evaluasi RPS Studi Kasus: Program Studi Sistem Informasi” yang dilakukan oleh Yohannes Accrus Alldy Indrawan pada tahun 2019. Penelitian ini membahas tentang proses penyusunan kurikulum pada Universitas Ma Chung di mana pada proses yang sedang berjalan memerlukan banyak data yang harus diproses namun belum terdapat sistem yang mengelola data tersebut sehingga proses manual yang berjalan terkesan memakan waktu. Masalah lain yang timbul adalah pendistribusian RPS yang hanya dilihat pada awal semester saja sementara mahasiswa tidak memiliki pegangan terhadap RPS yang telah disusun. Solusi yang ditawarkan dalam penelitian ini berupa sistem informasi yang mampu mengelola banyak data dalam penyusunan kurikulum sehingga proses berlangsung lebih efektif serta akses yang dimiliki mahasiswa untuk melihat RPS yang telah disusun oleh dosen. Hasil dari penelitian ini setelah diterapkannya sistem usulan membuat proses yang berjalan dalam kurikulum berlangsung pada Universitas Ma Chung lebih efektif dan efisien [6].

Pembuatan sistem informasi pengelolaan RPS pada penelitian-penelitian sebelumnya kebanyakan menggunakan bahasa pemrograman PHP dengan basis data MySQL [1], [2], [4]–[6]. Selain PHP yang merupakan bahasa sisi *server* murni, terdapat bahasa pemrograman lain seperti JavaScript yang dapat digunakan untuk pengembangan skala penuh.

Berbeda dengan penelitian sebelumnya, penelitian ini akan merancang sebuah sistem pengelolaan RPS berbasis *website* untuk mendukung pengelolaan RPS pada Program Studi Sistem Informasi Universitas Andalas dengan memanfaatkan Node.js dengan Bahasa pemrograman JavaScript yang memiliki performa yang baik dalam pembuatan *website* [7] serta memanfaatkan React.js untuk membuat *User Interface* (UI) *website* yang menarik. Penelitian ini juga memfokuskan pada keterkaitan setiap komponen dalam pengisian RPS seperti Capaian Pembelajaran Mata Kuliah (CPMK) untuk setiap Capaian Pembelajaran Lulusan (CPL), referensi, dan Komponen Penilaian yang digunakan dalam setiap Pertemuan. Pada penelitian ini, sistem informasi dibangun dengan memisahkan *back-end* dan *front-end*. Selain untuk mengoptimalkan proses kerja sehingga menjadi lebih cepat dan tepat, hal ini juga dilakukan untuk memudahkan pengembangan sistem kedepannya. Pengembangan dapat dilakukan dengan mengintegrasikan bagian *back-end* sistem

dengan aplikasi atau sistem informasi lainnya seperti sistem informasi monitoring perkuliahan berdasarkan RPS.

1.1. *Back-end*

Dalam proyek *web* Java tradisional, kode *front-end* dan *back-end* tercampur menjadi satu. Karena sumber daya statis dan sumber daya dinamis digabungkan bersama di *server*, tekanan *server* menjadi sangat besar. Jika beberapa masalah muncul di *server*, *front-end* dan *back-end* akan bermasalah pada saat yang bersamaan sehingga pengalaman pengguna akan menjadi sangat buruk. Oleh karena itu, Pemisahan *front-end* dan *back-end* telah menjadi arah dari pembangunan sistem atau aplikasi berbasis *web* [8].

Back-end dan *front-end* adalah istilah yang digunakan oleh pemrogram dan profesional komputer untuk mendeskripsikan lapisan yang membentuk program komputer atau situs web, yang digambarkan berdasarkan akses yang dimiliki oleh pengguna. Dalam konteks ini, pengguna mengacu pada entitas yang bisa berupa manusia atau digital.

Back-end mengacu pada bagian dari aplikasi komputer atau kode program yang memungkinkannya beroperasi dan tidak dapat diakses oleh pengguna. Sebagian besar data dan sintaks operasi disimpan dan diakses di *back-end* sistem komputer. *Back-end* juga disebut lapisan akses data perangkat lunak yang mencakup semua fungsi yang perlu diakses dan dinavigasi dengan cara digital. Aplikasi *back-end* dapat berinteraksi langsung dengan *front-end* atau dapat dipanggil dari program perantara yang memediasi aktivitas *front-end* dan *back-end*.

Industri pengembangan *web* dapat dikelompokkan menjadi dua jenis pengembang. Pengembang *front-end* dan Pengembang *back-end*. Pengembang *back-end* membangun logika bisnis di balik aplikasi *web*. Pekerjaan seperti menambahkan dan mengambil sorotan berita ke dan dari aplikasi web, atau mengirim email dari formulir berbasis web, atau mengautentikasi kredensial pengunjung adalah bagian tugas dari pengembang *back-end*. Pengembang *back-end* harus memiliki pengetahuan tentang basis data baik rasional maupun irasional dan perlu mengetahui bahasa pemrograman seperti seperti PHP, .NET, Java, dan lainnya [9].

1.2. *Front-end*

Lapisan di atas *back-end* adalah *front-end* dan mencakup semua perangkat lunak atau perangkat keras yang merupakan bagian dari antarmuka pengguna. Pengguna manusia atau digital berinteraksi langsung dengan berbagai aspek *front-end* suatu program, termasuk data yang dimasukkan pengguna, tombol, program, situs web, dan fitur lainnya. Sebagian besar fitur ini dirancang oleh para profesional pengalaman pengguna (*User Experience*) agar dapat diakses dan mudah digunakan.

Pengembang *front-end* harus memiliki pengetahuan tentang HTML, CSS, dan bahasa pemrograman untuk menambahkan efek dan lebih banyak lagi ke *front-end* seperti JavaScript [9].

1.3. *MongoDB*

MongoDB merupakan salah satu bagian dari basis data non-relasional (NoSQL) yang awalnya dikembangkan untuk mengatasi permasalahan performa pada *Relational Database Management System* (RDBMS) dalam memproses data tidak terstruktur seperti dokumen, data media sosial, email, dll. Basis data NoSQL memiliki kemampuan untuk mengolah data yang tidak terstruktur secara efisien [10].

MongoDB sebagai basis data *open source* merupakan salah satu basis data yang banyak digunakan untuk menangani data-data dalam jumlah besar dengan tetap menyediakan performa tinggi dan skalabilitas yang mudah. MongoDB menyimpan data dalam skrip yang fleksibel, seperti JSON. Model dokumen dipetakan ke objek dalam kode aplikasi sehingga mudah untuk digunakan [11]. Mehra dkk dalam penelitiannya terkait penggunaan MERN Stack dalam proses pembangunan *web* [11] menjabarkan karakteristik MongoDb sebagai berikut:

- Basis data dengan tujuan penggunaan yang umum
- Memiliki rancangan skema yang fleksibel
- Memiliki skalabilitas dan *load balancing*
- Kerangka Agregasi
- Memiliki Fitur keamanan
- JSON, MapReduce

1.4. *Node.js*

Node.js adalah bagian dari JavaScript pada sisi *server*. Pada mulanya, JavaScript yang sudah ada sejak awal pengembangan *world wide web* memegang peran integral dalam membuat *front-end* situs *web* menjadi interaktif. JavaScript selalu diyakini sebagai bahasa skrip sisi *client* dan tidak ada hubungannya dengan pemrograman sisi *server*. Namun, dengan berkembangnya JavaScript sisi *server* seperti Node.js, konsep tersebut berubah. Saat ini JavaScript tidak hanya menjadi bahasa pemrograman *client side scripting*, tetapi juga mampu mengeksekusi di sisi *server*. Node.js berbeda dengan JavaScript. Node.js dibangun di atas JavaScript sehingga menggunakan bahasa yang sama, sehingga mempermudah *web developer* untuk mengembangkan aplikasi *front-end* maupun *back-end* [9].

Node.js juga menyediakan banyak *library* atau modul JavaScript yang membantu menyederhanakan pengembangan aplikasi web. Node.js juga dapat mengeksekusi kode dengan sangat cepat karena berjalan pada V8 JavaScript Engine dari Google Chrome. Kelebihan lainnya dari Node.js adalah bersifat *asynchronous* dan *event-driven* sehingga memungkinkan untuk implementasi *server* yang dapat menangani banyak *request* pada waktu yang bersamaan.

1.5. *React.js*

React.js adalah *open source library* JavaScript yang dapat membuat proses pengembangan antarmuka pengguna yang interaktif menjadi lebih mudah. *Syntax* React.js ditulis secara deklaratif sehingga memudahkan untuk melakukan pemeriksaan

kesalahan pada kode. Penggunaan React.js juga memudahkan pengembang *web* untuk melakukan pengembangan ataupun *maintenance* ketika terjadi hal yang tidak sesuai, karena React.js dikembangkan dengan berbasis komponen sehingga pengembang dapat langsung memeriksa komponen yang bermasalah tanpa mengganggu kinerja komponen lainnya [12].

1.6. Express.js

Express.js adalah *framework* atau kerangka kerja yang dibangun di atas Node.js. *Framework* ini menyediakan *Application Programming Interface* (API) yang disederhanakan untuk beberapa fungsi inti Node.js. Express.js menyediakan banyak fungsionalitas tambahan melalui modul HTTP yang tidak perlu ditulis ulang dari awal untuk tugas-tugas umum dalam menangani permintaan, menentukan rute, atau mengedit aset statis.

2. METODE

Proses pengembangan sistem pada penelitian ini menggunakan model SDLC (*Software Development Life Cycle*) [13]. SDLC atau siklus hidup pengembangan perangkat lunak merupakan proses

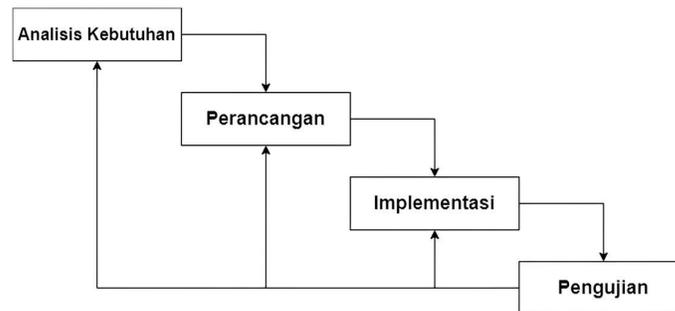
pembuatan metodologi yang digunakan untuk mengembangkan sebuah sistem. SDLC juga dapat didefinisikan sebagai pola untuk mengembangkan sistem perangkat lunak yang terdiri dari tahapan perencanaan (*planning*), analisis (*analyst*), perancangan (*design*), implementasi (*implementation*), uji coba (*testing*) dan pengelolaan (*maintenance*) [14].

Salah satu metode dari SDLC adalah Metode *Waterfall*. Metode ini dikenal dengan siklus hidup klasik yang menggambarkan pendekatan yang sistematis dan berurutan pada pengembangan perangkat lunak. Metode ini sering digunakan dalam pengembangan perangkat lunak. Pada metode ini, tahapan berikutnya tidak akan dilaksanakan sebelum tahapan sebelumnya selesai dilaksanakan [14].

Kelebihan dari metode ini diantaranya adalah:

- Dapat menjamin kualitas sistem yang dibangun karena dilaksanakan secara bertahap
- Metode ini juga dapat meminimalisir kesalahan yang mungkin terjadi

Model SDLC yang digunakan dalam penelitian ini adalah model *Waterfall*. Untuk lebih jelasnya dapat dilihat pada Gambar 1.



Gambar 1. Metode Pengembangan Sistem

2.1. Analisis Kebutuhan

Sebelum melakukan analisis, dilakukan pengumpulan data yang diperlukan terhadap kebutuhan sistem melalui kuesioner, wawancara, analisis dokumen dan studi literatur yang berkaitan dengan pembangunan sistem. Analisis difokuskan pada kebutuhan fungsi *software* untuk memenuhi kebutuhan pengguna.

2.2. Perancangan

Setelah analisis dilakukan, tahapan selanjutnya adalah mengubah kebutuhan-kebutuhan fungsi *software* tersebut menjadi sebuah bentuk rancangan sistem dan aliran proses dari sistem yang akan dirancang menggunakan diagram UML.

2.3. Implementasi

Sistem Pengelolaan RPS dibuat berbasis *website*. Sistem dibuat menggunakan bahasa pemrograman Typescript, yang merupakan bahasa pemrograman berbasis JavaScript. JavaScript adalah bahasa pemrograman yang digunakan dalam pengembangan *website* agar lebih dinamis dan interaktif.

Sistem dibuat dengan menggunakan *Integrated Development Environment* (IDE) Visual Studio Code sebagai software khusus yang menyediakan beragam fitur untuk mengembangkan sistem. Sistem juga menggunakan *library* React.js untuk membangun antarmuka pengguna yang menarik dan mudah dipahami. Penyimpanan data yang digunakan adalah MongoDB yang memiliki banyak keunggulan seperti performa yang cepat, pengelolaan basis data yang lebih mudah, dan kemampuan menampung banyak data bervariasi. Arsitektur serupa juga pernah digunakan untuk penelitian-penelitian sebelumnya terkait pembuatan sistem informasi dan menunjukkan performa yang bagus [15], [16].

2.4. Pengujian

Setelah implementasi selesai, tahapan selanjutnya adalah pengujian (*testing*) terhadap sistem yang dirancang untuk memastikan sistem telah memenuhi persyaratan dan menghasilkan keluaran yang sesuai dengan kebutuhan. Adapun

teknik pengujian yang digunakan adalah *black box testing* yang berfokus pada spesifikasi fungsional dari perangkat lunak [17]. Skenario pengujian dilakukan untuk menentukan langkah – langkah dalam melakukan pengujian. Pengujian dilakukan dengan menjalankan program atau aplikasi yang telah selesai dibuat. Skenario yang digunakan:

- Memasukkan data normal untuk setiap butir uji dan membandingkan hasil yang diharapkan dengan hasil yang diperoleh.
- Memasukkan data abnormal/ tidak sesuai format untuk setiap butir uji dan membandingkan hasil yang diharapkan dengan hasil yang diperoleh.
- Mengosongkan salah satu data dan membandingkan hasil yang diharapkan dengan hasil yang diperoleh.

Pengujian dilakukan oleh tim peneliti dan dosen Program Studi Sistem Informasi sebagai salah satu target pengguna sistem informasi.

3. HASIL

Sistem Informasi yang telah selesai dirancang merupakan sistem informasi berbasis *website*. Pada penelitian ini, fitur yang pertama kali dikembangkan adalah fitur untuk dosen dengan beberapa fungsi utama seperti login, melihat daftar RPS, menambah atau memperbaharui RPS, melihat detail RPS, menambah atau memperbaharui CPMK, Referensi, Komponen Penilaian, dan Pertemuan Mingguan.

1. Halaman *Login*

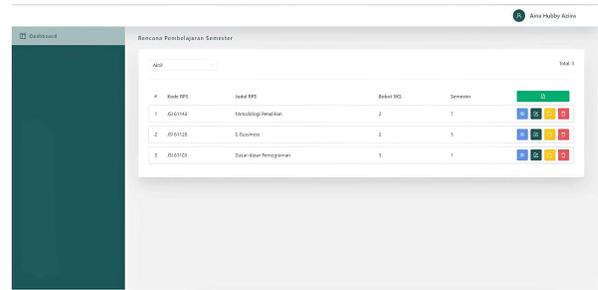
Halaman login merupakan halaman pertama yang akan berinteraksi dengan dosen saat membuka sistem pengelolaan RPS. Dosen akan memasukkan *username* dan *password* untuk selanjutnya masuk ke halaman List RPS. Halaman *Login* dapat dilihat pada Gambar 2.



Gambar 2. Halaman Login

2. Halaman *List RPS*

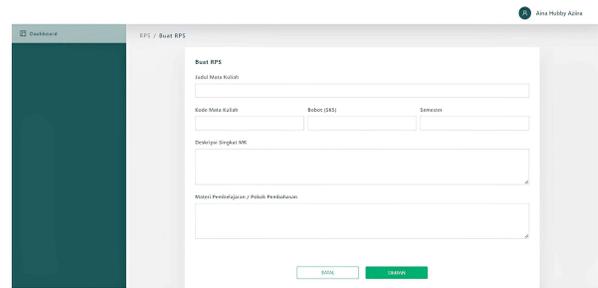
Halaman ini berisi daftar RPS yang sudah dibuat oleh dosen. Pada halaman ini, dosen bisa membuat RPS baru. Dosen juga dapat melihat, memperbaiki, atau menghapus RPS yang pernah ditambahkan sebelumnya. Halaman *List RPS* dapat dilihat pada Gambar 3.



Gambar 3. Halaman List RPS

3. Halaman *Tambah / Edit RPS*

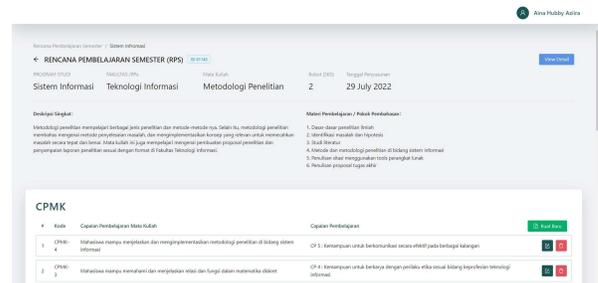
Digunakan untuk menambah atau memperbaharui data RPS, dapat diakses dengan menekan tombol tambah / edit di halaman List RPS. Halaman untuk menambah RPS dapat dilihat pada Gambar 4.



Gambar 4. Halaman *Tambah RPS*

4. Halaman *Detail RPS*

Halaman ini berisi *detail* dari setiap RPS yang dibuat. Untuk menuju halaman ini, pengguna harus menekan tombol Lihat (tombol berwarna biru) yang ada pada setiap RPS di halaman *List RPS* (Gambar 3). Pada Halaman ini, dosen bisa menambah, memperbaharui, atau menghapus komponen-komponen RPS yaitu CPMK, Referensi, Komponen Penilaian, dan Pertemuan Mingguan. Halaman *Detail RPS* dapat dilihat pada Gambar 5.

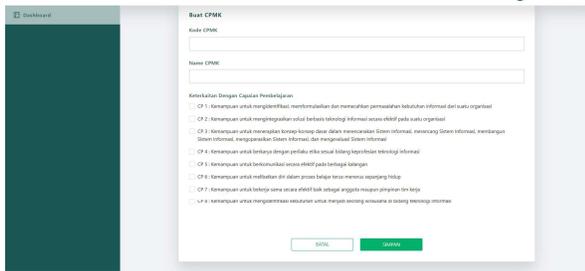


Gambar 5. Halaman *Detail RPS*

5. Halaman *Tambah / Edit CPMK*

Digunakan untuk menambah atau memperbaharui CPMK, dapat diakses dengan menekan tombol tambah / edit di halaman *Detail RPS* pada bagian CPMK. Setiap pembuatan CPMK baru, dosen harus memilih CP yang pencapaiannya didukung oleh CPMK

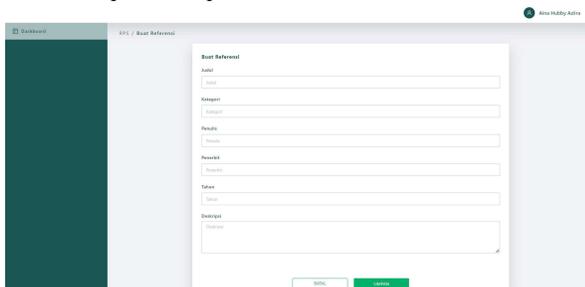
tersebut. Halaman untuk menambah CPMK dapat dilihat pada Gambar 6.



Gambar 6. Halaman Tambah CPMK

6. Halaman Tambah / Edit Referensi

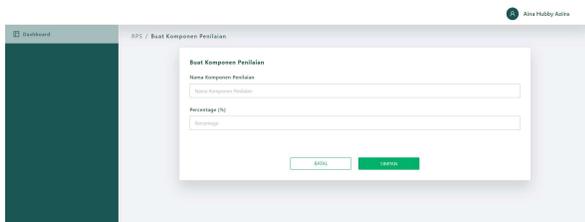
Digunakan untuk menambah atau memperbaharui Referensi, dapat diakses dengan menekan tombol tambah / edit di halaman *Detail* RPS pada bagian referensi. Halaman untuk menambah referensi dapat dilihat pada Gambar 7.



Gambar 7. Halaman Tambah Referensi

7. Halaman Tambah / Edit Komponen Penilaian

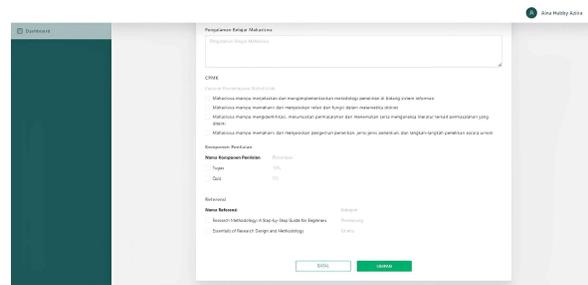
Digunakan untuk menambah atau memperbaharui Komponen Penilaian, dapat diakses dengan menekan tombol tambah / edit di halaman *Detail* RPS pada bagian Komponen Penilaian. Halaman untuk menambah Komponen Penilaian dapat dilihat pada Gambar 8.



Gambar 8. Halaman Tambah Komponen Penilaian

8. Halaman Tambah / Edit Pertemuan Mingguan

Digunakan untuk menambah atau memperbaharui data Pertemuan Mingguan, dapat diakses dengan menekan tombol tambah / edit di halaman *Detail* RPS pada bagian Pertemuan Mingguan. Pada halaman ini, Komponen CPMK, referensi, dan penilaian sudah otomatis ditambahkan berdasarkan pengisian sebelumnya yang dilakukan pada masing-masing komponen. Halaman untuk menambah Pertemuan Mingguan dapat dilihat pada Gambar 9.



Gambar 9. Halaman Tambah Pertemuan Mingguan

4. PEMBAHASAN

Pembuatan sistem informasi pengelolaan RPS pada penelitian-penelitian sebelumnya kebanyakan menggunakan bahasa pemrograman PHP dengan basis data MySQL [1], [2], [4]–[6]. Hasil dari penelitian-penelitian ini adalah sebuah sistem berbasis *website* yang dibuat dengan PHP dan memanfaatkan *Content Management System* (CMS) terkenal seperti Wordpress. Adanya penggabungan HTML dengan sintaksis pada struktur ini dapat menjadi salah satu faktor penghambat untuk memperluas sistem dengan fungsionalitas yang baru yang tentu saja harus menyesuaikan dengan perkembangan kurikulum dan kebutuhan program studi. Selain itu, penggunaan CMS juga memiliki masalah keamanan karena sangat rentan akan *cyber attack* dan akses ilegal.

Berbeda dengan penelitian sebelumnya, *website* yang dihasilkan pada penelitian ini dikembangkan dengan JavaScript sebagai bahasa pemrograman untuk mengembangkan *front end* dan *back end* yang terpisah. Dengan arsitektur yang terpisah, pengembangan sistem menjadi lebih mudah untuk dilakukan sehingga dapat menyesuaikan perubahan kebutuhan program studi.

Pemanfaatan React.js untuk membuat *User Interface* (UI) *website* juga dapat membuat halaman *website* menjadi dinamis dan dapat disesuaikan dengan kebutuhan pengguna sistem tanpa terlalu bergantung pada *template* yang tersedia seperti pada CMS. Selain untuk mengoptimalkan proses kerja sehingga menjadi lebih cepat dan tepat, pemanfaatan JavaScript sangat sesuai diterapkan pada sistem informasi yang akan selalu berkembang seperti sistem informasi pengelolaan RPS yang akan selalu menyesuaikan dengan perkembangan kurikulum dan kebutuhan program studi. Dari segi kewanaman, sistem yang dibuat tanpa menggunakan CMS akan cenderung lebih aman dari akses ilegal sehingga keamanan data juga dapat lebih terjaga.

4.1. Analisis Sistem

Berdasarkan hasil observasi dan wawancara yang telah dilakukan, diperoleh beberapa fakta: (1) Program Studi Sistem Informasi rutin melakukan evaluasi kurikulum sebagai upaya strategis untuk memperoleh keunggulan bersaing dan sebagai upaya perbaikan kualitas Pendidikan. Evaluasi secara menyeluruh terhadap kurikulum dilakukan setiap 5 (lima) tahun sekali. Sedangkan

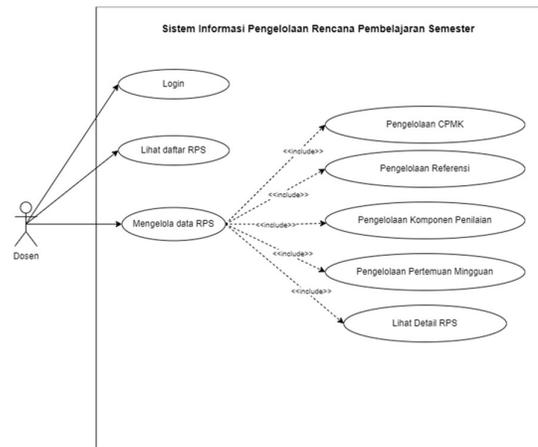
peninjauan secara berkala terhadap kurikulum dilakukan minimal sekali setahun berdasarkan kebutuhan program studi; (2) RPS disusun setiap awal semester dan harus memiliki struktur baku yang digunakan oleh semua dosen untuk mempermudah dalam evaluasi dan pengembangan; (3) RPS disusun dan disimpan oleh dosen yang tergabung dalam tim pengajar mata kuliah, tanpa sistem pengarsipan terpusat; (4) Salah satu indikator yang dijadikan penilaian pada saat reakreditasi program studi adalah RPS yang digunakan pada program studi. Berdasarkan fakta-fakta tersebut dapat disimpulkan bahwa pengelolaan RPS berupa penulisan yang terstruktur dan adanya wadah untuk mengarsipkan perubahan RPS yang dilakukan selama proses pengembangan, penting dilakukan.

Pengumpulan data tersebut menghasilkan beberapa kebutuhan untuk sistem informasi Rencana Pembelajaran Semester seperti:

- Pada penelitian ini, fitur yang pertama kali dikembangkan adalah fitur bagi dosen untuk melihat, menambah, memperbaiki, maupun menghapus RPS untuk setiap mata kuliah yang diampu dosen.
- Membangun suatu sistem yang dapat digunakan sebagai wadah pengarsipan RPS untuk mempermudah proses monitoring dan evaluasi pembelajaran.
- Membangun suatu sistem untuk mempermudah dosen agar dapat melakukan pengisian RPS secara *online*.

Kebutuhan fungsional sistem dimodelkan dengan menggunakan *Use Case Diagram* yang mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case Diagram* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi tersebut.

Sistem ini dirancang untuk dapat digunakan oleh tiga aktor, yaitu admin, dosen, dan mahasiswa. Akan tetapi, untuk pengembangan pertama, penelitian ini akan berfokus kepada fungsi yang dapat digunakan oleh dosen sebagai salah satu aktor. *Use Case Diagram* yang menunjukkan interaksi antara dosen dan sistem informasi pengelolaan RPS, dapat dilihat pada Gambar 10.



Gambar 10. *Use Case Diagram*

4.2. Rancangan Sistem

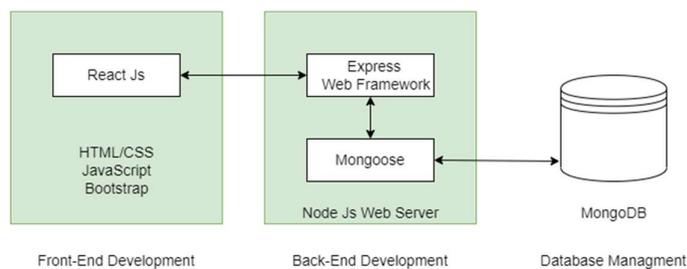
1. Arsitektur Umum Sistem Informasi

Arsitektur umum sistem informasi yang dikembangkan adalah dengan menggunakan *MERN Stack*. *MERN* adalah singkatan dari MongoDB, Express.js, React.js, Node.js, yang merupakan empat teknologi utama yang membentuk susunan sistem.

- MongoDB — basis data dokumen
- Express(.js) — kerangka kerja web Node.js
- React(.js) — *framework* JavaScript sisi *client*
- Node(.js) — *server web* JavaScript utama

MERN Stack adalah serangkaian teknologi serbaguna dan efisien yang digunakan untuk membangun aplikasi *web master* yang dapat diskalakan, termasuk elemen *front-end*, *back-end*, dan basis data (*database*). *MERN Stack* merupakan sistem berbasis JavaScript yang mudah digunakan untuk membuat situs web dan aplikasi interaktif [18].

Menggunakan arsitektur *MERN*, *front end*, *back end*, dan basis data dapat dibangun dengan mudah, dengan seluruhnya menggunakan JavaScript dan JSON. Arsitektur *MERN* yang digunakan pada sistem, dapat dilihat pada Gambar 11.



Gambar 11. Arsitektur *MERN* untuk Pembangunan Sistem Informasi Pengelolaan RPS

2. Basis Data (nonrelasional) NoSQL

Penelitian ini menggunakan basis data NoSQL berbasis dokumen. Basis data tipe ini tidak menyimpan data dalam bentuk tabel, akan

tetapi menggunakan dokumen terstruktur layaknya JSON (JavaScript Object Notation).

Tabel 1. Rancangan basis data

No	Collection	Field
1	RPS	<ul style="list-style-type: none"> • rps_id • rps_kode • rps_nama • rps_sks • rps_semester • rps_rev • rps_status • rps_deskripsi • rps_materi • rps_penulis
2	Pertemuan Mingguan	<ul style="list-style-type: none"> • session_id • session_rps_id • session_minggu • session_material • session_metode • session_pengalaman_mhs • session_penilaian • session_referensi • session_status
3	Penilaian	<ul style="list-style-type: none"> • penilaian_id • penilaian_rps_id • penilaian_nama • penilaian_persentase • penilaian_status
4	Referensi	<ul style="list-style-type: none"> • ref_id • ref_rps_id • ref_judul • ref_penulis • ref_penerbit • ref_tahun • ref_deskripsi • ref_kategori • ref_status
5	CP	<ul style="list-style-type: none"> • cp_id • cp_judul • cp_isi • cp_id_pembuat • cp_tipe • cp_status
6	CPMK	<ul style="list-style-type: none"> • cpmk_id • cpmk_rps_id • cpmk_kode • cpmk_nama • cpmk_cp_id • cpmk_status
7	Pengguna	<ul style="list-style-type: none"> • user_id • user_username • user_password • user_status • user_tipe

Walaupun bersifat dinamis, rancangan awal basis data terdiri dari tujuh *collection* atau yang lebih dikenal dengan tabel pada sistem basis data SQL. *Collection* tersebut terdiri dari *collection* RPS, Pertemuan Mingguan, Penilaian, Referensi, CP, CPMK, dan Pengguna. Setiap *collection* terdiri dari beberapa *field* atau yang lebih dikenal dengan kolom pada sistem basis data relasional. Daftar *collection* dan *field* yang akan digunakan pada sistem dapat dilihat pada Tabel 1.

4.3. Implementasi Sistem

Implementasi sistem dimulai dengan proses pembuatan *back-end*. *Back-end* dibuat dengan menggunakan bahasa pemrograman Node.js. Pada tahap ini, pembuatan fungsi untuk setiap fitur sistem dibuat pada bagian *controller*. Pada tahap ini juga, sistem dihubungkan dengan MongoDB sebagai basis data. Setelah proses pembuatan *back-end* selesai, *Application Programming Interface* (API) dari *back-end* akan dihubungkan ke bagian *front-end*, yang sebelumnya sudah ditambahkan *library* React.js sehingga dapat memaksimalkan proses pengembangan aplikasi web yang dinamis dan mempermudah penulisan kode JavaScript. Sistem diimplementasikan dengan membuat fitur-fitur utama untuk dosen. Setiap dosen akan mempunyai *username* dan *password* yang digunakan untuk *login*. Setelah *login*, dosen dapat melihat daftar RPS yang sudah pernah dibuat. Dosen dapat menambah atau memperbaharui RPS dengan menambah atau memperbaharui komponen-komponen RPS seperti CPMK, Referensi, Komponen Penilaian, dan Pertemuan Mingguan.

Tabel 2. Daftar komponen uji *black box testing*

No	Komponen Uji
1	Login
2	Logout
3	Button tambah RPS
4	Tambah RPS
5	Edit RPS
6	Arsip RPS
7	Hapus RPS
8	Button detail RPS
9	Klik button tambah CPMK
10	Tambah CPMK
11	Edit CPMK
12	Hapus CPMK
13	Tambah Referensi
14	Edit Referensi
15	Hapus Referensi
16	Tambah Komponen Penilaian
17	Edit Komponen Penilaian
18	Hapus Komponen Penilaian

19	Menu Pertemuan Mingguan
20	Tambah CPMK
21	Edit CPMK
22	Hapus CPMK
23	View RPS

4.4. Pengujian Sistem

Pengujian sistem informasi pada penelitian ini menggunakan metode *black box testing*. Pengujian *black box testing* digunakan untuk mengetahui keberhasilan modul sistem pada sisi fungsionalitas, khususnya pada bagian input data, apakah sudah sesuai dengan apa yang diharapkan atau belum. Selain itu, *black box testing* juga digunakan untuk menguji fungsi dari *button* dan *link* yang terdapat pada aplikasi. Skenario pengujian sebagaimana yang dijelaskan pada bagian 3.4 dilakukan pada 23 butir uji yang dapat dilihat pada Tabel 2.

Hasil yang diperoleh dari pengujian seluruh komponen uji menunjukkan bahwa sistem dapat berfungsi sesuai dengan kebutuhan. Pengujian juga dilakukan oleh sebagian dosen pada Program Studi Sistem Informasi dengan hasil yang menunjukkan bahwa sistem yang dibangun dapat memudahkan dosen dalam membuat RPS secara *online* dan membantu dosen melakukan pengawasan dan evaluasi pembelajaran, dengan memanfaatkan penyimpanan RPS yang terpusat dan lebih terstruktur.

5. KESIMPULAN

Dengan adanya sistem informasi pengelolaan rencana pembelajaran semester (RPS), Program Studi Sistem Informasi bisa memiliki sebuah sistem pengelolaan RPS yang sesuai dengan kebutuhan, dapat digunakan dalam proses pembelajaran maupun Re-Akreditasi, dan minim dari kesalahan yang mungkin dilakukan oleh penggunaannya. Dengan sistem yang dikembangkan, dosen dapat membuat RPS secara terstruktur serta memiliki arsip RPS yang pernah dikembangkan untuk nantinya dijadikan bahan evaluasi pembelajaran. Sistem yang dikembangkan juga dapat bermanfaat bagi Gugus Kendali Mutu (GKM) program studi dengan adanya pengarsipan dokumen RPS yang tersentralisasi. Dalam lingkup yang lebih luas, kepentingan organisasi untuk memiliki pengelolaan dokumen yang rapi dan terstruktur juga dapat dipenuhi.

DAFTAR PUSTAKA

- [1] B. Yanto and R. P. Sari, "Elektronik Pembelajaran Semester (E-RPS) Berbasis WEB Fakultas Ilmu Komputer Universitas Pasir Pengaraian," *Riau Journal of Computer Science*, vol. 05, no. 02, pp. 98–107, 2019, [Online]. Available: <http://e-journal.upp.ac.id/index.php/RJOCS/article/view/1753/1365>.
- [2] N. S. Adilah, L. Hadjaratie, and R. Yusuf, "Pengembangan Sistem Informasi Rencana Pembelajaran Semester dan Evaluasi Capaian Pembelajaran Lulusan Berbasis Progressive Web App," *Diffusion - Journal of System and Information Technology*, vol. 2, no. 2, pp. 84–96, 2022.
- [3] I. Sumartono and H. budi Santoso, "Analisis Perancangan Sistem Rencana Pembelajaran Terpadu dalam Mendukung Efektifitas dan Mutu Pengajaran Dosen," *Jurnal Teknik dan Informatika*, vol. 6, no. 1, pp. 12–17, 2019.
- [4] M. K. Gita Ayu Syafarina, M.Kom, Agus Setiawan, "Perancangan Aplikasi Rencana Pembelajaran Semester (RPS) untuk Meningkatkan Pencapaian Pembelajaran bagi Dosen," *Technologia*, vol. 10, no. 4, pp. 202–206, 2019.
- [5] R. Nurdin, "SIM (Sistem Informasi Manajemen) Kurikulum Perguruan Tinggi Keagamaan Islam Berbasis CMS Wordpress," *J. TA'DIB*, vol. 22, no. 1, pp. 2580–2771, 2019.
- [6] Y. A. A. Indrawan, "Perancangan Sistem Informasi Instrumen Kurikulum Dan Evaluasi Rps Studi Kasus: Program Studi Sistem Informasi," *Kurawal - J. Teknologi, Informasi dan Industri*, vol. 2, no. 1, pp. 12–22, 2019, doi: [10.33479/kurawal.2019.2.1.12-22](https://doi.org/10.33479/kurawal.2019.2.1.12-22).
- [7] P. Studi, M. Teknologi, and U. Amikom, "Perbandingan Performa Kinerja Node . js , PHP , dan Python dalam Aplikasi REST Performance Comparison of Node . js , PHP , and Python Performance," *Cogito Smart Journal*, vol. 4, no. 1, pp. 171–187, 2018.
- [8] Y. Gong, F. Gu, K. Chen, and F. Wang, "The Architecture of Micro-services and the Separation of Frond-end and Back-end Applied in a Campus Information System," in *IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, 2020.
- [9] H. Shah and T. R. Soomro, "Node. Js Challenges in Implementation," *Global Journals of Computer. Science and Technology: E Network, Web & Security*, vol. 17, no. 2, pp. 73–83, 2017.
- [10] M. Silalahi, "Perbandingan Performansi Database MongoDB Dan Mysql Dalam Aplikasi File Multimedia Berbasis Web," *Computer Based Information System Journal*, vol. 6, no. 1, p. 63, 2018, doi: [10.33884/cbis.v6i1.574](https://doi.org/10.33884/cbis.v6i1.574).
- [11] M. Mehra, M. Kumar, A. Maurya, and C. Sharma, "MERN stack web development," *Annals. R.S.C.B.*, vol. 25, no. 6, pp. 11756–11761, 2021.
- [12] Nasution and L. Iswari, "Penerapan React JS Pada Pengembangan FrontEnd," *J. Automata. - UII*, vol. 2, no. 2, pp. 193–200, 2021, [Online]. Available: https://drive.google.com/file/d/115_6camzfdvBC74On3IOQkGIC2nflA-3/view.
- [13] A. Sarkar, "Overview of Web Development Life cycle in Software Engineering," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* © 2018 IJSRCSEIT, vol. 3, no. 6, pp. 2456–3307, 2018.
- [14] A. A. Wahid, "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi," *Jurnal Ilmu-Ilmu Informatika dan Manajemen STMIK*, pp. 1–5, 2020.
- [15] L. O. Badru, V. Vasudevan, G. I. Lingam, and M. G. M. Khan, "MERN Stack Web-Based Education Management Information Systems for Pacific Island Countries," *SN Computer Science*, vol. 4, no. 1, 2022, doi: [10.1007/s42979-022-01457-7](https://doi.org/10.1007/s42979-022-01457-7).

- [16] J. P. Laudon, Kenneth C., Laudon, "Management Information System," *IJRESM - International Journal of Research in Engineering, Science and Management*, vol. 4, no. 5, pp. 144–146, 2021.
- [17] H. Yulianton, A. Trisetyarso, W. Suparta, B. S. Abbas, and C. H. Kang, "Web Application Vulnerability Detection Using Taint Analysis and Black-box Testing," *IOP Conference Series: Materials Science and Engineering*, vol. 879, no. 1, pp. 7–12, 2020, doi: 10.1088/1757-899X/879/1/012031.
- [18] "MERN Stack Explained," *MongoDB*, 2022. <https://www.mongodb.com/mern-stack>.