



PERANCANGAN SISTEM INFORMASI AKADEMIK PESANTREN BERBASIS WEB MENGGUNAKAN METODE *RESEARCH AND DEVELOPMENT*

Sukma Hidayatullah¹, Lukman Rosyidi²

^{1,2} Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
hidayatullahsukma@gmail.com, lukman@nurulfikri.ac.id

Abstract

Pesantren Teknologi Informasi dan Komunikasi (PeTIK) is a technology-based educational institution that faces challenges in managing academic data due to its manual system using Google Spreadsheets. This study aims to design and develop a web-based academic information system to enhance efficiency and accuracy in recording grades, attendance, and lecture schedules. The Research and Development (R&D) method with a qualitative approach was employed, involving admins, teachers, and students as research subjects. Data were collected through literature studies, interviews, observations, and documentation, and analyzed descriptively. The system was developed using Node.js and Express.js for the backend, React.js for the frontend, and MySQL as the database. Testing was conducted using black-box methods and user evaluations. The results show that the system improved work efficiency, indicated by reduced time in academic data recording, fewer input errors, and a streamlined academic workflow. The system also enhanced accessibility and transparency of academic information. In conclusion, the academic information system successfully replaced manual processes with integrated, efficient, and user-friendly digital workflows. Further development is recommended to include notification features, report generation, and automated schedule-change management to support more optimal academic operations.

Keywords: Academic Information System, JavaScript, Node.js, React.js, Web-based Application

Abstrak

Pesantren Teknologi Informasi dan Komunikasi (PeTIK) merupakan lembaga pendidikan berbasis teknologi yang mengalami kendala dalam pengelolaan data akademik akibat penggunaan sistem manual berbasis *Google Spreadsheet*. Penelitian ini bertujuan merancang dan membangun sistem informasi akademik berbasis web guna meningkatkan efisiensi dan akurasi dalam pencatatan nilai, kehadiran, dan penjadwalan perkuliahan. Penelitian ini menggunakan metode *Research and Development* (R&D) dengan pendekatan kualitatif. Subjek penelitian melibatkan admin, pengajar, dan santri. Data dikumpulkan melalui studi literatur, wawancara, observasi, serta dokumentasi, dan dianalisis secara deskriptif. Implementasi sistem dilakukan menggunakan Node.js dan Express.js untuk *backend*, React.js untuk *frontend*, dan MySQL sebagai basis data. Pengujian dilakukan dengan metode *black box testing* dan evaluasi pengguna. Hasil penelitian menunjukkan bahwa sistem mampu meningkatkan efisiensi kerja, ditunjukkan melalui pengurangan waktu pencatatan data akademik, penurunan kesalahan *input*, dan penyederhanaan alur kerja akademik. Sistem juga terbukti meningkatkan aksesibilitas dan transparansi data akademik. Kesimpulannya, sistem informasi akademik ini berhasil menggantikan metode manual dengan proses digital yang terintegrasi, efisien, dan mudah digunakan. Disarankan pengembangan lebih lanjut mencakup fitur notifikasi, cetak laporan, dan manajemen ubah jadwal otomatis untuk mendukung operasional akademik yang lebih optimal.

Kata kunci: Aplikasi berbasis Web, JavaScript, Node.js, React.js, Sistem Informasi Akademik

1. PENDAHULUAN

Di era yang serba digital ini khususnya setelah terjadinya pandemi *covid-19*, lembaga pendidikan perlu melakukan digitalisasi untuk meningkatkan akurasi serta efisiensi dalam komunikasi dan administrasi akademik [1]. Namun, demikian, tidak semua lembaga pendidikan melakukan hal

demikian seperti halnya Pesantren Teknologi Informasi dan Komunikasi (PeTIK) Depok.

PeTIK merupakan lembaga pendidikan berbasis teknologi yang mengajarkan ilmu IT dibarengi dengan nilai keislaman. Tetapi, proses pencatatan, pengelolaan dan

penyimpanan data akademik di PeTIK masih menggunakan Google Spreadsheet yang masih kurang efisien dan rentan terjadi kesalahan pencatatan data, seperti dalam pembuatan jadwal yang masih manual sehingga rentan terjadinya bentrok jadwal, pencatatan nilai dengan membuat format penilaian serta rumus sendiri dan pencatatan data akademik santri. Pengajar perlu membuat berkas, format penilaian serta presensi kehadiran sebelum melakukan pencatatan nilai maupun kehadiran sehingga prosesnya dinilai tidak efisien.

Hal ini karena pada awalnya PeTIK hanya memiliki santri sedikit karena itu masih mudah dalam melakukan pencatatan akademiknya. Namun, saat ini PeTIK menerima santri lebih banyak sehingga membuat proses pencatatan akademik menjadi lebih rumit karena jumlah santri yang bertambah. Untuk mengatasi masalah ini, diperlukan sebuah sistem informasi yang mampu melakukan pencatatan akademik secara efisien dan akurat [2].

Rumusan masalah yang diangkat pada penelitian ini adalah bagaimana merancang *website* sistem informasi akademik yang dapat menawarkan solusi dari permasalahan yang telah disebutkan tadi dan dapat memenuhi kebutuhan pendidikan di PeTIK serta bagaimana tingkat usability sistem informasi akademik berdasarkan pengujian *black box* yang dilakukan oleh pengguna untuk memastikan fungsionalitas dan kemudahan pengguna dalam menggunakan sistem informasi akademik.

Manfaat yang didapatkan dari penelitian ini bagi pesantren PeTIK adalah mempermudah admin dalam membuat jadwal perkuliahan IT di pesantren PeTIK dengan fitur yang meminimalisir terjadinya bentrok jadwal, mempermudah pengajar dalam melakukan pencatatan nilai santri sehingga pengajar tidak perlu membuat format penilaian dan rumusnya sendiri. Selain itu juga mempermudah santri dalam monitoring data kehadiran dan data nilainya sendiri secara *real time* dan mempermudah melakukan pencatatan akademik lainnya kepada pengajar seperti manajemen data pengajar, santri, mata kuliah dan kehadiran.

Penelitian ini didukung oleh studi sebelumnya yang menunjukkan keberhasilan perancangan sistem informasi akademik berbasis web di berbagai lembaga pendidikan. Misalnya penelitian yang dilakukan di MTs. Arrahmah yang menggunakan HTML, CSS dan PHP menunjukkan bahwa sistem informasi akademik memudahkan admin, wali kelas/guru dalam mengolah data akademik dan nilai-nilai siswa-siswi yang tadinya sistem penilaian dilakukan manual menggunakan aplikasi Office sederhana [3]. Penelitian lain di sekolah SMP Rahmat Islamiyah yang dirancang menggunakan HTML, CSS dan PHP menunjukkan bahwa sistem informasi akademik membantu mempercepat proses pencatatan data akademik [4]. Studi di Pondok Pesantren Al Halim Garut menggunakan metodologi *Rational Unified Process* (RUP) menunjukkan bahwa sistem informasi akademik mampu menjadi penghubung antara orang tua

santri dengan pengurus pondok pesantren tanpa harus datang langsung ke pondok pesantren selain itu sistem informasi akademik juga memudahkan dalam melakukan monitoring data akademik seperti nilai santri, presensi kehadiran santri dan jadwal [5].

2. METODE PENELITIAN

Bagian ini menjelaskan tentang jenis penelitian seperti rancangan penelitian dan tahapan penelitian yang dilakukan.

2.1 Metode Pengumpulan Data, Instrumen Penelitian, dan Metode Pengujian

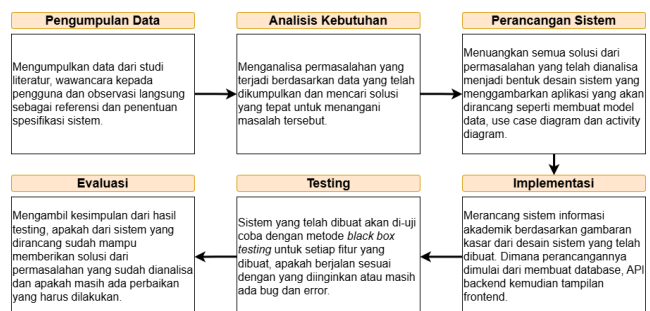
Penelitian ini mengadopsi metode *Research and Development* (R&D) yang bertujuan untuk merancang sistem informasi akademik berbasis web untuk Pesantren Teknologi Informasi dan Komunikasi (PeTIK) Depok. Dalam melakukan analisis data, penelitian ini menggunakan metode kualitatif. Metode kualitatif digunakan untuk memahami kebutuhan pengguna dan permasalahan yang dihadapi berdasarkan hasil wawancara kepada pengguna dan observasi langsung [6]. Data yang terkumpul, diorganisir dan dianalisis untuk mengidentifikasi pola dan tema yang relevan dengan spesifikasi sistem informasi akademik yang akan dirancang.

Teknik pengumpulan data dilakukan melalui wawancara kepada perwakilan admin, pengajar dan santri Pesantren Teknologi Informasi dan Komunikasi (PeTIK) Depok untuk mendapatkan informasi langsung proses pembuatan kurikulum dan jadwal perkuliahan, proses pencatatan akademik seperti pencatatan data pengajar, santri, presensi kehadiran dan pencatatan nilai. Selain itu, analisis berkas yang digunakan untuk pencatatan akademik di PeTIK juga dilakukan untuk mengetahui format yang biasa digunakan pada pencatatan akademik.

Pengujian sistem dilakukan melalui pendekatan *black box testing* yang dilakukan oleh perwakilan dari pengguna. *Black box testing* menguji setiap fitur sistem apakah berjalan sesuai fungsinya tanpa melihat kodenya [7]. Setelah pengguna melakukan pengujian, pengguna dimintai *feedback* terkait sistem yang dirancang.

2.2 Tahapan Penelitian

Pada bagian ini menjelaskan tahapan-tahapan penelitian dan hasil dari tahapan penelitian tersebut.



Gambar 1. Database Mirroring Architecture

Pada Gambar 1 tahapan-tahapan penelitian adalah sebagai berikut:

1. Pengumpulan Data

Tahap awal bertujuan menghimpun informasi dasar tentang kebutuhan Sistem Informasi Akademik Pesantren PeTIK. Peneliti menerapkan (1) studi literatur menelaah karya ilmiah, buku, serta standar pengelolaan data akademik yang relevan; (2) wawancara mendalam dengan calon pengguna untuk menggali alur kerja akademik dan spesifikasi sistem yang diinginkan; serta (3) observasi langsung atas praktik pencatatan nilai, kehadiran, dan jadwal perkuliahan melalui dokumen arsip pesantren.

2. Analisis Kebutuhan

Data yang terkumpul dianalisis guna memetakan persoalan inti, seperti kesulitan pengajar dalam pengelolaan nilai dan kendala admin saat menyusun jadwal kuliah yang kerap bentrok. Berdasarkan temuan tersebut, dirumuskan fitur-fitur solusi misalnya modul pencatatan akademik terintegrasi dan *generator* jadwal otomatis. Pada tahap ini pula ditetapkan tumpukan teknologi: Node.js sebagai *server-side runtime*, MySQL sebagai basis data, dan React JS untuk antarmuka pengguna.

3. Perancangan Sistem

Tahap ini berfokus pada penyusunan rancangan sistem berdasarkan hasil analisis kebutuhan sebelumnya. Proses dimulai dengan membuat model data, yaitu menentukan tabel-tabel utama beserta hubungan antar tabel yang akan menjadi dasar proses bisnis di dalam aplikasi. Setelah itu, disusun *use case diagram* untuk menggambarkan skenario penggunaan sistem oleh berbagai jenis pengguna [8]. Sebagai pelengkap, dibuat pula *activity diagram* yang menggambarkan alur kerja utama dalam sistem, seperti proses pembuatan jadwal, pengisian nilai, dan pencatatan kehadiran.

4. Implementasi

Tahapan implementasi merupakan proses menerjemahkan rancangan sistem menjadi aplikasi yang dapat dijalankan. Pembuatan sistem dimulai dari membangun *database* menggunakan MySQL sesuai dengan struktur data yang telah dirancang. Kemudian, API *backend* dikembangkan menggunakan Node.js dan Express.js, sedangkan antarmuka pengguna (*frontend*) dibangun dengan React.Js. Setelah keduanya selesai, dilakukan proses integrasi antara *frontend* dan *backend* dengan bantuan Axios, sehingga data dapat ditampilkan dan dikelola secara dinamis melalui antarmuka aplikasi.

5. Testing

Tahap pengujian dilakukan untuk memastikan bahwa setiap fitur dalam sistem berfungsi sebagaimana mestinya. Metode yang digunakan adalah *black box testing*, yaitu pengujian yang berfokus pada hasil keluaran dari setiap fungsi tanpa

melihat struktur internal kode. Selain itu, sistem juga diuji langsung oleh beberapa pengguna yang mewakili pihak terkait, seperti tenaga pengajar dan admin akademik. Umpan balik dari pengguna ini menjadi dasar untuk memperbaiki kekurangan atau kesalahan yang masih ditemukan sebelum sistem diterapkan secara penuh.

6. Evaluasi

Setelah pengujian selesai, dilakukan evaluasi menyeluruh untuk melihat apakah sistem benar-benar menjawab kebutuhan akademik di Pesantren PeTIK. Evaluasi ini mencakup peninjauan terhadap kelancaran fungsi sistem, kemudahan dalam penggunaan, serta efisiensi alur kerja yang ditawarkan. Saran dan masukan dari pengguna dikumpulkan dan dijadikan bahan pertimbangan untuk menyempurnakan sistem, agar lebih siap digunakan dalam mendukung proses akademik secara efektif dan berkelanjutan.

3. HASIL DAN PEMBAHASAN

Pada bagian ini membahas implementasi sistem dari perancangan yang sudah dibuat mulai dari pembuatan rancangan sistem berupa *model data*, *use case*, *database*, API *backend*, tampilan *frontend* aplikasi web yang responsif sampai pengujian sistem yang dibuat.

3.1 Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan melalui tiga metode, yaitu studi literatur, wawancara, dan observasi langsung.

Pertama, hasil studi literatur menunjukkan bahwa sistem informasi akademik merupakan sistem terintegrasi yang dirancang untuk mendukung pengelolaan data akademik secara lebih efisien dan akurat. Sistem ini umumnya mencakup pengelolaan data santri, jadwal perkuliahan, kehadiran, dan penilaian. Selain itu, sistem yang baik perlu didukung oleh antarmuka yang mudah digunakan, serta kemampuan akses bagi berbagai pihak yang terlibat, seperti admin, pengajar, dan peserta didik. Temuan ini menjadi dasar dalam merancang fitur dan struktur sistem yang akan dikembangkan.

Kedua, wawancara dilakukan dengan salah satu pengajar di Pesantren PeTIK yang juga bertanggung jawab dalam pengelolaan jadwal akademik.

Tabel 1. Hasil Wawancara dengan Pengajar

No	Pertanyaan	Jawaban
1	Apa kendala yang sering bapak alami dalam melakukan pencatatan akademik?	File yang tidak terorganisasi.
2	Bagaimana pesantren PeTIK melakukan pengelolaan akademiknya? seperti pencatatan data santri, data nilai dan data jadwal dll	Pencatatan akademik menggunakan <i>Google Drive</i> .

No	Pertanyaan	Jawaban
3	Bagaimana bapak membuat jadwal perkuliahan PeTIK?	Secara manual menggunakan Google Spreadsheet.
4	Apa kendala yang bapak alami dalam pembuatan jadwal?	Karena masih dibuat secara manual, terkadang terdapat beberapa jadwal yang bentrok dengan jadwal lain.
5	Kalau untuk pencatatan nilainya itu bagaimana?	Pencatatan nilai dibuat manual juga oleh setiap pengajar tapi dengan format yang sudah ditentukan tinggal pengajar yang menyesuaikan tergantung mata kuliahnya.
6	Kendala apa yang dialami saat pencatatan nilai?	Karena harus dibuat manual jadi kurang efisien apalagi kalau pertemuannya banyak dan salah rumus.
7	Kendala apa yang dialami saat pencatatan presensi kehadiran santri?	Kalau presensi dibuat di tempat terpisah yaitu di <i>e-learning</i> PeTIK jadi kalau mau rekap harus diambil dari <i>e-learning</i> kemudian dibuat lagi di Google Drive jadi belum terintegrasi dengan sistem pencatatan utama.

Berdasarkan Tabel 1 hasil wawancara, diperoleh informasi bahwa proses pencatatan akademik saat ini masih dilakukan secara manual menggunakan Google Drive dan Spreadsheet, sehingga berpotensi menimbulkan kesalahan, khususnya dalam penjadwalan yang kerap mengalami benturan antar kelas atau pengajar. Selain itu, pencatatan nilai juga dilakukan secara terpisah oleh masing-masing pengajar, dengan format yang tidak seragam. Untuk data kehadiran, pencatatan masih dilakukan melalui sistem *e-learning* yang terpisah dari sistem akademik utama, sehingga proses rekapitulasi menjadi kurang efisien.

Ketiga, observasi terhadap struktur berkas di Google Drive menunjukkan bahwa setiap angkatan memiliki sistem penyimpanan yang berbeda-beda. Data akademik seperti nilai, jadwal, silabus, dan dokumen santri disimpan dalam struktur folder yang tidak seragam, yang menyebabkan sulitnya pencarian data dan berpotensi menimbulkan duplikasi atau kehilangan informasi. Misalnya, pada angkatan XI, beberapa data dicampur dalam satu folder, sementara pada angkatan XII telah terdapat pemisahan yang lebih rapi berdasarkan jenis dokumen seperti RPP, silabus, dan laporan proyek.

Berdasarkan hasil pengumpulan data ini, dapat disimpulkan bahwa sistem informasi akademik yang dibutuhkan adalah sistem yang mampu menyederhanakan proses penjadwalan, menyatukan pencatatan nilai dan kehadiran dalam satu platform, serta menyediakan manajemen dokumen yang terstruktur dan mudah diakses. Temuan ini menjadi pijakan utama dalam tahap perancangan dan pengembangan sistem.

3.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan berdasarkan hasil wawancara, observasi langsung, dan dokumentasi akademik yang diperoleh dari Pesantren PeTIK. Tujuan dari tahap ini adalah untuk mengidentifikasi kebutuhan sistem secara menyeluruh, baik dari sisi pengguna maupun teknis, serta memahami kendala-kendala yang selama ini dihadapi dalam proses pengelolaan akademik.

Berdasarkan data yang terkumpul, ditemukan sejumlah permasalahan yang memerlukan solusi sistematis melalui perancangan sistem informasi akademik. Ringkasan kondisi saat ini dan kebutuhan yang muncul disajikan pada Tabel 2 berikut.

Tabel 2. Hasil Analisis Wawancara

No	Kondisi Sekarang	Kebutuhan yang ditemukan
1	Pencatatan akademik menggunakan <i>Google Drive</i> , tidak terorganisasi dengan baik	Sistem manajemen terpusat untuk seluruh data akademik yang rapi, terstruktur, dan mudah diakses
2	Pembuatan jadwal dilakukan secara manual di <i>Google Spreadsheet</i>	Sistem pembuatan jadwal yang otomatis dan terintegrasi
3	Jadwal sering mengalami bentrok antar kelas/pengajar	Fitur validasi bentrok jadwal otomatis saat pembuatan jadwal
4	<i>Folder</i> dan struktur <i>file</i> berbeda-beda setiap angkatan, menyulitkan pencarian data	Sistem penyimpanan data akademik dengan struktur yang konsisten antar angkatan
5	Data nilai dibuat manual oleh pengajar, menggunakan format masing-masing	<i>Form input</i> nilai terstandarisasi, dengan perhitungan otomatis
6	Sering terjadi kesalahan rumus dalam pengolahan nilai di <i>spreadsheet</i>	Fitur perhitungan nilai otomatis (termasuk rata-rata, nilai huruf, dan peringkat)
7	Data kehadiran santri tercatat di <i>e-learning</i> terpisah, rekap harus dibuat manual di <i>Google Drive</i>	Sistem kehadiran yang terintegrasi dengan sistem utama, dan otomatis menghitung persentase kehadiran
8	Rekap data akademik (nilai, kehadiran, jadwal) harus dicari manual dari berbagai <i>folder</i>	<i>Dashboard</i> akademik yang menampilkan data secara <i>real-time</i> berdasarkan peran pengguna

Dari Tabel 2, dapat disimpulkan bahwa sistem informasi akademik yang akan dikembangkan harus menjawab delapan permasalahan utama yang meliputi: pengelolaan data terpusat, otomasi penjadwalan, validasi bentrok jadwal, standarisasi dokumen dan *input* nilai, perhitungan otomatis nilai dan kehadiran, serta integrasi data ke dalam satu dasbor akademik yang dapat diakses secara *real time*.

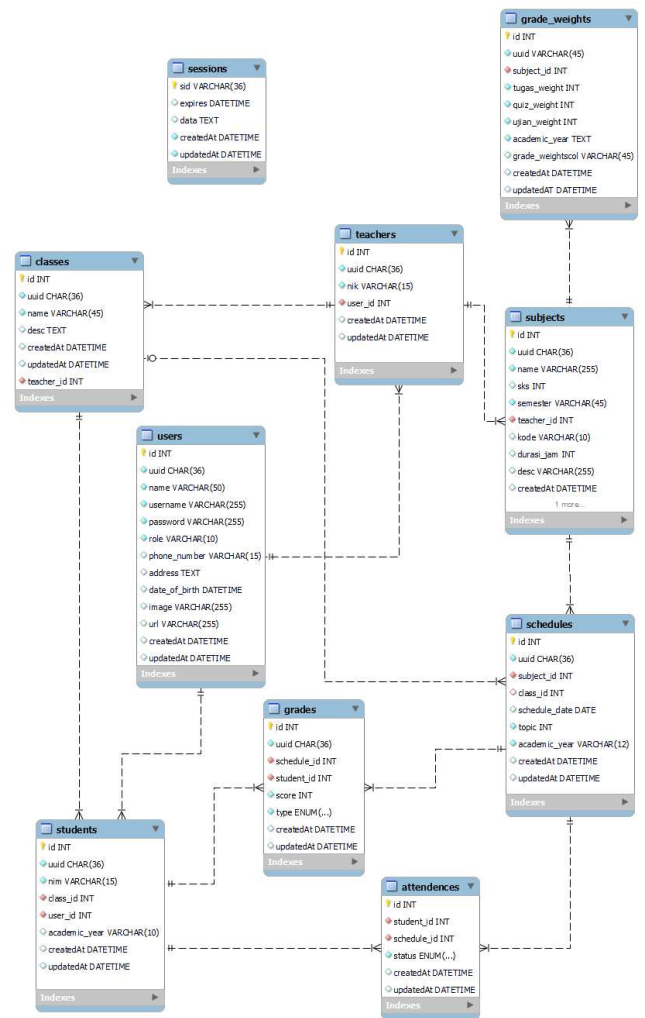
Untuk mendukung hal tersebut, sistem perlu mengakomodasi fitur akses berbasis peran (*role-based*

access control), yang memungkinkan pengguna seperti admin, pengajar, dan santri mengakses data dan fitur sesuai dengan hak akses masing-masing. Dengan demikian, kerahasiaan dan integritas data akademik dapat terjaga dengan baik.

3.3 Perancangan Sistem

Setelah tahap analisis kebutuhan selesai dilakukan, proses selanjutnya adalah perancangan sistem. Tahapan ini bertujuan untuk memvisualisasikan struktur sistem secara menyeluruh, sehingga proses implementasi dapat berjalan secara sistematis dan sesuai dengan kebutuhan pengguna. Perancangan sistem mencakup tiga aspek utama: model data, use case diagram, dan activity diagram.

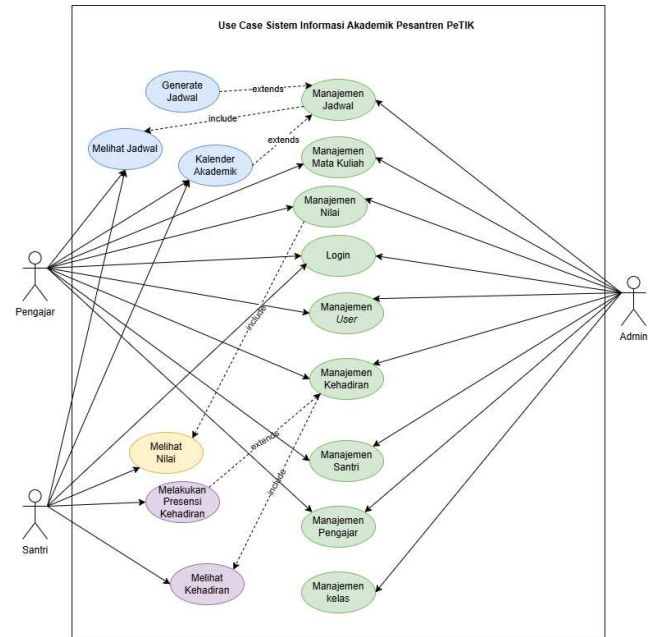
Model data dirancang untuk menggambarkan struktur penyimpanan dan hubungan antar entitas yang digunakan dalam basis data sistem informasi akademik. Perancangan ini menggunakan pendekatan *physical data model*, yaitu model yang menunjukkan entitas dalam bentuk tabel lengkap dengan atribut, tipe data, serta relasi antar entitas secara rinci [9].



Gambar 2. Physical model data Sistem Informasi Akademik PeTIK

Pada Gambar 2 terlihat bahwa model data fisik yang dikembangkan merepresentasikan seluruh elemen penting dalam proses akademik, meliputi tabel *users*, *students*, *teachers*, *subjects*, *classes*, *schedules*, *grades*, *attendances*, *grade_weights*, dan *sessions*.

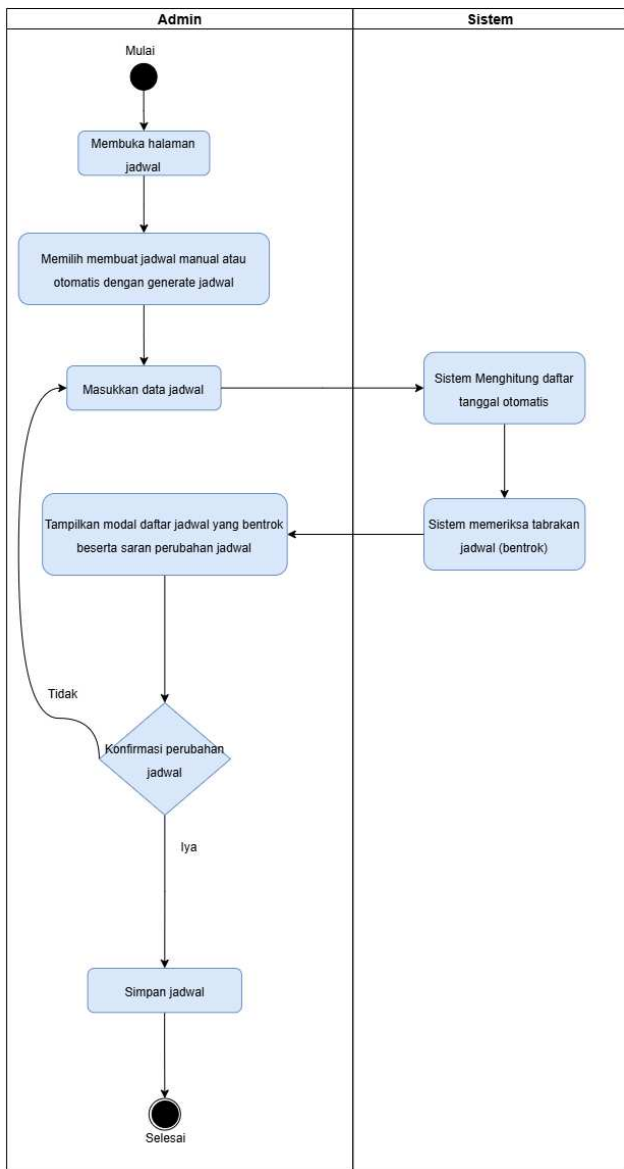
Setelah merancang *physical model data*, selanjutnya adalah merancang *use case diagram* untuk memvisualisasikan interaksi antara pengguna (aktor) dan sistem. Diagram ini memberikan gambaran umum mengenai fitur-fitur yang tersedia dalam sistem, serta siapa saja yang dapat mengakses masing-masing fitur.



Gambar 3. Use Case Diagram Sistem Informasi Akademik PeTIK

Pada Gambar 3 terlihat bahwa setiap pengguna memiliki hak akses yang berbeda sesuai dengan perannya dalam sistem. Admin memiliki kewenangan penuh untuk mengelola seluruh komponen sistem, termasuk data pengguna, jadwal, mata kuliah, nilai, kehadiran, santri, pengajar, dan kelas. Pengajar memperoleh akses terhadap fitur-fitur yang berkaitan langsung dengan kegiatan pembelajaran, seperti melihat jadwal dan kalender akademik, mengelola nilai, serta mencatat kehadiran santri. Sementara itu, Santri sebagai pengguna akhir memiliki akses terbatas, yang mencakup fitur presensi, serta tampilan jadwal dan nilai. Pembagian peran ini dirancang untuk menjaga efisiensi, keamanan, dan keteraturan dalam pengelolaan data akademik melalui sistem yang terintegrasi.

Terakhir adalah merancang *activity diagram*. *Activity diagram* menggambarkan alur aktivitas dalam sistem secara lebih detail, terutama untuk memvisualisasikan bagaimana suatu proses dijalankan oleh sistem maupun pengguna [10]. Diagram ini membantu memetakan urutan langkah-langkah logis dalam setiap *use case*. Berikut adalah *activity diagram* utama dalam sistem.



Gambar 4. Activity Diagram Pembuatan Jadwal

Pada Gambar 4 terlihat proses pembuatan jadwal dimulai saat admin membuka halaman jadwal dan memilih antara menambahkan satu jadwal atau menggunakan fitur *generate* untuk membuat beberapa jadwal sekaligus. Sistem kemudian meminta admin mengisi formulir yang mencakup mata kuliah, kelas, dan waktu perkuliahan. Sebelum jadwal disimpan, sistem melakukan validasi bentrok. Jika ditemukan konflik, sistem akan merekomendasikan waktu alternatif terdekat. Admin dapat memilih untuk menyetujui atau mengubah jadwal tersebut. Setelah dikonfirmasi, jadwal baru akan disimpan secara otomatis.

3.4 Hasil Implementasi

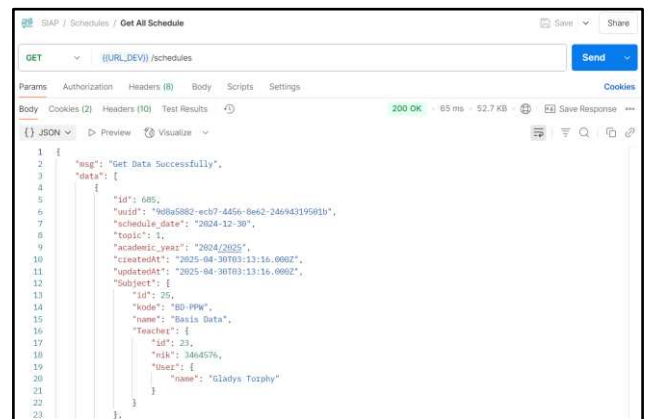
Implementasi sistem informasi akademik dilakukan berdasarkan hasil perancangan yang telah disusun sebelumnya. Sistem dibangun menggunakan pendekatan arsitektur *client-server* dengan pemisahan antara *frontend* dan *backend*, yang berkomunikasi melalui REST API. *Backend* dikembangkan menggunakan Express.js,

sedangkan *frontend* dibangun dengan React.js. Sistem ini juga terhubung dengan basis data MySQL untuk menyimpan dan mengelola data akademik secara terstruktur.

1) Implementasi *Backend*

Backend mengelola berbagai *endpoint* yang saling terintegrasi. Autentikasi pengguna dilakukan melalui *endpoint* `/api/v1/login`, yang akan mengarahkan pengguna ke *dashboard* sesuai perannya. Seluruh API memiliki kontrol akses berbasis *role*.

Pengelolaan data pengguna, santri, pengajar, kelas, dan jadwal dilakukan melalui *endpoint* terpisah, masing-masing memiliki validasi dan relasi antar tabel. Data akademik seperti nilai dan presensi juga dikelola terstruktur dengan dukungan filter berdasarkan mata kuliah, kelas, dan pertemuan.



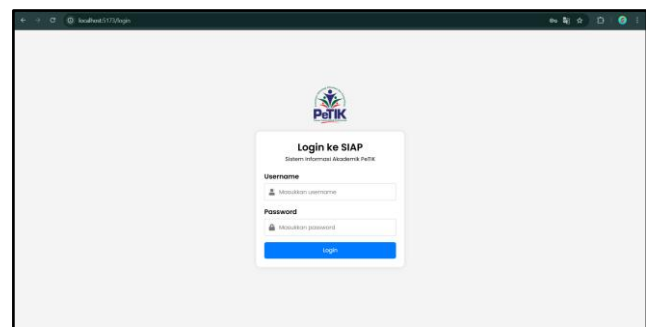
Gambar 5. Response GET schedules

Pada Gambar 5 terlihat *response* dari *backend* untuk menampilkan data jadwal dengan menggunakan *method* GET dan *endpoint* `/api/v1/schedules`.

2) Implementasi *Frontend*

Antarmuka pengguna dirancang modular menggunakan komponen React.

a. Halaman *Login*



Gambar 6. Halaman login

Tabel 3. Hasil Pengujian dengan Pengajar

No	Fitur	Aksi Pengujian	Waktu	Interaksi Pengguna	Respons Sistem	Hasil Pengujian
1	Jadwal Perkuliahan	Tambah Jadwal	15 Mei 2025, 16:10	Admin mengisi <i>form</i> jadwal dan menekan tombol simpan	Sistem menyimpan dan menampilkan data pada kalender akademik	Berhasil
2		<i>Generate</i> Jadwal	15 Mei 2025, 16:20	Admin mengisi <i>form</i> jadwal dan menekan tombol simpan	Sistem membuat <i>generate</i> jadwal selama satu semester dengan menghindari bentrok jadwal	Berhasil
3		Lihat Jadwal	15 Mei 2025, 16:35	Admin membuka halaman jadwal	Sistem menampilkan daftar jadwal sesuai filter tahun, kelas, mata, kuliah	Berhasil
4		Lihat Kalender	15 Mei 2025, 16:37	Pengguna melihat kalender akademik	Sistem menampilkan kalender akademik dengan menampilkan jadwal yang ada beserta hari libur nasional	Berhasil
5		<i>Update</i> Jadwal	15 Mei 2025, 16:45	Admin memilih jadwal lalu mengubah data	Sistem menampilkan <i>form edit</i> dan menyimpan perubahan	Berhasil
6		Hapus Jadwal	15 Mei 2025, 16:50	Admin menekan tombol hapus pada jadwal	Sistem menghapus dan memperbarui tampilan jadwal	Berhasil
7	Manajemen Data User	Tambah Santri	15 Mei 2025, 16:55	Admin mengisi data santri dan simpan	Sistem membuat <i>user</i> baru dengan <i>role</i> santri	Berhasil
8		Tambah Pengajar	15 Mei 2025, 17:00	Admin mengisi data pengajar dan simpan	Sistem membuat <i>user</i> baru dengan <i>role</i> pengajar	Berhasil
9		Tambah <i>User</i>	15 Mei 2025, 17:10	Admin mengisi data <i>user</i> sesuai <i>role</i>	Sistem membuat <i>user</i> baru dengan <i>role</i> yang dipilih oleh admin	Berhasil
10		Lihat Daftar Pengajar	15 Mei 2025, 17:25	Admin membuka halaman pengajar	Sistem menampilkan daftar pengajar	Berhasil
11		Lihat Daftar <i>User</i>	15 Mei 2025, 17:30	Admin membuka halaman <i>user</i>	Sistem menampilkan daftar <i>user</i>	Berhasil
12		<i>Update</i> Data Santri	15 Mei 2025, 17:35	Admin mengubah data santri yang dipilih	Sistem menampilkan <i>form update</i> dan menyimpan perubahan	Berhasil
13		<i>Update</i> Data Pengajar	15 Mei 2025, 17:45	Admin mengubah data pengajar yang dipilih	Sistem menampilkan <i>form update</i> dan menyimpan perubahan	Berhasil
14		<i>Update</i> Data <i>User</i>	15 Mei 2025, 17:55	Admin mengubah data <i>user</i> yang dipilih	Sistem menampilkan <i>form update</i> dan menyimpan perubahan	Berhasil
15		Hapus <i>User</i>	15 Mei 2025, 18:00	Admin menghapus salah satu <i>user</i>	Sistem menghapus data <i>user</i> beserta data relasi (santri/pengajar)	Berhasil
16	Nilai Santri	Tambah Nilai	16 Mei 2025, 14:00	Pengajar memilih mata kuliah dan mengisi nilai tugas/ <i>quiz</i> /ujian	Sistem menyimpan nilai dan menghitung nilai akhir	Berhasil
17		Lihat Nilai	16 Mei 2025, 14:10	Pengajar atau santri membuka halaman nilai	Sistem menampilkan nilai berdasarkan mata kuliah dan pertemuan	Berhasil
18		<i>Update</i> Nilai	16 Mei 2025, 14:15	Pengajar memilih nilai lalu mengubah nilainya	Sistem memperbarui data nilai dan rekap nilai akhir	Berhasil
19		Hapus Nilai	16 Mei 2025, 14:50	Pengajar menghapus nilai santri	Sistem menghapus nilai dan memperbarui tampilan	Berhasil
20		Rekap Nilai	16 Mei 2025, 14:55	Pengajar membuka halaman rekap	Sistem menampilkan nilai akhir, nilai huruf, dan peringkat	Berhasil
21		Presensi Kehadiran	<i>Input</i> Kehadiran	16 Mei 2025, 15:00	Pengajar memilih jadwal lalu <i>input</i> kehadiran santri	Sistem menyimpan status kehadiran per pertemuan
22	Lihat Kehadiran		16 Mei 2025, 15:10	Pengajar atau santri membuka halaman presensi	Sistem menampilkan tabel kehadiran berdasarkan mata kuliah	Berhasil

No	Fitur	Aksi Pengujian	Waktu	Interaksi Pengguna	Respons Sistem	Hasil Pengujian
23		<i>Update Kehadiran</i>	16 Mei 2025, 15:15	Pengajar mengubah status kehadiran santri	Sistem memperbarui status kehadiran	Berhasil
24	Data Mata Kuliah	Tambah Mata Kuliah	16 Mei 2025, 15:25	Admin membuat mata kuliah	Sistem menyimpan mata kuliah	Berhasil
25		Lihat Mata Kuliah	16 Mei 2025, 15:30	Admin atau pengajar membuka halaman mata kuliah	Sistem menampilkan nilai berdasarkan mata kuliah dan pertemuan	Berhasil
26		<i>Update Mata Kuliah</i>	16 Mei 2025, 15:40	Admin mengubah data mata kuliah	Sistem memperbarui data nilai dan rekap nilai akhir	Berhasil
27		Hapus Mata Kuliah	16 Mei 2025, 15:45	Admin menghapus mata kuliah	Sistem menghapus nilai dan memperbarui tampilan	Berhasil
28	Data Kelas	Tambah Kelas	16 Mei 2025, 15:50	Admin membuat kelas	Sistem menyimpan mata kuliah	Berhasil
29		Lihat Kelas	16 Mei 2025, 16:10	Admin atau pengajar membuka halaman kelas	Sistem menampilkan nilai berdasarkan mata kuliah dan pertemuan	Berhasil
30		<i>Update Kelas</i>	16 Mei 2025, 16:20	Admin mengubah data kelas	Sistem memperbarui data nilai dan rekap nilai akhir	Berhasil
31		Hapus Kelas	16 Mei 2025, 16:25	Admin menghapus kelas	Sistem menghapus nilai dan memperbarui tampilan	Berhasil
32	<i>Dashboard</i>	Lihat <i>dashboard</i>	16 Mei 2025, 16:30	Pengguna membuka halaman <i>dashboard</i>	Sistem menampilkan halaman <i>dashboard</i> sesuai dengan <i>role</i> pengguna	Berhasil
33	Profile	Lihat <i>Profile</i>	16 Mei 2025, 16:35	Pengguna membuka halaman <i>profile</i>	Sistem menampilkan data pengguna yang sedang <i>login</i>	Berhasil
34		<i>Update Profile</i>	16 Mei 2025, 16:40	Pengguna mengubah data diri sendiri	Sistem mengubah data pengguna dan menyimpannya.	Berhasil
35		Autentikasi dan Akses	<i>Login</i>	15 Mei 2025, 16:00	Pengguna mengisi <i>username</i> dan <i>password</i>	Sistem memverifikasi dan mengarahkan ke <i>dashboard</i> sesuai <i>role</i>
36	<i>Logout</i>		15 Mei 2025, 18:00	Pengguna menekan tombol <i>logout</i>	Sistem menghapus sesi dan kembali ke halaman <i>login</i>	Berhasil
37	<i>Session</i>		16 Mei 2025, 16:55	Pengguna menunggu selama 30 menit setelah <i>login</i> dan tidak melakukan aktivitas apa pun di aplikasi	Sistem otomatis melakukan <i>logout</i> akun ketika tidak ada aktivitas selama 30 menit di aplikasi berupa gerakan <i>cursor mouse</i> , <i>scroll</i> atau klik.	Berhasil
38		Akses Terbatas	16 Mei 2025, 16:45	<i>User</i> mencoba membuka halaman bukan hak aksesnya	Sistem mengarahkan ke halaman yang sesuai atau <i>error 403</i>	Berhasil

2. Wawancara

Setelah pengujian teknis dilakukan, proses evaluasi dilanjutkan dengan wawancara kepada perwakilan dari masing-masing kelompok pengguna, yaitu admin akademik, pengajar, dan santri. Wawancara bertujuan untuk mengevaluasi pengalaman penggunaan sistem serta mengidentifikasi potensi pengembangan lebih lanjut.

Admin Akademik menyatakan bahwa fitur *generate* jadwal otomatis sangat membantu dalam menyusun jadwal mata pelajaran secara efisien. Proses yang sebelumnya memakan waktu cukup lama kini dapat dilakukan dengan cepat dan lebih terorganisir. Admin juga merasa bahwa antarmuka sistem mudah dipahami meskipun tanpa panduan tertulis. Saran yang diberikan lebih mengarah pada peningkatan stabilitas dan keberlanjutan sistem agar dapat berjalan secara konsisten dalam jangka panjang.

Pengajar menyampaikan bahwa fitur pencatatan nilai, presensi, dan jadwal mengajar memberikan kemudahan signifikan dalam menjalankan tugas akademik harian. Aplikasi dinilai lebih efisien dibandingkan metode manual sebelumnya, terutama dalam melakukan *input* nilai dan presensi, serta kemudahan dalam memantau kalender akademik. Pengguna juga merasa cepat terbiasa dengan sistem tanpa memerlukan pelatihan khusus.

Santri sebagai pengguna akhir merasakan kemudahan dalam mengakses nilai dan data kehadiran secara mandiri. Mereka mengapresiasi tampilan aplikasi yang sederhana dan informatif, serta transparansi informasi yang ditampilkan. Sebelumnya, rekap nilai baru tersedia di akhir semester, namun kini dapat diakses secara *real-time* setelah dimasukkan. Hal ini meningkatkan keterlibatan dan kesadaran santri terhadap perkembangan akademik mereka.

Secara keseluruhan, hasil wawancara menunjukkan bahwa sistem informasi akademik ini telah berhasil memenuhi kebutuhan utama pengguna dan memberikan pengalaman penggunaan yang baik. Respons positif dari seluruh pihak menunjukkan bahwa sistem tidak hanya berjalan sesuai rancangan teknis, tetapi juga memberikan dampak nyata dalam mendukung proses administrasi dan pembelajaran di lingkungan Pesantren PeTIK.

4. KESIMPULAN

Penelitian ini berhasil merancang dan membangun sistem informasi akademik berbasis web untuk Pesantren PeTIK dengan memanfaatkan Node.js dan Express.js di sisi *backend*, React.js di sisi *frontend*, serta MySQL sebagai basis data terstruktur. Pengujian fungsional dan evaluasi pengguna mengonfirmasi bahwa:

1. Admin akademik kini dapat menyusun jadwal dalam waktu singkat dan meminimalkan konflik antar pengajar.
2. Pengajar lebih mudah mengelola nilai dan presensi melalui satu antarmuka terpusat.
3. Santri memperoleh akses mandiri dan *real-time* terhadap informasi akademik mereka.

Dengan demikian, sistem ini sukses menggantikan prosedur manual dan meningkatkan efisiensi sekaligus transparansi di lingkungan Pesantren PeTIK. Untuk penelitian selanjutnya, berikut beberapa pengembangan yang bisa ditambahkan sebagai:

1. Menambahkan fitur *generate* ubah jadwal otomatis yang ketika mengubah satu jadwal maka jadwal setelahnya kan berubah menyesuaikan jadwal yang diubah.
2. Menambahkan fitur notifikasi, pengumuman dan cetak laporan seperti nilai dan kehadiran.

DAFTAR PUSTAKA

- [1] A. Setiawan, S. Samsugi, and D. Alita, "Rancang Bangun Sistem Informasi Akademik SMK Taman Siswa 1 Tanjung Karang Berbasis Web," *Jurnal Informatika dan Rekayasa Perangkat Lunak*, vol. 4, no. 1, pp. 53–59, Mar. 2023, doi: <https://doi.org/10.33365/jatika.v4i1.2465>
- [2] T. Ardiansah, Y. Rahmanto and Z. Amir, "Penerapan *Extreme Programming* dalam Sistem Informasi Akademik SDN Kuala Teladas," *Journal of Information Technology, Software Engineering and Computer Science (ITSECS)*, vol. 1, 2023.
- [3] M. Solahudin, "Rancang Bangun Sistem Informasi Akademik Sekolah (SIAS) Berbasis *Website*," *DoubleClick: Journal of Computer and Information Technology*, vol. 4, no. 2, p. 107, Jan. 2021, doi: <https://doi.org/10.25273/doubleclick.v4i2.8315>
- [4] N. M. Arofiq, R. F. Erlangga, A. Irawan, M. and A. Saifudin, "Pengujian Fungsional Aplikasi Inventory Barang Kedatangan Dengan Metode Black Box Testing Bagi Pemula," *Jurnal Ilmu Komputer dan Science*, vol. 2, no. 5, pp. 1322-1330, 2023
- [5] A. Sutedi, Y. Septiana, and R. A. Halim, "Sistem Informasi Akademik Santri Berbasis Web di Pondok Pesantren," *Jurnal Algoritma*, vol. 18, no. 1, pp. 151–161, 2021.
- [6] M. R. Fadli, "Memahami desain metode penelitian kualitatif," *Humanika*, vol. 21, no. 1, pp. 33–54, Apr. 2021, doi: [10.21831/hum.v21i1.38075](https://doi.org/10.21831/hum.v21i1.38075).
- [7] N. Nurhidayah and I. B., "Analisis Kualitatif Hubungan Budaya Kerja Organisasi dengan Opini Audit," *Owner (Riset J. Akuntansi)*, vol. 4, no. 2, p. 505, 2020, doi: [10.33395/owner.v4i2.303](https://doi.org/10.33395/owner.v4i2.303).
- [8] R. Destriana, et. al., "Diagram UML dalam Membuat Aplikasi Android Firebase "Studi Kasus Aplikasi Bank Sampah"," D.I. Yogyakarta: *Deepublish*, 2021.
- [9] I. R. Mukhlis, D. Hermansyah, and V. M. Lantang, "Rancangan Basis Data Transaksi pada PT. Bank Perkreditan Rakyat ABC Menggunakan MySQL dengan Model *Entity Relationship Diagram* (ERD) dan *Physical Data Model* (PDM)," *Journal of Advances in Information and Industrial Technology*, vol. 5, no. 1, pp. 1–10, 2023, doi: <https://doi.org/10.52435/jaiit.v5i1.305>.
- [10] F. A. Fikrian dan A. R. Adriansyah, "Pengembangan Aplikasi Doa Harian menggunakan Metode Hybrid untuk Web dan Mobile," *j. inform. terpadu*, vol. 8, no. 2, hlm. 93–103, 2022.