



Rancang Bangun Sistem Manajemen Jadwal Pelajaran Berbasis Mobile Menggunakan Flutter dan Node.js dengan Fitur Push Notification

Ja'far*, Wahyu Sri Utami

Fakultas Sains & Teknologi, Program Studi Informatika, Universitas Teknologi Yogyakarta, Yogyakarta, Indonesia

Email: ¹jakfarshafwan013@gmail.com, ²wahyu.utami@uty.ac.id

Email Penulis Korespondensi: jakfarshafwan013@gmail.com

Abstrak—Perkembangan teknologi informasi mendorong institusi pendidikan untuk melakukan transformasi digital dalam pengelolaan data dan layanan akademik, khususnya pada sistem penjadwalan pelajaran. Pada SMP Muhammadiyah Mlonggo Jepara, tahapan perancangan jadwal pembelajaran masih mengandalkan metode konvensional yang melibatkan papan pengumuman dan berkas tercetak. Metode tersebut menimbulkan berbagai permasalahan, antara lain mengakibatkan penundaan dalam diseminasi informasi, tingginya potensi kesalahan data, serta tidak adanya mekanisme terpusat untuk menjaga konsistensi dan akurasi jadwal. Permasalahan perancangan jadwal berdampak langsung pada efektivitas kegiatan belajar mengajar bagi guru maupun siswa. Penelitian ini bertujuan untuk merancang dan mengimplementasikan aplikasi jadwal pelajaran berbasis mobile menggunakan Flutter dan Node.js yang dilengkapi dengan fitur push notification otomatis guna meningkatkan efisiensi, akurasi, dan ketepatan waktu manajemen jadwal sekolah. Metode pengembangan perangkat lunak yang digunakan adalah model Waterfall, yang meliputi tahap analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Sistem backend dikembangkan menggunakan RESTful API berbasis Node.js/Express dengan basis data MySQL untuk menjamin integritas dan konsistensi data. Aplikasi ini dilengkapi dengan mekanisme validasi data dan deteksi konflik jadwal (conflict detection) untuk mencegah terjadinya benturan waktu pada saat proses input dan pembaruan jadwal, sehingga akurasi informasi dapat terjaga. Pengujian fungsional dilakukan menggunakan metode Blackbox Testing terhadap seluruh fitur utama aplikasi. Temuan hasil penelitian menunjukkan bahwa aplikasi mampu menyajikan jadwal pelajaran secara real-time, mempermudah proses pembaruan informasi, serta memberikan notifikasi kepada guru dan siswa sebagai pengingat kegiatan belajar mengajar. Hasil pengujian menunjukkan bahwa 100% fungsi berjalan sesuai dengan spesifikasi yang telah ditetapkan. Dengan demikian, aplikasi ini dapat meningkatkan efisiensi, akurasi, dan efektivitas dalam pengelolaan jadwal pelajaran di lingkungan sekolah. Secara keseluruhan, aplikasi mampu menyajikan jadwal pelajaran secara real-time, mempercepat proses pembaruan informasi, serta memberikan notifikasi pengingat kepada guru dan siswa, sehingga mendukung pengelolaan jadwal pembelajaran yang lebih efektif dan andal di lingkungan sekolah.

Kata Kunci: Flutter; Jadwal Pelajaran; Node.js; Notifikasi; RESTful API

Abstract—The development of information technology encourages educational institutions to undertake digital transformation in data management and academic services, particularly in the lesson scheduling system. At Muhammadiyah Mlonggo Junior High School in Jepara, the learning schedule design stage still relies on conventional methods involving bulletin boards and printed files. This method causes various problems, including delays in information dissemination, a high potential for data errors, and the absence of a centralized mechanism to maintain schedule consistency and accuracy. Schedule design problems have a direct impact on the effectiveness of teaching and learning activities for both teachers and students. This study aims to design and implement a mobile-based lesson schedule application using Flutter and Node.js equipped with an automatic push notification feature to improve the efficiency, accuracy, and timeliness of school schedule management. The software development method used is the Waterfall model, which includes the stages of needs analysis, system design, implementation, testing, and maintenance. The backend system was developed using a Node.js/Express-based RESTful API with a MySQL database to ensure data integrity and consistency. This application is equipped with a data validation mechanism and schedule conflict detection to prevent time conflicts during the input and update process of the schedule, so that information accuracy can be maintained. Functional testing was conducted using the Blackbox Testing method on all the main features of the application. The findings of the research results show that the application is able to present lesson schedules in real-time, simplify the process of updating information, and provide notifications to teachers and students as reminders of teaching and learning activities. The test results show that 100% of the functions run according to the established specifications. Thus, this application can improve efficiency, accuracy, and effectiveness in managing lesson schedules in the school environment. Overall, the application is able to present lesson schedules in real-time, speed up the process of updating information, and provide reminder notifications to teachers and students, thus supporting more effective and reliable management of learning schedules in the school environment.

Keywords: Flutter; Class Schedule; Node.js; Notification; RESTful API

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi dalam beberapa dekade terakhir telah membawa perubahan signifikan dalam pengelolaan sistem pendidikan. Digitalisasi sistem informasi di lingkungan sekolah dinilai mampu meningkatkan efisiensi, akurasi, dan kecepatan penyampaian informasi akademik kepada seluruh pemangku kepentingan (Safitri & Puspitorini, 2021). Hal ini menuntut institusi pendidikan untuk terus beradaptasi agar proses administrasi tidak lagi menghambat kegiatan utama di sekolah.

Salah satu komponen paling vital dalam administrasi akademik adalah pengelolaan jadwal pelajaran. Jadwal berfungsi sebagai pedoman utama bagi guru dan siswa dalam melaksanakan kegiatan belajar mengajar secara terstruktur. Integrasi teknologi di sekolah memiliki dampak positif pada pembelajaran siswa dan prestasi akademik, terutama ketika diterapkan dalam praktik yang berpusat pada siswa, menjadi bagian dari inisiatif terstruktur, serta disertai dengan pengembangan profesional bagi guru (Gupta & Goel, 2024).



Manajemen penjadwalan yang efektif di sekolah menjadi krusial karena melibatkan koordinasi kompleks antara ketersediaan ruang, waktu, dan tenaga pengajar secara simultan (Bontekoe et al., 2006). Tanpa manajemen jadwal yang baik, alokasi waktu di sekolah menjadi tidak teratur sehingga pada akhirnya menurunkan kualitas pendidikan. Namun, implementasi teknologi digital dalam pengelolaan jadwal belum merata di setiap institusi. Beberapa sekolah masih mengandalkan metode konvensional seperti papan pengumuman fisik dan dokumen cetak yang ditempel di dinding kelas. Metode manual ini tidak hanya memakan waktu dalam penyusunannya, tetapi juga sangat tidak fleksibel terhadap dinamika perubahan yang sering terjadi di lingkungan sekolah. Kondisi serupa ditemukan di SMP Muhammadiyah Mlonggo Jepara, di mana penyampaian jadwal pelajaran masih dilakukan sepenuhnya secara manual. Guru dan siswa harus memeriksa papan pengumuman secara rutin untuk mendapatkan informasi terbaru. Cara ini sering kali menimbulkan kendala serius, terutama ketika ada perubahan jadwal mendadak yang tidak segera diketahui oleh pihak terkait karena keterbatasan media penyampaian informasi.

Masalah keterlambatan informasi ini berdampak langsung pada terganggunya efektivitas proses belajar mengajar. Risiko kesalahan jadwal atau tumpang tindih jam mengajar menjadi lebih tinggi saat pengelolaan dilakukan tanpa sistem yang terintegrasi. Selain itu, ketergantungan pada dokumen fisik membuat mobilitas guru dan siswa terhambat karena informasi tidak dapat diakses secara fleksibel dari luar lingkungan sekolah. Seiring dengan tingginya penggunaan perangkat mobile di kalangan siswa dan guru, pengembangan aplikasi jadwal pelajaran berbasis mobile dinilai sebagai solusi yang efektif dan sesuai dengan kebutuhan saat ini. Aplikasi pendidikan berbasis mobile dapat mendorong pembelajaran interaktif dan kolaboratif (Hussain et al., 2020).

Sejumlah penelitian terdahulu telah berupaya menjawab tantangan ini dengan platform yang beragam. Hananto dkk. (2022) mengembangkan aplikasi berbasis Android di SMPN 1 Tasikmadu untuk membantu keterlibatan siswa. Selaras dengan penelitian tersebut, Qottrunnada, Al Ikhsan, dan Eosina (2023) merancang sistem manajemen jadwal guru berbasis mobile pada MTs Sirojul Ikhsan, yang menunjukkan bahwa platform seluler memberikan kemudahan akses informasi jadwal secara lebih fleksibel dibandingkan metode manual. Hafizh, Dhika, dan Setiawan (2021) turut mendorong digitalisasi dengan menerapkan aplikasi jadwal pelajaran berbasis Java di SMAN 16 Bekasi sebagai pengganti pendataan konvensional. Sementara itu, Pangestu (2021) merancang aplikasi desktop berbasis Java di SMK Budi Asih yang berfokus pada penyusunan jadwal secara otomatis untuk meminimalkan konflik waktu mengajar. Sementara itu, penelitian yang dilakukan oleh Arif Budiman dkk. (2022) dalam jurnal berjudul “Aplikasi Manajemen Jadwal Mengajar Guru Sekolah Menengah Atas” membahas pengembangan aplikasi berbasis teknologi untuk membantu proses penjadwalan kegiatan mengajar guru di tingkat SMA. Tujuan dari penelitian ini adalah untuk menciptakan solusi digital yang dapat mengelola jadwal mengajar secara otomatis dan efisien. Hasil penelitian menunjukkan bahwa aplikasi yang dibangun mampu meningkatkan efisiensi dalam proses penjadwalan, meminimalkan konflik waktu, dan memberikan kemudahan dalam pendistribusian informasi kepada guru.

Selain platform seluler dan desktop, sistem berbasis web juga banyak dikembangkan untuk akses daring. Safitri dan Puspitorini (2021) merancang sistem penjadwalan menggunakan PHP dan MySQL yang terbukti meningkatkan efektivitas distribusi jadwal bagi guru. Sementara itu, Yona (2022) menggunakan model pengembangan perangkat lunak Waterfall untuk memastikan sistem yang dibangun stabil, terstruktur, dan memiliki dokumentasi teknis yang baik. Hal ini sejalan dengan penelitian yang dilakukan oleh Pratiwi, Anardani, dan Putera (2023) yang mengimplementasikan metode Waterfall dalam rancang bangun sistem informasi penjadwalan mata pelajaran untuk meningkatkan efisiensi manajemen data sekolah secara terstruktur.

Meskipun digitalisasi jadwal telah banyak dilakukan, masih terdapat celah penelitian (*research gap*) yang krusial. Kesenjangan pertama terletak pada keterbatasan mobilitas, di mana sebagian besar sistem terdahulu masih berbasis web atau desktop yang kurang praktis bagi pengguna yang selalu bergerak (Hafizh et al., 2021; Safitri & Puspitorini, 2021; Pangestu, 2021). Meskipun penelitian oleh Pratiwi dkk. (2023) telah berhasil membangun fondasi sistem yang stabil dengan metode *Waterfall*, fokusnya masih terbatas pada pengelolaan informasi secara statis di sisi admin. Pada sistem yang sudah berbasis mobile sekalipun, seperti penelitian Qottrunnada dkk. (2023), tantangan berikutnya muncul pada efisiensi pengembangan lintas platform dan sinkronisasi data secara real-time. Kedua, terdapat kekurangan pada mekanisme pengingat proaktif; sistem yang ada umumnya bersifat pasif sehingga risiko keterlambatan informasi tetap tinggi karena minimnya fitur *notification* (Budiman et al., 2021). Ketiga, pemanfaatan *framework multiplatform* modern seperti Flutter dalam sistem sekolah masih sangat terbatas dibandingkan teknologi *native* Android (Hananto et al., 2022; Qottrunnada et al., 2023). Terakhir, fitur notifikasi yang tersedia pada beberapa penelitian sebelumnya belum terintegrasi secara otomatis dengan perubahan data di sisi server, sehingga belum mampu memberikan peringatan real-time saat terjadi pembaruan jadwal mendadak (Yona, 2022). Selain itu, pemanfaatan *framework multiplatform* modern seperti Flutter dalam konteks sistem informasi sekolah masih relatif terbatas.

Berdasarkan celah tersebut, penelitian ini menawarkan kebaruan melalui pengembangan aplikasi jadwal pelajaran berbasis Flutter dan Node.js dengan mekanisme otomatisasi notifikasi proaktif berbasis perubahan data di server. Berbeda dengan pendekatan notifikasi konvensional, sistem yang dikembangkan mampu mengirimkan peringatan secara real-time kepada pengguna setiap kali admin melakukan perubahan jadwal pada backend Node.js melalui RESTful API. Dengan pendekatan ini, notifikasi tidak hanya berfungsi sebagai pengingat, tetapi sebagai *real-time alert system* yang memastikan informasi terbaru langsung diterima oleh guru dan siswa tanpa keterlambatan.

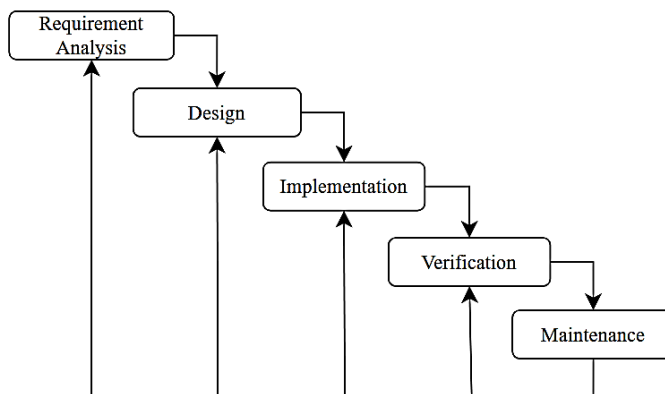
Melalui pendekatan teknologi ini, SMP Muhammadiyah Mlonggo Jepara diharapkan dapat mentransformasi administrasi manual menuju sistem digital yang lebih transparan dan efisien. Dengan aplikasi ini, guru dan siswa dapat mengakses jadwal secara real-time melalui smartphone masing-masing, sehingga meminimalisir risiko kesalahan

informasi. Selain memberikan kemudahan akses, implementasi aplikasi ini diharapkan mampu meningkatkan literasi digital bagi tenaga pendidik maupun siswa (Prasasti & Wahyudi, 2025). Secara akademik, penelitian ini diharapkan dapat menjadi referensi dalam pengembangan sistem informasi sekolah berbasis arsitektur API modern yang mendukung pembaruan data dan notifikasi secara proaktif serta skalabel.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak (software engineering) dalam merancang dan mengimplementasikan aplikasi jadwal pelajaran berbasis mobile. Pendekatan ini dipilih karena mampu menyediakan kerangka kerja sistematis dalam menghasilkan perangkat lunak yang memenuhi kebutuhan pengguna secara terstruktur dan terdokumentasi dengan baik (Pressman & Maxim, 2021). Model pengembangan perangkat lunak yang digunakan adalah Waterfall Model (Model Sekuensial Linear). Model Waterfall merupakan pendekatan klasik dalam rekayasa perangkat lunak yang menekankan pada urutan tahapan yang jelas, sistematis, dan terdokumentasi secara ketat (Royce, 2021; Sommerville, 2016). Setiap tahap harus diselesaikan secara lengkap sebelum melanjutkan ke tahap berikutnya, sehingga model ini cocok digunakan pada sistem dengan kebutuhan yang relatif stabil dan ruang lingkup yang jelas, seperti sistem informasi akademik di lingkungan sekolah.



Gambar 1. Metode Waterfall

Tahapan pengembangan sistem menggunakan model Waterfall yang diterapkan dalam penelitian dapat dilihat pada Gambar 1. Model ini terdiri dari lima tahap utama, yaitu *Requirement Analysis*, *Design*, *Implementation*, *Verification*, dan *Maintenance*, yang dikerjakan secara berurutan dan saling berkaitan. Setiap tahapan harus diselesaikan secara menyeluruh sebelum melanjutkan ke tahap berikutnya, sehingga meminimalkan kesalahan pada tahap akhir pengembangan. Alur sekuensial pada Gambar 1 menjadi dasar dalam pelaksanaan metodologi penelitian ini, di mana hasil dari setiap tahap digunakan sebagai input bagi tahap selanjutnya untuk memastikan pengembangan aplikasi berjalan sistematis dan terstruktur.

Tahapan-tahapan yang dilaksanakan dalam penelitian ini meliputi:

1. *Requirement Analysis* (Analisis Kebutuhan): Tahap analisis kebutuhan merupakan fondasi utama dalam model Waterfall. Pada tahap ini dilakukan pengumpulan kebutuhan sistem melalui observasi dan wawancara dengan pihak sekolah, guru, dan siswa di SMP Muhammadiyah Mlonggo Jepara. Tujuan tahap ini adalah mengidentifikasi kebutuhan fungsional dan non-fungsional sistem, serta permasalahan yang muncul akibat penggunaan metode konvensional dalam pengelolaan jadwal pelajaran. Menurut Sommerville (2016), analisis kebutuhan yang komprehensif akan meminimalkan kesalahan desain pada tahap selanjutnya.
2. *Design* (Perancangan Sistem): Tahap perancangan bertujuan untuk menerjemahkan kebutuhan sistem ke dalam bentuk desain teknis. Perancangan dilakukan menggunakan Unified Modeling Language (UML), yang meliputi Use Case Diagram, Class Diagram, dan Activity Diagram untuk memodelkan interaksi pengguna dengan sistem (Seidl et al., 2015). Selain itu, perancangan basis data dilakukan menggunakan Entity Relationship Diagram (ERD) untuk memastikan integritas dan konsistensi data.
3. *Implementation* (Pengkodean): Pada tahap implementasi, desain sistem diwujudkan ke dalam kode program menggunakan framework Flutter untuk pengembangan aplikasi mobile lintas platform (Chiandra & Laksana, 2024; Singh & Lodhe, 2021), serta Node.js dengan framework Express.js dan MySQL untuk penyediaan layanan RESTful API dan pengelolaan basis data yang terstruktur (Elmasri & Navathe, 2020). Sistem ini juga mengintegrasikan Firebase Cloud Messaging (FCM) sebagai media pengiriman *push notifications* yang berfungsi untuk memberikan pengingat jadwal secara otomatis serta notifikasi *real-time* saat terjadi perubahan jadwal guna menjamin akurasi informasi bagi pengguna.
4. *Verification* (Verifikasi dan Pengujian): Tahap verifikasi dilakukan untuk memastikan bahwa sistem yang dibangun telah sesuai dengan kebutuhan yang telah ditentukan pada tahap awal. Metode pengujian yang digunakan adalah



Blackbox Testing, yaitu teknik pengujian yang berfokus pada fungsionalitas sistem tanpa memperhatikan struktur internal kode.

5. *Maintenance* (Pemeliharaan): Tahap pemeliharaan merupakan tahap lanjutan setelah sistem diimplementasikan dan digunakan oleh pengguna. Sommerville (2016) menyatakan bahwa pemeliharaan merupakan bagian penting dari siklus hidup perangkat lunak untuk menjaga stabilitas dan keberlanjutan sistem dalam jangka panjang.

2.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengidentifikasi dan memahami secara mendalam permasalahan yang dihadapi oleh SMP Muhammadiyah Mlonggo Jepara terkait pengelolaan jadwal pelajaran yang masih bersifat konvensional, serta menentukan solusi teknologi yang paling sesuai. Merujuk pada temuan dari observasi dan studi wawancara dengan pihak sekolah, guru, dan siswa, diperoleh informasi bahwa proses penyampaian jadwal melalui papan pengumuman atau lembar kertas sering menimbulkan keterlambatan informasi dan kesalahan data. Oleh karena itu, diperlukan sistem digital yang mampu menyajikan informasi jadwal secara cepat, akurat, dan mudah diakses.

Analisis kebutuhan dilakukan untuk menjembatani kesenjangan antara proses pengelolaan jadwal pelajaran saat ini dengan sistem berbasis digital yang diusulkan. Berdasarkan pengguna utama (admin, guru, dan siswa), ditemukan bahwa proses penyampaian jadwal melalui media cetak sering kali tidak efisien, rentan terhadap keterlambatan informasi, dan kesulitan dalam pembaruan mendadak (Yona, 2022). Kondisi ini mendorong perancangan aplikasi yang memprioritaskan penyajian informasi yang cepat, akurat, dan real-time.

2.2.1 Kebutuhan Fungsional

Kebutuhan fungsional sistem didefinisikan sebagai berikut:

1. Autentikasi Pengguna: Sistem harus menyediakan fitur login dan logout untuk tiga peran pengguna: Admin, Guru, dan Siswa.
2. Manajemen Data: Admin memiliki hak akses penuh untuk mengelola data guru, siswa, kelas, mata pelajaran, dan data pengguna (user).
3. Pengelolaan Jadwal: Admin dapat membuat, mengubah, dan menghapus jadwal pelajaran berdasarkan parameter hari, jam, kelas, dan guru pengampu.
4. Tampilan Jadwal: Guru dan siswa dapat melihat jadwal mengajar atau jadwal belajar mereka secara real-time melalui aplikasi mobile.
5. Notifikasi jadwal: Sistem mengirimkan notifikasi jadwal ke perangkat mobile guru dan siswa.

2.2.2 Kebutuhan Non-Fungsional

Aspek non-fungsional yang harus dipenuhi meliputi:

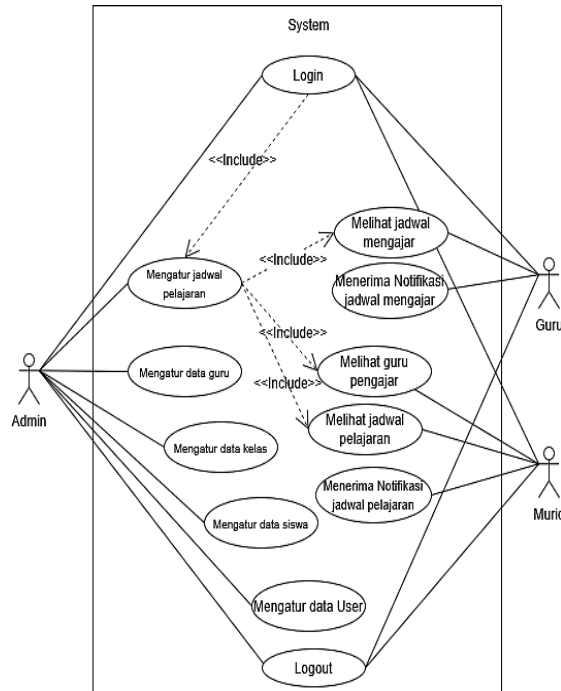
1. Kinerja (Performance): Aplikasi harus mampu memproses dan menampilkan data dengan waktu respons yang cepat dan efisien.
2. Keamanan (Security): Sistem harus menerapkan mekanisme autentikasi login yang kuat untuk membatasi hak akses berdasarkan peran pengguna.
3. Kompatibilitas (Compatibility): Aplikasi harus dapat berjalan stabil pada berbagai perangkat mobile berbasis Android.

2.3 Perancangan Sistem

Perancangan sistem dilakukan untuk memberikan gambaran secara menyeluruh mengenai cara kerja aplikasi jadwal pelajaran berbasis mobile yang dikembangkan. Tahapan ini berfokus pada desain alur interaksi pengguna, struktur data, serta integrasi antara komponen frontend dan backend. Perancangan dilakukan menggunakan pendekatan object-oriented dengan bantuan model Unified Modeling Language (UML), yang meliputi Use Case Diagram, Class Diagram dan Activity Diagram.

2.3.1 Use Case Diagram

Use Case Diagram yang merepresentasikan interaksi antara aktor dan sistem aplikasi jadwal pelajaran berbasis mobile pada Gambar 2. Diagram ini melibatkan tiga aktor utama, yaitu admin, guru, dan siswa, yang masing-masing memiliki peran dan hak akses berbeda sesuai dengan fungsi sistem. Aktor admin memiliki peran paling dominan karena bertanggung jawab terhadap pengelolaan data guru, data siswa, data kelas, data mata pelajaran, serta data pengguna (user). Selain itu, admin juga memiliki kewenangan utama dalam menyusun dan mengatur jadwal pelajaran. Pada proses pengelolaan jadwal, terdapat relasi `<<include>>` yang menunjukkan bahwa sistem mewajibkan ketersediaan data pendukung sebelum jadwal dapat diproses dan disimpan. Hal ini menegaskan bahwa fungsionalitas pendukung tersebut mutlak diperlukan dalam menjalankan use case utama (Oktarina & Nasari, 2024).

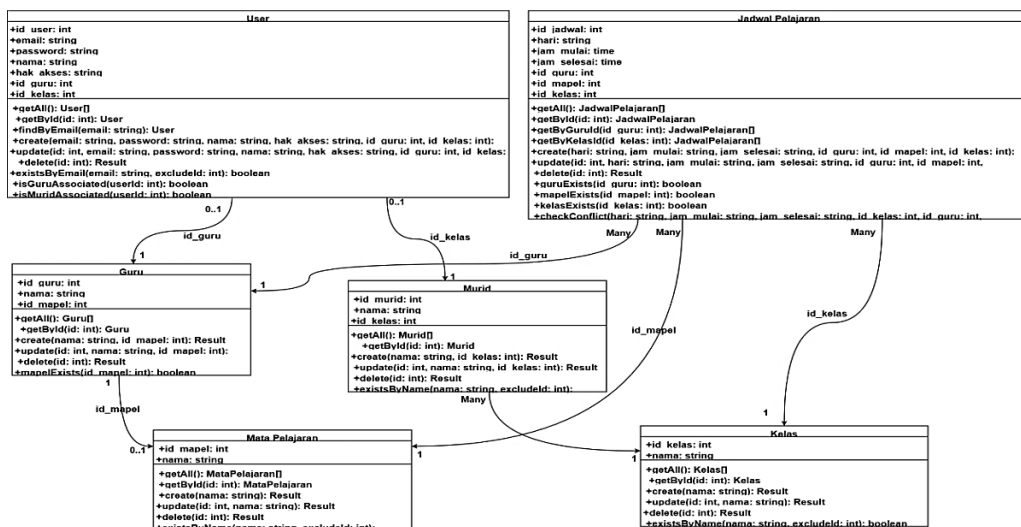


Gambar 2. Use Case Diagram Sistem Jadwal Pelajaran

Aktor guru dan siswa dapat mengakses jadwal masing-masing serta menerima notifikasi pengingat otomatis sebelum kegiatan belajar dimulai guna memastikan penyampaian informasi yang tepat waktu dan akurat. Mekanisme ini diterapkan untuk menjaga keamanan data serta memastikan pembatasan akses sesuai dengan peran pengguna. Dengan demikian, *Use Case Diagram* pada Gambar 2 menjelaskan mengenai ruang lingkup fungsional sistem serta hubungan interaksi antara pengguna dan sistem aplikasi jadwal pelajaran (Safitri & Puspitorini, 2021).

2.3.2 Class Diagram

Class diagram menggambarkan struktur sistem manajemen jadwal pelajaran yang terdiri dari enam entitas utama: user, guru, murid, mata pelajaran, kelas, dan jadwal pelajaran. Setiap entitas memiliki atribut dan metode dasar untuk pengelolaan data (CRUD) serta validasi seperti pengecekan email dan konflik jadwal. Diagram ini juga menunjukkan relasi antar-entitas, misalnya guru mengajar mata pelajaran, kelas berisi banyak murid, dan seluruhnya terintegrasi dalam jadwal pelajaran.

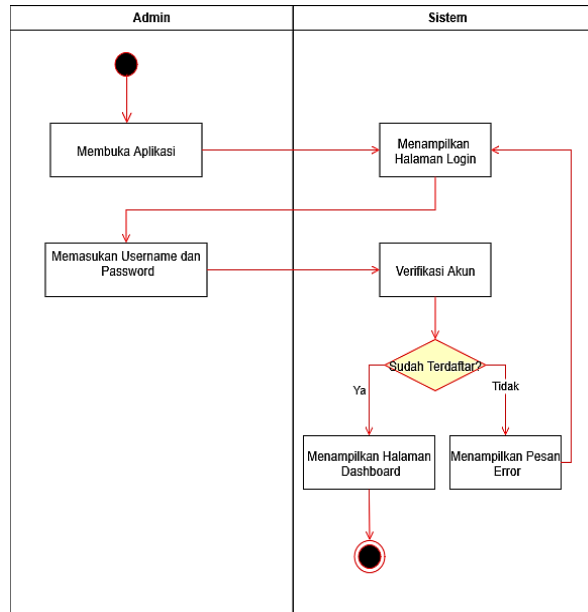


Gambar 3. Class Diagram Sistem Jadwal Pelajaran

Class Diagram pada Gambar 3 mengilustrasikan struktur statis dan hubungan antar-entitas dalam sistem aplikasi jadwal pelajaran. Entitas inti terdiri dari user, guru, murid, mata pelajaran, kelas, dan jadwal pelajaran, di mana setiap kelas mendefinisikan atribut spesifik beserta metode fungsional untuk mendukung operasional sistem. Hubungan antar-kelas menunjukkan dependensi data yang sistematis, seperti asosiasi kelas user dengan guru dan murid untuk manajemen hak akses, serta relasi multiplicity antara guru, kelas, dan mata pelajaran yang beragregasi pada entitas jadwal pelajaran guna menjamin integritas referensi data.

2.3.3 Activity Diagram

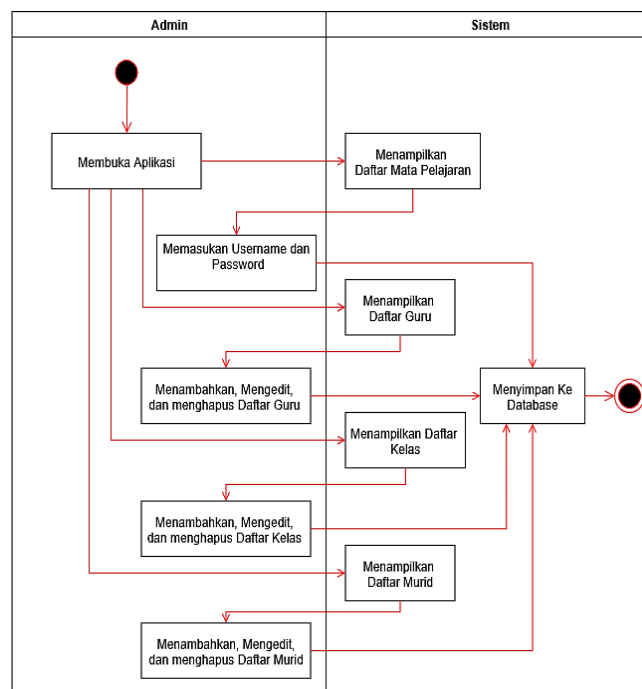
Activity diagram untuk proses login memodelkan alur kerja otentikasi pengguna secara sekuensial (Seidl et al., 2015). Pada Gambar 4 menunjukkan proses dimulai ketika pengguna (admin, guru, atau siswa) mengakses aplikasi. Sistem akan menampilkan antarmuka login yang mengharuskan pengguna memasukkan username dan password. Data credential ini kemudian diverifikasi terhadap basis data (Elmasri & Navathe, 2020). Jika valid, pengguna diarahkan ke halaman dashboard utama, sedangkan jika tidak valid, sistem menampilkan pesan kesalahan.



Gambar4. Activity Diagram Login

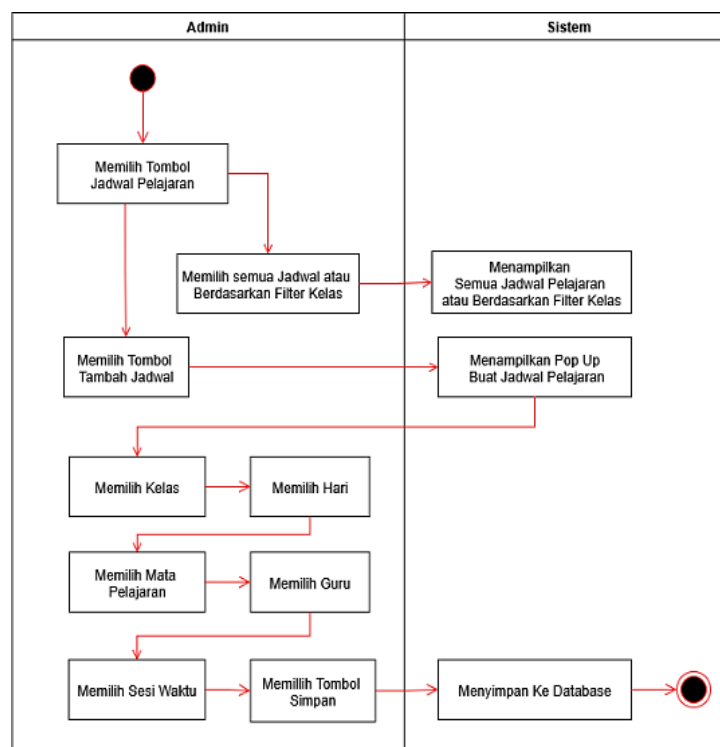
Proses ini sangat vital untuk menjamin keamanan sistem (security) dan membatasi akses hanya bagi pengguna yang memiliki hak wewenang, sesuai dengan kebutuhan non-fungsional yang telah ditetapkan. Dengan adanya autentikasi, sistem dapat memastikan bahwa data dan informasi sensitif tetap terlindungi dari akses yang tidak sah.

Activity diagram pada Gambar 5 antara admin dan sistem menggambarkan alur proses pengelolaan data utama dalam aplikasi, meliputi data mata pelajaran, guru, kelas, siswa, dan user (Yona, 2022). Proses dimulai ketika admin membuka aplikasi dan sistem menampilkan daftar data yang tersimpan di basis data. Admin dapat melakukan pengelolaan terhadap setiap entitas data, seperti menambah, mengubah, atau menghapus informasi yang diperlukan.



Gambar 5. Activity Diagram Input data

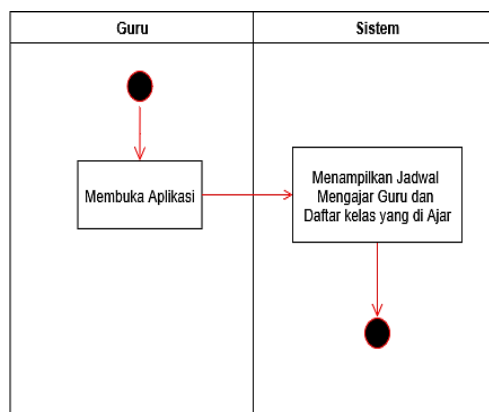
Setiap perubahan yang dilakukan akan otomatis disimpan oleh sistem ke dalam basis data untuk memastikan konsistensi dan keakuratan informasi. Diagram ini menunjukkan bahwa seluruh aktivitas pengelolaan data dilakukan secara terintegrasi dan berurutan, mulai dari mata pelajaran, guru, kelas, hingga siswa, sehingga proses administrasi jadwal pelajaran menjadi lebih efisien, terstruktur, dan mudah dikontrol oleh pihak sekolah.



Gambar 6. Activity Diagram Jadwal Pelajaran

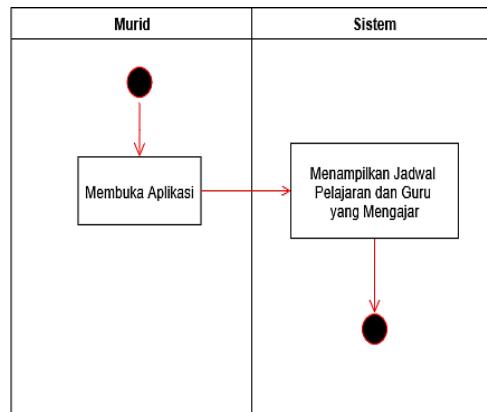
Activity diagram jadwal pelajaran pada Gambar 6 menggambarkan alur aktivitas admin dalam mengelola jadwal pelajaran melalui sistem aplikasi. Proses dimulai ketika admin membuka menu jadwal pelajaran, kemudian sistem menampilkan seluruh data jadwal yang tersimpan atau berdasarkan filter kelas tertentu sesuai permintaan pengguna.

Admin dapat menambahkan jadwal baru dengan mengisi form yang berisi informasi kelas, hari, mata pelajaran, guru pengajar, serta waktu pelajaran. Setelah seluruh data diisi dengan benar, admin menekan tombol simpan, dan sistem secara otomatis menyimpan data tersebut ke dalam basis data MySQL (Elmasri & Navathe, 2020). Diagram ini menunjukkan bahwa proses pembuatan jadwal berlangsung secara sistematis dan interaktif, sehingga memudahkan admin dalam menyusun jadwal pelajaran secara akurat dan efisien.



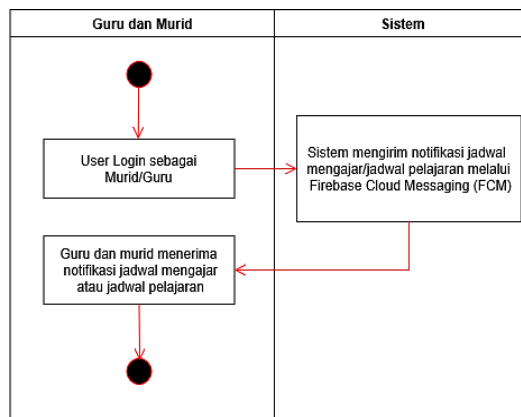
Gambar 7. Activity Diagram Guru

Activity diagram guru pada Gambar 7 memberikan gambaran alur aktivitas pengguna dengan peran guru dalam mengakses jadwal mengajar melalui aplikasi mobile. Proses dimulai ketika guru membuka aplikasi dan sistem secara otomatis menampilkan halaman utama berisi daftar jadwal mengajar yang telah ditentukan oleh admin. Sistem kemudian menampilkan informasi lengkap, seperti nama mata pelajaran, kelas, hari, dan waktu pelajaran. Melalui alur ini, guru dapat dengan mudah mengetahui jadwal mengajarnya secara real-time tanpa harus menunggu informasi dari pihak sekolah, sehingga proses perencanaan dan pelaksanaan kegiatan pembelajaran menjadi lebih efisien dan terorganisir.



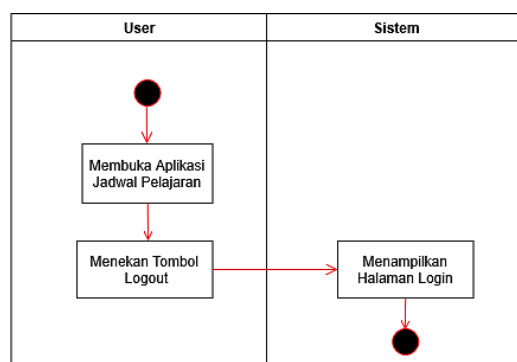
Gambar 8. Activity Diagram Murid

Activity diagram murid pada Gambar 8 memodelkan alur aktivitas siswa dalam mengakses jadwal pelajaran harian melalui aplikasi mobile. Setelah login, sistem secara otomatis akan menampilkan jadwal yang sesuai dengan kelas yang diikuti siswa tersebut. Tampilan ini mencakup informasi mata pelajaran, guru pengajar, hari, dan waktu pelajaran secara real-time. Alur ini bertujuan untuk meningkatkan kemandirian siswa dalam mempersiapkan kegiatan belajar dan memastikan informasi jadwal tersampaikan secara teratur.



Gambar 9. Activity Diagram Notifikasi

Activity Diagram pada Gambar 9 mengilustrasikan mekanisme pengiriman notifikasi jadwal yang diimplementasikan secara *asynchronous* menggunakan Firebase Cloud Messaging (FCM). Berbeda dengan proses penarikan data konvensional, sistem memanfaatkan FCM sebagai jembatan antara Node.js di sisi *backend* dan Flutter di sisi *client*. Mekanisme ini memungkinkan pengiriman *push notification* secara *real-time* kepada guru dan murid tanpa mengharuskan aplikasi dalam kondisi aktif atau terbuka. Proses diawali dengan sinkronisasi token perangkat saat autentikasi, yang kemudian digunakan oleh server untuk mendorong informasi jadwal secara otomatis berdasarkan parameter waktu yang telah ditentukan, guna menjamin ketepatan penyampaian informasi.

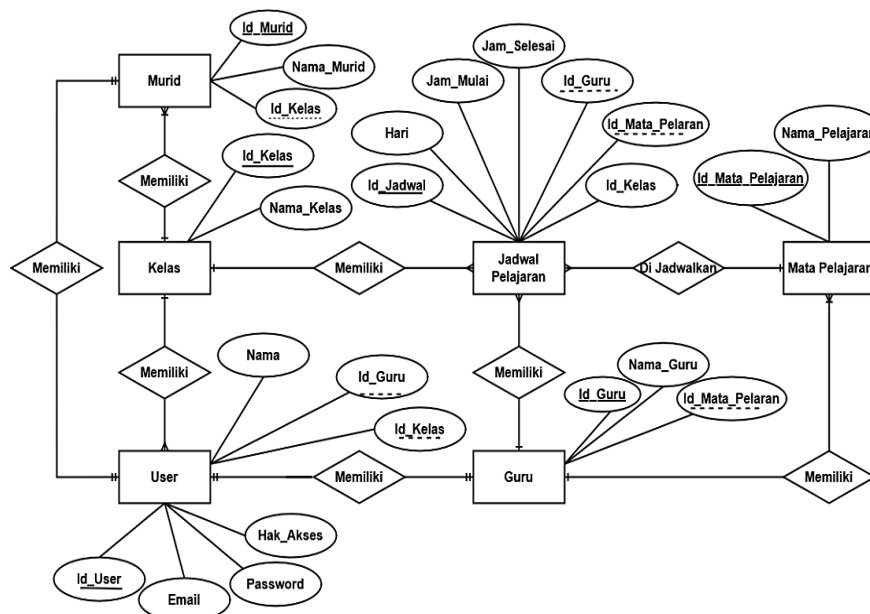


Gambar 10. Activity Diagram Logout

Alur logout merupakan bagian integral dari mekanisme keamanan sistem. Pada Gambar 10 menjelaskan proses ketika pengguna menekan tombol logout dari antarmuka aplikasi. Sistem kemudian mengakhiri sesi pengguna dan mengarahkan kembali ke halaman login. Proses ini penting untuk memutus sesi aktif dan melindungi data pengguna dari akses yang tidak sah.

2.3.4 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) pada Gambar 11 memodelkan arsitektur basis data sistem jadwal pelajaran yang mengintegrasikan enam entitas utama: User, Guru, Murid, Kelas, Mata Pelajaran, dan Jadwal Pelajaran. Entitas User berfungsi sebagai basis autentikasi yang memiliki relasi satu-ke-satu (*one-to-one*) dengan entitas Guru dan Murid guna mendefinisikan peran serta hak akses pengguna secara spesifik. Entitas Murid terikat secara struktural pada entitas Kelas, sementara entitas Guru memiliki keterkaitan langsung dengan entitas Mata Pelajaran yang mencerminkan spesialisasi pengajaran tenaga pendidik dalam sistem.

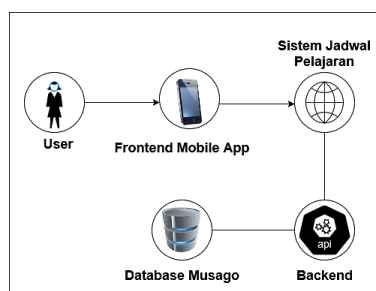


Gambar 11. Entity Relationship Diagram (ERD)

Entitas Jadwal Pelajaran berperan sebagai entitas sentral yang menghubungkan berbagai relasi fungsional melalui koordinasi atribut hari, jam mulai, dan jam selesai. Entitas ini mengonsolidasikan referensi dari entitas Kelas, Guru, dan Mata Pelajaran untuk membentuk satu kesatuan informasi kegiatan operasional sekolah. Melalui implementasi relasi kardinalitas yang tepat, rancangan ERD ini menjamin integritas referensial dan konsistensi data, sehingga proses distribusi informasi jadwal kepada pengguna melalui aplikasi mobile dapat berjalan secara akurat dan terstruktur.

2.3.5 Arsitektur Model

Arsitektur model yang diilustrasikan pada Gambar 12 merepresentasikan mekanisme interaksi berlapis antara pengguna dan seluruh komponen teknis aplikasi jadwal pelajaran. Pada lapisan terluar, *User* berinteraksi secara langsung melalui *Frontend Mobile App* yang dibangun. Komponen *frontend* ini bertindak sebagai jembatan komunikasi yang mengirimkan permintaan (*request*) kepada bagian *Backend* melalui protokol layanan API. Struktur ini memastikan bahwa logika presentasi pada sisi klien terpisah secara jelas dari logika bisnis pada sisi server, sehingga meningkatkan efisiensi proses pertukaran data secara *real-time*.



Gambar 12. Arsitektur Model

Pada komponen *Backend* berbasis Node.js mengelola pemrosesan logika, validasi data, serta integrasi dengan firebase untuk memfasilitasi pengiriman *push notifications*. Seluruh data operasional dikelola di dalam *Database Musago* yang berfungsi sebagai repositori pusat untuk informasi jadwal. Integrasi FCM memungkinkan server untuk mengirim notifikasi pengingat jadwal secara otomatis ke perangkat pengguna meskipun aplikasi tidak sedang digunakan, guna menjamin ketepatan penyampaian informasi. Koordinasi antara komponen *frontend*, *backend*, Firebase, dan basis data ini menghasilkan sistem yang terintegrasi dalam mendistribusikan jadwal pelajaran.

3. HASIL DAN PEMBAHASAN

Implementasi aplikasi jadwal pelajaran ini mengintegrasikan arsitektur full-stack modern yang menghubungkan Flutter, Node.js, dan Firebase secara sistematis sesuai dengan tahapan perancangan metode. Pada tahap pengembangan antarmuka, Flutter digunakan untuk membangun aplikasi yang adaptif bagi siswa, sementara Node.js diimplementasikan sebagai backend untuk mengelola logika bisnis dan kontrol administrasi sekolah. Data jadwal disimpan dan dikelola melalui Firebase sebagai database real-time, yang memungkinkan sinkronisasi data terjadi secara instan tanpa perlu memuat ulang aplikasi. Fitur utama dalam transformasi ini adalah sistem notifikasi otomatis yang memanfaatkan Firebase Cloud Messaging (FCM); di mana setiap kali admin melakukan perubahan jadwal pada server Node.js, sistem secara otomatis memicu pengiriman pesan ke perangkat pengguna. Melalui alur integrasi ini, metode pengelolaan jadwal konvensional berhasil dialihkan menjadi sistem dinamis yang secara otomatis meningkatkan efisiensi waktu dan kedisiplinan koordinasi di lingkungan sekolah.

Hasil implementasi menunjukkan bahwa sistem mampu menyajikan jadwal secara real-time serta mengirimkan notifikasi otomatis kepada pengguna. Berdasarkan hasil pengujian blackbox, seluruh fungsi utama sistem berjalan sesuai dengan kebutuhan yang telah didefinisikan pada tahap analisis, sehingga sistem dinilai layak untuk digunakan dalam mendukung manajemen jadwal pelajaran.

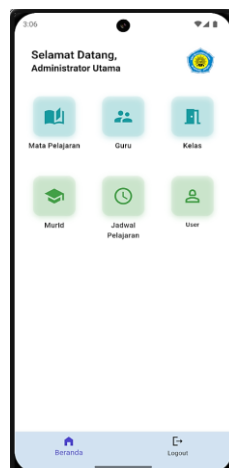
3.1 Implementasi Antarmuka

Penelitian ini menghasilkan aplikasi jadwal pelajaran berbasis Flutter dan Node.js (Express) dengan database MySQL untuk SMP Muhammadiyah Mlonggo Jepara. Aplikasi ini mendigitalisasi metode konvensional menjadi sistem yang lebih efisien, akurat, dan real-time. Dikembangkan melalui tahapan rekayasa perangkat lunak yang sistematis, aplikasi ini memastikan integrasi optimal antara antarmuka pengguna dan layanan backend.



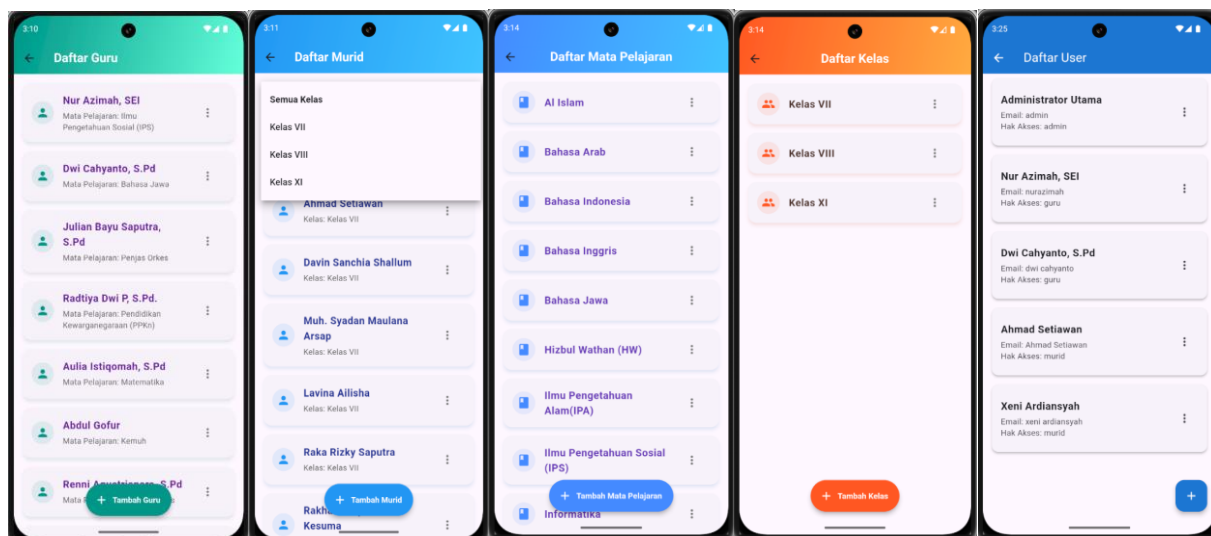
Gambar 13. Splash Screen dan Log in

Alur kerja aplikasi dimulai dengan Splash Screen berlogo sekolah, dilanjutkan ke halaman login (Gambar 13). Proses autentikasi dilakukan melalui RESTful API yang memvalidasi data pengguna pada database. Setelah validasi berhasil, sistem mengarahkan pengguna ke dashboard spesifik berdasarkan peran mereka sebagai admin, guru, atau siswa.



Gambar 14. Dashboard Admin

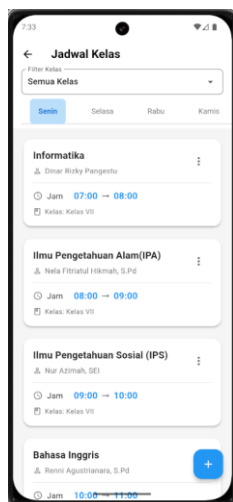
Pada Gambar 14 admin memiliki hak akses penuh untuk mengelola seluruh data sistem, termasuk data guru, siswa, kelas, mata pelajaran, pengguna, dan jadwal pelajaran. Melalui dashboard administrasi, admin dapat menambah, mengubah, atau menghapus data sesuai kebutuhan, dan setiap perubahan akan tersimpan di basis data. Proses penginputan data ini dilakukan secara terpusat guna memastikan tidak adanya duplikasi data yang dapat mengganggu performa sistem. Pengelolaan jadwal pelajaran menjadi bagian penting dari peran admin, di mana admin dapat mengatur waktu pelajaran, penjadwalan guru, serta memastikan tidak terjadi bentrok antar kelas. Selain itu, admin juga memiliki wewenang untuk memantau aktivitas login pengguna guna menjaga keamanan akun dari akses yang tidak sah. Setiap pembaruan jadwal langsung tersinkronisasi ke seluruh sistem, sehingga pengguna selalu mendapatkan informasi terkini. Dengan hak akses ini, admin berperan menjaga integritas data, kelancaran operasional aplikasi, dan efisiensi manajemen akademik di lingkungan sekolah. Keberadaan kendali penuh di tangan admin ini bertujuan untuk memberikan standarisasi pengelolaan informasi agar seluruh laporan pendidikan yang dihasilkan menjadi lebih akurat, transparan, dan dapat dipertanggungjawabkan.



Gambar 15. Manajemen Data Guru, Murid, Mata Pelajaran, Kelas, dan User

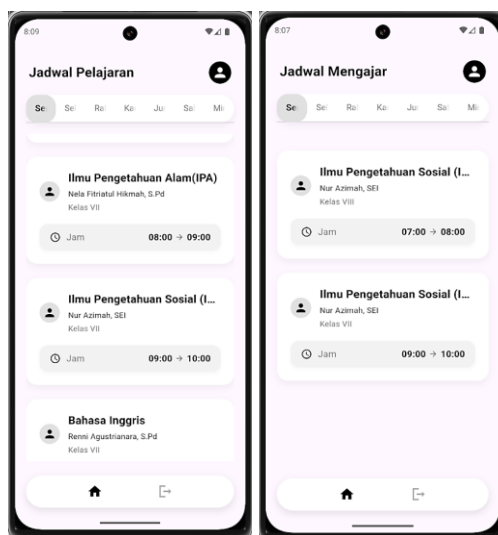
Fitur manajemen data pada Gambar 15 memungkinkan admin untuk mengelola seluruh entitas sekolah secara terpusat, mulai dari data guru, murid, mata pelajaran, hingga akun pengguna.

1. **Pengelolaan Guru dan Murid:** Pada bagian daftar guru, admin dapat memantau seluruh pengajar beserta mata pelajaran yang diampu. Sementara pada daftar murid, tersedia fitur filter per kelas (VII, VIII, dan IX) untuk memudahkan pencarian data siswa. Admin memiliki kendali penuh untuk menambah, mengedit, atau menghapus data tersebut melalui pop-up form yang terhubung langsung dengan database.
2. **Pengelolaan Mata Pelajaran dan Kelas:** Fitur ini mencakup pengaturan mata pelajaran aktif—seperti Al-Islam, Bahasa Indonesia, dan Bahasa Inggris—serta struktur kelas yang tersedia. Pengelolaan ini menjadi fondasi penting dalam penyusunan jadwal akademik agar data tetap konsisten dan terbaru di dalam sistem.
3. **Manajemen Pengguna (User):** Admin dapat membuat akun untuk guru dan murid serta mengatur hak akses bagi seluruh pengguna yang terdiri dari admin, guru, dan murid. Melalui tampilan ini, informasi detail seperti nama lengkap, email, dan peran (role) dapat dikelola secara efisien untuk menjamin keamanan akses sistem.



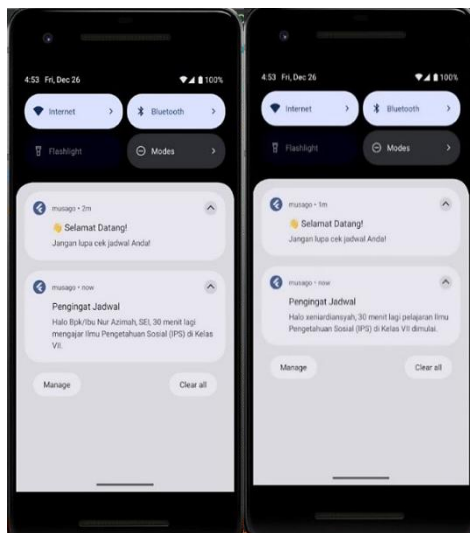
Gambar 16. Jadwal Pelajaran

Fitur jadwal kelas pada Gambar 16 berfungsi untuk memudahkan admin dalam mengelola dan memantau jadwal mata pelajaran di setiap kelas. Melalui fitur ini, admin dapat melihat daftar jadwal yang tersusun berdasarkan hari dan dapat memfilter tampilan jadwal berdasarkan kelas tertentu. Setiap jadwal menampilkan informasi lengkap seperti nama mata pelajaran, guru pengajar, waktu pelaksanaan, dan kelas yang bersangkutan. Admin juga dapat menambahkan jadwal baru dengan mengisi data kelas, hari, mata pelajaran, guru, serta sesi waktu melalui form “buat jadwal pelajaran”. Dengan adanya fitur ini, proses penjadwalan menjadi lebih terstruktur, efisien, dan mudah diperbarui tanpa harus melakukan perubahan manual pada data di sistem.



Gambar 17. Jadwal Pelajaran Murid dan Jadwal Mengajar Guru

Pada Gambar 17 menggambarkan fitur jadwal pelajaran siswa, yang dirancang agar siswa dapat melihat jadwal belajar harian dengan mudah melalui tab hari senin hingga jumat. Setiap jadwal menampilkan nama mata pelajaran, guru pengajar, serta durasi waktu pelajaran secara terstruktur. Fitur ini mempermudah siswa dalam mengatur dan memahami jadwal kelas harian mereka. Sementara itu, fitur dashboard jadwal mengajar diperuntukkan bagi guru untuk menampilkan jadwal mengajar dan kelas yang diampu, sehingga guru dapat mengakses informasi jadwal terkini, mempersiapkan kegiatan pembelajaran, serta mengelola waktu secara lebih efektif.



Gambar 18. Notifikasi Jadwal Pelajaran Murid dan Jadwal Mengajar Guru

Notifikasi yang ditampilkan pada Gambar 18 merupakan hasil implementasi fitur pengingat jadwal dalam aplikasi. pengiriman push notification pengingat jadwal yang bersifat *asynchronous* dan tidak bergantung pada kondisi aplikasi dalam keadaan aktif. Notifikasi tetap dapat diterima oleh pengguna meskipun aplikasi berada pada kondisi *background* atau tertutup. Pada sistem yang yang di buat, FCM digunakan sebagai perantara komunikasi antara *backend* berbasis Node.js dan aplikasi mobile berbasis Flutter. Proses diawali dari sisi server ketika admin melakukan pengolahan data jadwal pelajaran pada sistem backend. Berdasarkan data waktu pelaksanaan jadwal yang tersimpan di basis data, server secara otomatis menjadwalkan pengiriman notifikasi melalui FCM untuk dikirimkan 30 menit sebelum jadwal pelajaran atau jadwal mengajar dimulai. Mekanisme ini memastikan pengguna memperoleh pengingat lebih awal.



3.2 Pengujian Black Box

Pengujian dilakukan menggunakan metode blackbox testing, yaitu dengan memberikan input dan mengamati output tanpa mengetahui struktur atau kode program di dalam sistem. Metode ini digunakan untuk memastikan bahwa setiap fitur pada aplikasi jadwal pelajaran berfungsi sesuai dengan kebutuhan dan menghasilkan keluaran yang diharapkan. Pengujian dilakukan terhadap seluruh fitur utama aplikasi yang dapat diakses oleh admin, guru, dan murid. Hasil pengujian menunjukkan bahwa seluruh fungsi aplikasi berjalan dengan baik sesuai skenario uji yang telah direncanakan. Adapun hasil pengujian setiap fitur ditunjukkan pada tabel berikut:

Tabel 1. *Blackbox Testing*

Fitur	Skenario Uji	Input	Output
Login admin	Masukan username dan password	Username: admin password: admin123	Berhasil masuk ke dashboard admin
Login guru	Masukan username dan password	Username: nurazimah password: nurazimah123	Berhasil masuk ke halaman dashboard guru
Login murid	Masukan username dan password	Username: xeniardiansyah password: xeni123	Berhasil masuk ke halaman dashboard murid
Daftar guru	Melihat daftar guru, menambahkan guru, edit data	Input nama guru, input mata pelajaran	Admin berhasil menambahkan data guru, berhasil edit data guru, berhasil hapus data guru
Daftar mata pelajaran	Melihat daftar mata pelajaran, menambahkan mata pelajaran, edit data mata pelajaran, hapus data matapelajaran	Input mata pelajaran baru	sistem berhasil menampilkan daftar mata pelajaran, berhasil menambahkan mata pelajaran baru, berhasil mengedit data mata pelajaran, berhasil menghapus mata pelajaran
Daftar kelas	Melihat daftar kelas, menambahkan, mengedit dan menghapus daftar kelas	Input nama kelas baru	Admin berhasil menampilkan, menambahkan, mengedit dan menghapus data kelas
Daftar murid	Melihat daftar murid, menamb-ahkan, mengedit, menghapus data murid	Input nama murid baru dan pilih kelas	Admin berhasil menampilkan, menamb-ahkan, mengedit dan menghapus daftar murid
Membuat jadwal pelajaran	Melihat semua jadwal pelajaran, melihat jadwal pelajaran berdasar-kan filter, Menambahkan, mengedit, dan menghapus jadwal pelajaran	Input kelas, hari, mata pelajaran, guru/pen-gajar, sesi waktu	Admin berhasil menampilkan jadwal pelajaran, menampilkan jadwal pelajaran berdasarkan filter,berhasil menambahkan, mengedit, dan menghapus jadwal pelajaran
Tambah User	Melihat daftar user, menambahkan, mengedit dan menghapus daftar user	Input hak akses (admin, guru, murid), Input nama lengkap, email/username, password, input pilih guru, input pilih kelas	Admin berhasil melihat daftar user, menamb-ahkan user guru (hak akses guru), menamb-ahkan user murid (hak akses murid), menamb-ahkan user admin (hak akses admin), mengedit dan menghapus user
Notifikasi	Sistem menampilkan notifikasi otomatis		Sistem Berhasil menampilkan notifikasi otomatis
Log-out	Logout user admin, murid, dan guru		Berhasil logout dan kembali ke halaman login

Berdasarkan hasil pengujian black-box pada Tabel 1, seluruh fitur utama aplikasi mulai dari autentikasi multi-level, pengelolaan data induk (CRUD), hingga mekanisme pembuatan jadwal telah berfungsi valid dengan tingkat keberhasilan 100%. Integrasi antara frontend Flutter dan backend Node.js terbukti berjalan sinkron, ditandai dengan pengiriman notifikasi jadwal secara real-time serta akurasi sistem validasi dalam menangani input data. Tidak ditemukan bug mayor selama pengujian, dan sistem memberikan pesan kesalahan yang tepat saat menerima data tidak valid. Secara keseluruhan, hasil ini menegaskan bahwa aplikasi telah memenuhi kebutuhan fungsional pengguna dan siap diimplementasikan untuk meningkatkan efisiensi operasional manajemen jadwal di SMP Muhammadiyah Mlonggo Jepara.



4. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa pengembangan aplikasi jadwal pelajaran berbasis mobile menggunakan Flutter dan Node.js di SMP Muhammadiyah Mlonggo Jepara berhasil menjawab permasalahan utama pada sistem penjadwalan konvensional. Sistem yang dikembangkan mampu mengintegrasikan aplikasi mobile dengan backend berbasis RESTful API serta basis data MySQL secara terpusat, sehingga mendukung pengelolaan data jadwal yang lebih terstruktur, konsisten, dan mudah diperbarui. Keberadaan mekanisme deteksi konflik dalam proses pengelolaan jadwal juga berperan penting dalam mencegah terjadinya bentrok jadwal mengajar maupun jadwal pelajaran, sehingga meningkatkan akurasi informasi yang disampaikan kepada pengguna. Hasil pengujian fungsional menggunakan metode Black-box Testing menunjukkan bahwa seluruh fitur utama sistem, termasuk autentikasi pengguna, pengelolaan data akademik, penyajian jadwal secara real-time, serta notifikasi pengingat berbasis push notification, berjalan sesuai dengan spesifikasi yang telah ditetapkan. Implementasi notifikasi yang terintegrasi dengan perubahan data pada server memungkinkan guru dan siswa memperoleh informasi terbaru secara proaktif, termasuk pengingat sebelum kegiatan belajar mengajar dimulai. Dengan demikian, aplikasi ini terbukti mampu meningkatkan efektivitas koordinasi dan mengurangi risiko keterlambatan dan meningkatkan kesiapan pengguna dalam kegiatan belajar mengajar. Meskipun demikian, penelitian ini masih memiliki beberapa keterbatasan. Sistem yang dikembangkan belum menyediakan fitur auto-scheduling atau penyusunan jadwal secara otomatis berbasis algoritma optimasi, sehingga proses pembuatan jadwal awal masih memerlukan intervensi admin. Selain itu, pengujian sistem masih terbatas pada satu lingkungan sekolah dengan jumlah pengguna yang relatif kecil. Oleh karena itu, penelitian selanjutnya disarankan untuk mengembangkan fitur auto-scheduling guna meningkatkan efisiensi manajemen jadwal, memperluas skala implementasi pada lebih banyak sekolah, serta mengeksplorasi penerapan algoritma optimasi atau kecerdasan buatan agar sistem tidak hanya berfungsi sebagai media diseminasi informasi, tetapi juga sebagai alat pendukung pengambilan keputusan dalam penjadwalan akademik.

REFERENCES

- Bontekoe, Tj. R., Kester, D., & Skilling, J. (2006). Scheduling for schools. *AIP Conference Proceedings*, 872(1), 525–532. <https://doi.org/10.1063/1.2423315>
- Budiman, A., Alhamidi, A., Iswandy, E., & Asmara, R. (2021). Aplikasi manajemen jadwal mengajar guru sekolah menengah atas. *Simtika*, 4(2), 40–46. <https://ejournal.undhari.ac.id/index.php/simtika/article/view/354>
- Chiandra, E. M., & Laksana, R. P. (2024). Pengembangan aplikasi penjadwalan digital untuk sekolah Bhinneka Tunggal Ika berbasis mobile. *Journal of Computer Science and Information Technology*, 2(1), 1–14. <https://doi.org/10.70248/jcsit.v2i1.1297>
- Elmasri, R., & Navathe, S. B. (2015). *Fundamentals of Database Systems*. Pearson Education. <https://books.google.co.id/books?id=tRyBCgAAQBAJ>
- Gupta, K., & Goel, S. (2024). Integration of technology into education and its effects on student learning. *International Journal for Multidisciplinary Research*, 6(2). <https://doi.org/10.36948/ijfmr.2024.v06i02.15586>
- Hafids, M. A., Setiawan, H. S., & Dhika, H. (2022). Rancangan aplikasi jadwal pelajaran berbasis Java di sekolah SMAN 16 Bekasi. *Jurnal PkM Pengabdian kepada Masyarakat*, 5(4), 408–414. <https://doi.org/10.30998/jurnalpkm.v5i4.11257>
- Hananto, Anshari Anjas and , Dedi Gunawan, Ph.D (2022) *Aplikasi Penjadwalan Dan Catatan Sekolah Berbasis Android Di Smp Negeri 1 Tasikmadu*. Skripsi thesis, Universitas Muhammadiyah Surakarta.
- Hussain, A., Mkpojiogu, E. O., & Babalola, E. T. (2020). Using mobile educational apps to foster work and play in learning: A systematic review. *International Journal of Interactive Mobile Technologies (IJIM)*, 14(18), 178–193. <https://doi.org/10.3991/ijim.v14i18.16619>
- Oktarina, F. K., & Nasari, N. F. (2024). Analisis dan pemodelan proses bisnis menggunakan UML untuk mengidentifikasi sistem RPL Politeknik Kampar. *Jurnal CoSciTech (Computer Science and Information Technology)*, 5(3), 705–713. <https://doi.org/10.37859/coscitech.v5i3.7951>
- Pangestu, P. O. (2021). Aplikasi penjadwalan kegiatan belajar mengajar pada SMK Budi Asih berbasis Java. *Seminar Nasional Riset dan Inovasi Teknologi*, 5(1). <https://doi.org/10.30998/semnasristek.v5i1.4937>
- Prasasti, I. H., & Wahyudi, A. (2025). Pemanfaatan aplikasi mobile learning untuk meningkatkan literasi digital guru dan siswa. *Journal of Computer Science and Informatics (JOCSI)*, 2(2), 73–76. <https://doi.org/10.69747/jocsi.v2i2.83>
- Pratiwi, I., Anardani, S., & Putera, A. R. (2023). Rancang bangun sistem informasi penjadwalan mata pelajaran dengan metode waterfall. *JDMIS: Journal of Data Mining and Information Systems*, 1(1), 20–28. <https://doi.org/10.54259/jdmis.v1i1.1513>
- Pressman, R. S., & Maxim, B. R. (2019). *Software Engineering: A Practitioners Approach*. McGraw-Hill Education. <https://books.google.co.id/books?id=taIKxAEACAAJ>
- Qottrunnada, A., Ikhsan, S. H. A., & Eosina, P. (2025). Rancang bangun sistem manajemen jadwal guru pada MTs Sirojul Ikhsan berbasis mobile. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(5), 7497–7503. <https://doi.org/10.3640/jati.v9i5.14730>



- Royce, W. (2021). Managing the development of large software systems: concepts and techniques. *Ideas That Created the Future*. <https://api.semanticscholar.org/CorpusID:110643636>
- Safitri, & Puspitorini, S. (2021). Sistem informasi penjadwalan guru pada SMP Negeri 9 Muaro Jambi berbasis web. *Jurnal Karya Informatika (KARTIKA)*, 2(2), 13–18. <https://ojs.unh.ac.id/index.php/kartika/article/download/680/535>
- Seidl, M., Scholz, M., Huemer, C., & Kappel, G. (2015). The activity diagram. In M. Seidl, M. Scholz, C. Huemer, & G. Kappel (Eds.), *UML @ Classroom: An introduction to object-oriented modeling* (pp. 141–166). Springer International Publishing. https://doi.org/10.1007/978-3-319-12742-2_7
- Singh, K. J., & Lodhe, N. (2021). Cross platform development using Flutter. *VIVA-Tech International Journal for Research and Innovation*, 1(4), 1–6. <http://www.viva-technology.org/New/IJRI/2021/152.pdf>
- Sommerville, I. (2016). *Software Engineering*. Pearson. <https://books.google.co.id/books?id=tW4VngEACAAJ>
- Yona, T. (2022). Sistem penjadwalan mata pelajaran sekolah menggunakan metode pengembangan perangkat lunak model waterfall di SMP Negeri 2 Sumbergempol. *Indonesian Journal of Dedication and Educations*, 1(1), 1–5. <https://doi.org/10.33019/ijde.v1i1.1>