

Rancang Bangun Aplikasi Data Guru Berbasis React Native

Melly Jayanti¹, Iwan Purnama²

¹Fakultas Sains dan Teknologi, Program Studi Sistem Informasi, Universitas Labuhanbatu, Labuhanbatu, Indonesia

Email: ¹mellyjayanti2022@gmail.com, ²iwanpurnama2014@ulb.ac.id

Abstrak– Aplikasi *Teacher Management* ini dirancang untuk mempermudah pengelolaan data guru secara efisien dan terstruktur. Aplikasi ini dilengkapi dengan fitur tambah, edit, hapus, dan pencarian data, serta dirancang dengan tampilan yang sederhana dan mudah digunakan, sehingga mampu mendukung kinerja admin. Struktur proyek React Native dibuat secara modular untuk mempermudah pengembangan, dengan folder *src* berisi komponen UI, layout, dan tipe data. Komponen seperti *TeacherDetail*, *TeacherForm*, dan *TeacherList* dipisahkan untuk menjaga keteraturan. Sementara itu, API menggunakan PHP Native dengan struktur yang menangani operasi CRUD pada data guru, di mana folder *api/teachers* berisi file *create.php*, *read.php*, *read_one.php*, *update.php*, dan *delete.php*. Ke depannya, aplikasi ini akan ditambahkan fitur validasi, notifikasi, konfirmasi hapus, ekspor data, pagination, dan sistem login untuk meningkatkan fungsionalitas dan keamanan. Pengembangan fitur lanjutan ini akan semakin meningkatkan kemudahan, keamanan, dan keandalan sistem dalam mengelola informasi guru.

Kata Kunci: *Teacher Management*, CRUD, React Native, PHP Native, Pengelolaan Data Guru

Abstract – The *Teacher Management* application is designed to facilitate efficient and structured teacher data management. The application includes features such as add, edit, delete, and search, and is built with a simple and user-friendly interface to support admin performance. The React Native project structure is organized modularly to ease development, with the *src* folder containing UI components, layouts, and data types. Components such as *TeacherDetail*, *TeacherForm*, and *TeacherList* are separated to maintain organization. On the backend, the API uses PHP Native to handle CRUD operations for teacher data, with the *api/teachers* folder containing *create.php*, *read.php*, *read_one.php*, *update.php*, and *delete.php*. Future development will include features such as validation, notifications, delete confirmations, data export, pagination, and a login system to enhance functionality and security. These improvements aim to further increase the ease of use, security, and reliability of the system in managing teacher information.

Keywords: *Teacher Management*, CRUD, React Native, PHP Native, Teacher Data Management

1. PENDAHULUAN

Dalam lingkungan pendidikan modern, pengelolaan data guru yang efisien menjadi landasan krusial bagi operasional institusi. Ketersediaan informasi guru yang akurat dan mudah diakses tidak hanya menopang kegiatan administratif sehari-hari, tetapi juga berperan penting dalam pengambilan keputusan strategis terkait sumber daya manusia di lingkungan pendidikan. Namun, banyak institusi pendidikan, terutama di negara berkembang, masih menghadapi tantangan signifikan dalam mengelola data guru. Tantangan ini seringkali berakar pada ketergantungan terhadap proses manual yang memakan waktu, rentan terhadap kesalahan input, dan minim integrasi sistem. Misalnya, pencatatan data guru secara manual dalam lembar kerja atau dokumen fisik dapat menyebabkan duplikasi data, ketidaksesuaian informasi, dan kesulitan dalam melakukan pembaruan secara real-time. Keterbatasan ini secara langsung menghambat kinerja administrasi, memperlambat proses kepegawaian, dan menyulitkan manajemen dalam mengakses data relevan untuk keperluan perencanaan dan evaluasi.

Menyikapi permasalahan tersebut, dibutuhkan solusi berbasis teknologi informasi yang mampu mengotomatisasi proses, meningkatkan akurasi data, serta menyederhanakan akses dan pengelolaan informasi. Salah satu pendekatan yang relevan adalah pengembangan aplikasi berbasis mobile yang memungkinkan pengguna, khususnya pihak administrasi sekolah, untuk mengelola data guru secara efisien kapan saja dan di mana saja. Solusi ini harus mampu mengintegrasikan fitur-fitur dasar seperti Create, Read, Update, dan Delete (CRUD) dengan antarmuka pengguna (user interface) yang sederhana, intuitif, dan mudah digunakan oleh pengguna non-teknis.

Penelitian ini mengusulkan pengembangan aplikasi *Teacher Management* berbasis mobile menggunakan React Native sebagai kerangka kerja (framework) untuk pengembangan antarmuka dan PHP Native untuk pengelolaan data di sisi server. React Native dipilih karena mendukung pengembangan lintas platform dari satu basis kode, sehingga mempercepat proses pengembangan dan pemeliharaan. Di sisi lain, PHP Native digunakan untuk menangani operasi CRUD pada data guru yang terhubung dengan basis data MySQL, memastikan komunikasi antara aplikasi dan server berjalan optimal.

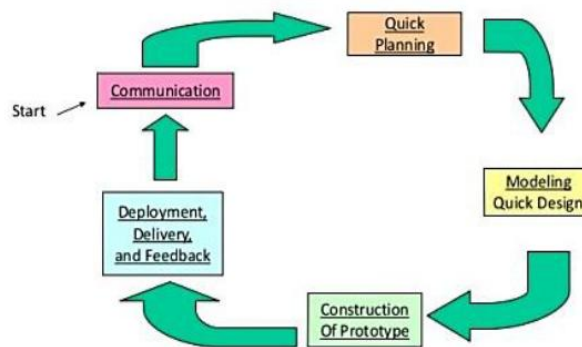
Beberapa penelitian sebelumnya telah mengangkat topik serupa. Misalnya, Axel Berkati et al. [1] rancang bangun aplikasi absensi guru dan staf tu dengan penerapan geolocation dan fingerprint berbasis android di smk gke mandomai, Endi Gunawan et al. [2] sistem informasi pendataan guru dan siswa pada sd negeri keraton 3 martapura berbasis web, Egga Asoka et al. [3] penggunaan framework react native dalam perancangan aplikasi penjualan goodday garden, Muhammad Abdul Karim et al. [4] analisis dan perancangan aplikasi mobile untuk donasi menggunakan metode hybrid berbasis react native, dan Herry Adi Chandra et al. [5][6][7][8][9] metode cepat pembuatan aplikasi web bootstrap akademik pada smk kota banjarmasin berbasis codeigniter. Berdasarkan studi-studi tersebut, terdapat gap yang menjadi fokus dalam penelitian ini, yakni ketiadaan aplikasi *Teacher Management* berbasis mobile cross-platform dengan struktur modular yang memisahkan komponen seperti *TeacherDetail*, *TeacherForm*, dan *TeacherList*, serta didukung oleh backend PHP Native yang ringan dan mudah dikembangkan. Selain itu, kebanyakan penelitian belum menyoroti pentingnya pengalaman pengguna dalam hal kesederhanaan antarmuka dan kemudahan adaptasi oleh staf administrasi non-teknis.

Dengan demikian, tujuan dari penelitian ini adalah untuk mengembangkan aplikasi Teacher Management berbasis React Native dan PHP Native yang mampu mendukung pengelolaan data guru secara efisien, akurat, dan mudah digunakan. Diharapkan aplikasi ini dapat menjadi solusi praktis dan terjangkau bagi institusi pendidikan, terutama di wilayah yang memiliki keterbatasan dalam hal sumber daya teknologi. Hasil dari penelitian ini juga diharapkan dapat menjadi dasar pengembangan sistem serupa untuk pengelolaan data pendidikan lainnya, seperti data siswa, jadwal, dan aktivitas belajar.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Metodologi penelitian ini akan mengadopsi pendekatan prototyping untuk pengembangan Aplikasi CRUD Data Guru. Pendekatan prototyping dipilih karena sifatnya yang iteratif dan adaptif, memungkinkan umpan balik berkelanjutan dari pengguna dan pemangku kepentingan sepanjang siklus pengembangan. Ini sangat penting untuk memastikan bahwa aplikasi yang dihasilkan tidak hanya fungsional tetapi juga memenuhi kebutuhan riil pengguna akhir. Proses ini melibatkan pembangunan prototipe awal yang cepat, pengujian oleh pengguna, pengumpulan umpan balik, dan revisi prototipe secara berulang hingga tercapai kesesuaian yang optimal antara aplikasi dan kebutuhan pengguna.



Gambar 2.1. Tahapan Metode *Prototyping*

2.2 Tahapan Penelitian dengan Model *Prototyping*

Tahapan penelitian dimulai dengan fase perancangan sistem awal, yang merupakan bagian dari tahap Quick Plan dalam model prototyping. Pada fase ini, analisis kebutuhan fungsional dan non-fungsional dari aplikasi Teacher Management dilakukan secara cepat namun komprehensif. Kebutuhan fungsional mencakup kemampuan dasar seperti menambah, membaca (menampilkan), memperbarui, dan menghapus data guru, serta fitur pencarian data. Kebutuhan non-fungsional meliputi aspek kinerja (kecepatan respons), keamanan (otorisasi akses), dan kemudahan penggunaan (user-friendliness). Berdasarkan analisis ini, diputuskan untuk mengadopsi arsitektur klien-server, di mana frontend akan dikembangkan menggunakan React Native dan backend API akan dibangun dengan PHP Native. Pemilihan React Native didasarkan pada kemampuannya untuk membangun aplikasi mobile lintas platform (iOS dan Android) dari satu basis kode, yang sangat menghemat waktu dan sumber daya. Sementara itu, PHP Native dipilih untuk backend karena kemudahannya dalam membangun RESTful API yang efisien dan integrasi yang baik dengan database relasional seperti MySQL, yang umum digunakan dalam pengembangan aplikasi berbasis web dan mobile. Setelah perancangan awal, masuk ke tahapan Modeling Quick Design. Pada fase ini, desain fungsionalitas dan antarmuka pengguna aplikasi mulai divisualisasikan. Ini mencakup perancangan struktur database untuk menyimpan data guru, serta desain awal antarmuka pengguna untuk halaman dashboard, formulir pendaftaran guru baru, dan formulir edit guru. Meskipun ini adalah desain awal, tujuannya adalah untuk memberikan gambaran yang cukup jelas tentang bagaimana aplikasi akan terlihat dan berfungsi.

Langkah selanjutnya adalah Construction of Prototype, di mana prototipe awal aplikasi dibangun berdasarkan desain cepat yang telah dibuat. Pada sisi frontend, struktur proyek React Native dibuat secara modular, dengan folder src yang berisi komponen UI, layout, serta data dan tipe data. Komponen seperti TeacherDetail, TeacherForm, dan TeacherList dipisahkan untuk menjaga keteraturan dan mempermudah pengembangan. File App.tsx, main.tsx, dan index.css menjadi entry point aplikasi. Pada sisi backend, API PHP Native dibangun untuk menangani operasi CRUD pada data guru. Folder api/teachers berisi file create.php, read.php, read_one.php, update.php, dan delete.php, masing-masing bertanggung jawab untuk menambahkan, membaca semua data, membaca satu data, memperbarui, dan menghapus data guru.

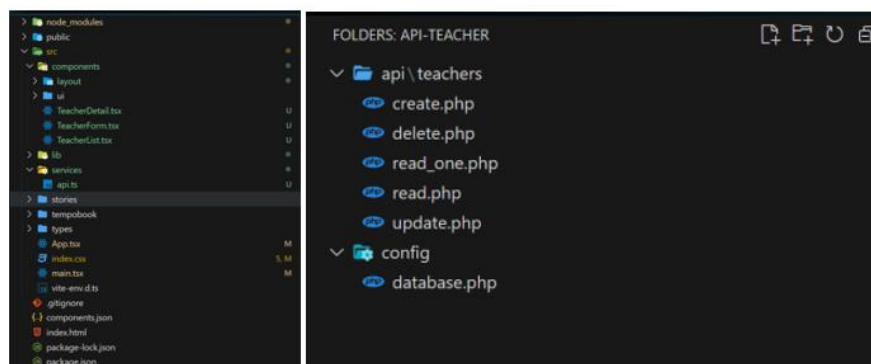
Sementara itu, folder config menyimpan database.php yang berfungsi mengelola koneksi ke database. Prototipe ini dibangun dengan cepat untuk mendapatkan model kerja yang dapat segera dievaluasi. Tahapan kritis dalam model prototyping adalah Deployment, Delivery, and Feedback. Prototipe yang telah dibangun kemudian diuji secara

langsung oleh pengguna atau stakeholder. Pada fase ini, pengguna akan berinteraksi dengan aplikasi, mencoba semua fitur yang tersedia, dan memberikan umpan balik mengenai fungsionalitas, kemudahan penggunaan, serta hal-hal yang perlu diperbaiki atau ditambahkan. Umpan balik ini sangat berharga karena mengidentifikasi potensi kesalahan, ketidaksesuaian dengan kebutuhan riil, atau area yang dapat ditingkatkan. Misalnya, pengguna mungkin memberikan masukan tentang tata letak tombol edit dan hapus pada dashboard, atau tentang kelengkapan kolom pada formulir pendaftaran guru baru.

Berdasarkan umpan balik yang terkumpul, tim pengembang akan melakukan Refinement atau perbaikan terhadap prototipe. Proses ini bersifat iteratif; jika umpan balik menunjukkan kebutuhan akan perubahan signifikan, siklus perancangan ulang, pembangunan prototipe baru, dan pengujian akan diulang hingga aplikasi memenuhi harapan pengguna. Misalnya, jika ada permintaan untuk penambahan fitur validasi input, notifikasi, atau konfirmasi hapus, fitur-fitur ini akan diimplementasikan pada iterasi berikutnya. Tujuan dari iterasi ini adalah untuk secara bertahap menyempurnakan aplikasi hingga mencapai versi final yang stabil dan sesuai dengan semua kebutuhan.

2.3 Struktur Folder

Struktur folder aplikasi ini dirancang dengan mempertimbangkan prinsip modularitas dan pemisahan tanggung jawab, baik untuk frontend berbasis React Native maupun backend berbasis PHP Native. Pemisahan ini bertujuan untuk mempermudah pengembangan secara terpisah, kolaborasi antar pengembang, dan pemeliharaan kode di masa depan. Pada sisi frontend, proyek React Native mengadopsi struktur yang sangat modular untuk memastikan keteraturan dan efisiensi dalam pengembangan. Folder `src` merupakan direktori utama yang menampung seluruh kode sumber aplikasi. Di dalam `src`, terdapat sub-folder untuk komponen UI (components), layout (layouts), serta file-file yang mendefinisikan struktur data (data) dan tipe data (types). Contoh komponen yang terpisah adalah `TeacherDetail` untuk menampilkan detail satu guru, `TeacherForm` untuk menangani formulir pendaftaran atau pengeditan guru, dan `TeacherList` untuk menampilkan daftar guru. Pemisahan komponen ini memungkinkan pengembang untuk bekerja pada bagian-bagian aplikasi secara independen dan mengintegrasikannya dengan mulus. File-file seperti `App.tsx`, `main.tsx`, dan `index.css` berfungsi sebagai entry point aplikasi, mengelola alur utama dan styling global. Struktur ini mendukung skalabilitas, mempermudah penambahan fitur baru, dan meningkatkan pemeliharaan kode karena setiap bagian memiliki tanggung jawab yang jelas.



Gambar 2.2 Struktur Folder *frontend* dan *backend*

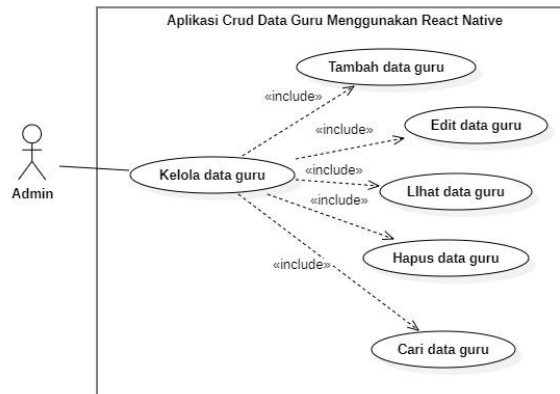
3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil implementasi dan pembahasan mendalam mengenai aplikasi *Teacher Management*, yang dikembangkan menggunakan React Native untuk frontend dan PHP Native untuk backend. Pembahasan akan mencakup detail fungsionalitas utama aplikasi, tampilan antarmuka pengguna, serta fitur-fitur yang telah diimplementasikan dan rencana pengembangan di masa mendatang. Fokus utama adalah bagaimana aplikasi ini secara efektif mempermudah pengelolaan data guru dan mendukung kinerja administratif. Bagian ini akan menguraikan setiap komponen aplikasi secara terperinci, disertai dengan justifikasi desain dan implikasi fungsionalnya.

3.1 Modeling Quick Design

Dalam penelitian ini, Unified Modeling Language (UML) dimanfaatkan sebagai alat bantu untuk merancang aplikasi yang akan dikembangkan. UML merupakan sebuah teknik pemodelan visual yang umum digunakan dalam sistem berbasis objek. Secara umum, UML dapat didefinisikan sebagai bahasa standar yang digunakan untuk memvisualisasikan, merancang, serta mendokumentasikan sistem perangkat lunak. Saat ini, UML telah menjadi standar dalam membuat cetak biru (blueprint) pengembangan perangkat lunak. Salah satu jenis diagram dalam UML adalah use case diagram, yang menggambarkan cara sistem digunakan dari perspektif pengguna. Dengan menyusun

skenario interaksi antara pengguna dan sistem, use case membantu mendeskripsikan perilaku sistem secara umum. [6] Use case diagram menunjukkan berbagai tindakan atau fungsionalitas yang dapat dilakukan oleh aktor terhadap sistem yang dirancang. Berikut Gambar 3.1 menampilkan Use Case Diagram:



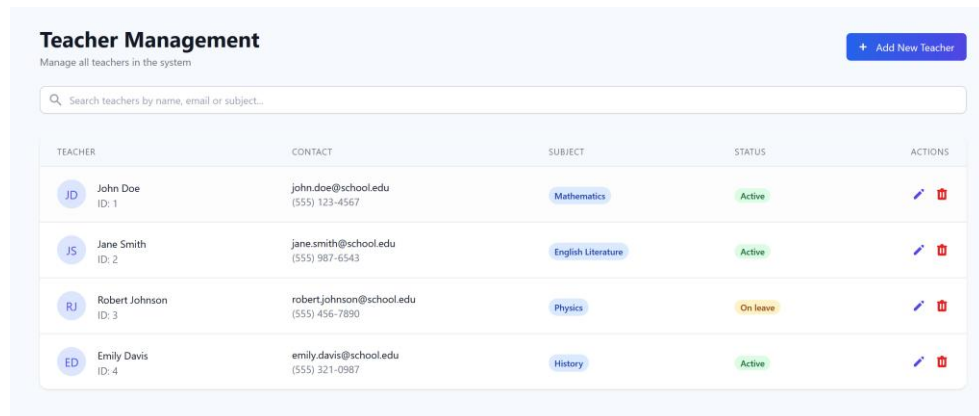
Gambar 3.1 Use Case diagram

Berikut merupakan UI (User Interface):

a. Dashboard (Halaman *Teacher Management*)

Halaman *Teacher Management*, yang berfungsi sebagai dashboard utama aplikasi, dirancang untuk menjadi pusat kendali bagi admin dalam mengelola data guru. Tampilan ini didesain secara minimalis namun informatif, dengan fokus pada kemudahan navigasi dan aksesibilitas data. Elemen paling menonjol pada halaman ini adalah tabel daftar guru, yang menyajikan ringkasan informasi esensial untuk setiap guru. Kolom-kolom dalam tabel ini mencakup nama lengkap guru, informasi kontak (seperti email dan nomor telepon), mata pelajaran yang diampu, serta status kepegawaian (misalnya, tetap, kontrak, atau honorer). Penyajian data dalam bentuk tabel ini memungkinkan admin untuk dengan cepat memindai dan memahami status keseluruhan data guru yang ada di sistem.

Untuk meningkatkan efisiensi pengelolaan data, halaman dashboard dilengkapi dengan fitur pencarian yang canggih. Fitur ini memungkinkan admin untuk mencari data guru berdasarkan berbagai kriteria, seperti nama, mata pelajaran, atau status kepegawaian. Implementasi fitur pencarian ini sangat krusial dalam sistem dengan volume data yang besar, karena dapat mengurangi waktu yang dibutuhkan untuk menemukan informasi spesifik dan meningkatkan responsivitas aplikasi. Di samping setiap baris data guru dalam tabel, terdapat tombol aksi yang esensial: "Edit" dan "Hapus". Tombol "Edit" mengarahkan admin ke formulir pengeditan data guru, memungkinkan mereka untuk memperbarui informasi yang sudah ada. Sementara itu, tombol "Hapus" berfungsi untuk menghapus data guru dari sistem. Desain tombol aksi ini dibuat intuitif untuk mencegah kesalahan dan memastikan bahwa operasi penting dapat dilakukan dengan mudah. Selain itu, untuk memfasilitasi penambahan data baru, terdapat tombol "Add New Teacher" yang menonjol. Tombol ini mengarahkan admin ke formulir pendaftaran guru baru, di mana mereka dapat memasukkan detail guru yang belum tercatat dalam sistem. Kehadiran tombol ini di dashboard utama menekankan kemudahan akses untuk proses input data, yang merupakan salah satu fungsi inti dari aplikasi CRUD. Keseluruhan tampilan halaman ini dirancang agar sederhana dan mudah digunakan, meminimalkan kurva pembelajaran bagi admin dan memastikan bahwa mereka dapat fokus pada tugas pengelolaan data tanpa terbebani oleh antarmuka yang rumit. Desain yang responsif juga dipertimbangkan agar dashboard dapat diakses dengan baik di berbagai ukuran layar, dari desktop hingga perangkat mobile, meskipun fokus utama aplikasi ini adalah manajemen data secara efisien. Dengan demikian, halaman *Teacher Management* ini tidak hanya berfungsi sebagai tampilan data, tetapi juga sebagai pusat interaksi yang dinamis antara admin dan sistem.



TEACHER	CONTACT	SUBJECT	STATUS	ACTIONS
John Doe ID: 1	john.doe@school.edu (555) 123-4567	Mathematics	Active	
Jane Smith ID: 2	jane.smith@school.edu (555) 987-6543	English Literature	Active	
Robert Johnson ID: 3	robert.johnson@school.edu (555) 456-7890	Physics	On leave	
Emily Davis ID: 4	emily.davis@school.edu (555) 321-0987	History	Active	

Gambar 3.2. Halaman Dashboard

Penggunaan React Native untuk mengembangkan frontend halaman dashboard memberikan beberapa keuntungan signifikan. Pertama, performa yang cepat dan responsif, karena React Native mengkompilasi kode JavaScript ke komponen UI native, menghasilkan pengalaman pengguna yang serupa dengan aplikasi native murni. Kedua, kemampuan untuk membangun aplikasi lintas platform dari satu basis kode sangat menghemat waktu dan sumber daya pengembangan. Hal ini memastikan konsistensi tampilan dan fungsionalitas di perangkat iOS dan Android, yang penting untuk menjangkau pengguna dengan perangkat yang beragam. Ketiga, modularitas React Native memungkinkan pengembangan untuk membuat komponen UI yang dapat digunakan kembali, seperti tabel data dan tombol, yang mempercepat proses pengembangan dan pemeliharaan. Misalnya, komponen `TeacherList` dapat digunakan untuk menampilkan data guru, dan di dalamnya terdapat sub-komponen untuk setiap baris guru dengan tombol edit dan hapus. Struktur ini mempermudah skalabilitas proyek, memungkinkan penambahan fitur baru dengan relatif mudah di masa mendatang.

b. Form Pendaftaran Guru Baru

Halaman Form Pendaftaran Guru Baru merupakan salah satu inti dari fungsi "Create" dalam aplikasi CRUD. Halaman ini dirancang khusus untuk memfasilitasi penambahan data guru baru ke dalam sistem. Desain formulir ini mengutamakan kelengkapan informasi dan kemudahan pengisian, memastikan bahwa semua data yang dibutuhkan dapat diinput dengan akurat dan efisien. Pengguna, khususnya admin, diminta untuk mengisi berbagai bidang informasi yang relevan mengenai guru yang akan didaftarkan. Bidang-bidang ini mencakup informasi personal seperti nama lengkap, alamat email, dan nomor telepon, yang sangat penting untuk komunikasi dan identifikasi. Selain itu, formulir juga meminta detail profesional seperti mata pelajaran yang diampu, departemen tempat guru bertugas, status kepegawaian (misalnya, guru tetap, kontrak, atau honorer), serta tanggal perekrutan. Informasi ini krusial untuk manajemen sumber daya manusia dan perencanaan akademik. Terakhir, formulir juga mencakup kolom untuk kualifikasi pendidikan guru, yang memberikan gambaran lengkap mengenai latar belakang akademis mereka. Setiap kolom input dalam formulir ini dirancang dengan rapi dan intuitif, menggunakan jenis input yang sesuai (teks, email, angka, tanggal) untuk memastikan validasi awal data. Penempatan label yang jelas dan placeholder yang informatif membantu pengguna memahami jenis informasi yang diharapkan di setiap kolom. Tata letak formulir yang terstruktur secara logis, membagi informasi menjadi beberapa bagian jika diperlukan, mengurangi beban kognitif pengguna dan mempercepat proses pengisian. Tujuan utama dari desain ini adalah untuk memastikan bahwa data yang masuk ke sistem adalah lengkap, valid, dan akurat. Hal ini sangat penting untuk menjaga integritas database dan mencegah masalah di kemudian hari yang mungkin timbul dari data yang tidak lengkap atau salah. Implementasi fitur validasi client-side (misalnya, memeriksa format email atau memastikan bahwa bidang wajib tidak kosong) dapat lebih lanjut meningkatkan kualitas data yang dimasukkan. Dengan demikian, halaman ini tidak hanya berfungsi sebagai alat input data, tetapi juga sebagai garda terdepan dalam menjaga kualitas dan keandalan informasi guru dalam sistem.

Gambar 3.3 Form pendaftaran

c. Form Edit Guru

Formulir Edit Guru adalah komponen kunci dalam fungsionalitas "Update" dari aplikasi Teacher Management. Halaman ini dirancang untuk memungkinkan admin memperbarui informasi guru yang sudah terdaftar dalam sistem. Keunggulan utama dari formulir ini adalah kemampuannya untuk secara otomatis menampilkan data yang sudah ada dari guru yang dipilih. Saat admin memilih opsi "Edit" untuk seorang guru tertentu dari halaman dashboard, aplikasi akan mengambil data lengkap guru tersebut dari backend dan mengisi setiap kolom dalam formulir. Data yang otomatis terisi meliputi nama lengkap, alamat email, nomor telepon, mata pelajaran yang diampu, departemen, status kepegawaian, tanggal perekrutan, alamat, dan kualifikasi pendidikan. Fitur pra-pengisian ini sangat penting karena meminimalkan upaya input manual bagi admin. Admin tidak perlu mengetik ulang semua informasi dari awal; mereka hanya perlu mengubah bagian-bagian yang memang memerlukan pembaruan. Ini secara signifikan meningkatkan efisiensi dan mengurangi risiko kesalahan input yang mungkin terjadi jika data harus dimasukkan secara manual dari awal.

Desain formulir edit ini juga mengutamakan kesederhanaan dan kemudahan penggunaan. Tata letak yang bersih dan navigasi yang intuitif memastikan bahwa admin dapat dengan cepat mengidentifikasi bidang yang perlu diubah dan melakukan perubahan tanpa kebingungan. Setelah perubahan dilakukan, admin dapat menyimpan pembaruan, dan data baru akan diperbarui di database. Aplikasi akan memberikan umpan balik visual (misalnya, notifikasi sukses atau error) untuk mengkonfirmasi status operasi. Proses ini sangat vital untuk menjaga akurasi dan relevansi data guru di sistem. Seiring berjalannya waktu, informasi guru seperti status kepegawaian, mata pelajaran yang diajar, atau kualifikasi pendidikan mungkin berubah, sehingga formulir edit yang efisien ini menjadi alat yang tak tergantikan untuk menjaga data tetap up-to-date. Dalam hal implementasi teknis, ketika admin mengklik tombol "Edit" di dashboard, aplikasi React Native akan mengirimkan permintaan ke backend PHP Native (spesifiknya, endpoint `read_one.php`) untuk mengambil data guru berdasarkan ID yang dipilih. Setelah data diterima oleh frontend, komponen `TeacherForm` akan digunakan kembali (atau komponen serupa yang mengelola formulir), tetapi kali ini dengan data yang sudah terisi. Pengguna dapat memodifikasi bidang-bidang yang relevan. Ketika formulir disubmit, data yang diperbarui akan dikirim ke backend melalui metode PUT (atau POST dengan indikator update) ke endpoint `update.php`. Skrip `update.php` kemudian akan memvalidasi data yang diterima dan memperbarui record yang sesuai di database. Seperti halnya proses create, backend akan memberikan respons yang sesuai, dan frontend akan menampilkan pesan sukses atau error kepada pengguna. Penggunaan kembali komponen `TeacherForm` untuk kedua operasi (tambah dan edit) merupakan praktik terbaik dalam pengembangan React Native, karena mengurangi duplikasi kode dan mempermudah pemeliharaan. Ini menunjukkan fleksibilitas dan efisiensi dalam arsitektur pengembangan aplikasi.

Edit Teacher Profile
Update teacher information

Full Name *	Email *
Emily Davis	emily.davis@school.edu
Phone Number *	Subject *
(555) 321-0987	History
Department *	Employment Status *
Humanities Department	Part-time
Hire Date *	
03 / 20 / 2021	
Address	
321 Elm St, City, State 12345	
Qualifications	
Bachelor of History, Teaching Credential	

Gambar 3.4 Form edit

d. Future Plan Aplikasi

Aplikasi Teacher Management yang saat ini telah dikembangkan memiliki fungsionalitas inti yang kuat, meliputi operasi dasar seperti penambahan, pengeditan, penghapusan, dan melihat data guru. Fitur pencarian data dan informasi status kepegawaian juga telah diintegrasikan untuk meningkatkan efisiensi. Namun, untuk menjadikannya sistem yang lebih komprehensif, aman, dan user-friendly, terdapat beberapa fitur tambahan yang telah direncanakan untuk implementasi di masa mendatang.

1. Validasi input data yang lebih robust. Saat ini, validasi mungkin masih dasar. Ke depannya, validasi akan diperluas untuk mencakup skenario yang lebih kompleks, seperti validasi format nomor telepon, memastikan email unik, atau membatasi panjang karakter pada bidang tertentu. Validasi ini akan dilakukan baik di sisi client (frontend) untuk memberikan umpan balik instan kepada pengguna, maupun di sisi server (backend) untuk memastikan integritas data di database. Tujuannya adalah untuk mencegah data tidak valid atau berbahaya masuk ke sistem, sehingga meningkatkan keandalan informasi.
2. Sistem notifikasi. Implementasi notifikasi akan memberikan umpan balik yang lebih jelas dan instan kepada pengguna setelah suatu operasi berhasil dilakukan (misalnya, "Data Guru Berhasil Ditambahkan") atau jika terjadi kegagalan ("Gagal Menambahkan Data Guru, Silakan Coba Lagi"). Notifikasi ini dapat berupa toast message, snackbar, atau pop-up yang informatif, sehingga pengguna mengetahui status dari setiap tindakan yang mereka lakukan tanpa harus memeriksa secara manual.
3. Konfirmasi hapus data. Untuk mencegah penghapusan data yang tidak disengaja, akan ditambahkan dialog konfirmasi sebelum data benar-benar dihapus. Pengguna akan diminta untuk mengkonfirmasi kembali tindakan mereka (misalnya, "Apakah Anda yakin ingin menghapus data guru ini?"), memberikan lapisan keamanan tambahan dan mengurangi risiko kehilangan data yang tidak disengaja.
4. Fitur ekspor data. Admin seringkali memerlukan data dalam format yang dapat digunakan di luar aplikasi, seperti untuk pelaporan atau analisis. Fitur ini akan memungkinkan ekspor data guru ke berbagai format populer seperti CSV (Comma Separated Values) atau PDF (Portable Document Format). Ini akan sangat mempermudah proses pelaporan, audit, dan analisis data eksternal.
5. Pagination pada daftar guru. Untuk aplikasi yang mungkin akan mengelola ribuan data guru, menampilkan semua data dalam satu halaman akan membebani kinerja dan sulit dinavigasi. Implementasi pagination akan memecah daftar guru menjadi beberapa halaman, sehingga hanya sebagian kecil data yang dimuat dan ditampilkan pada satu waktu. Ini tidak hanya meningkatkan performa aplikasi (terutama untuk loading data) tetapi juga meningkatkan kemudahan navigasi bagi pengguna.
6. Implementasi sistem login dan otentikasi. Saat ini, diasumsikan aplikasi dapat diakses langsung. Namun, di masa mendatang, akses ke aplikasi akan dibatasi hanya untuk pengguna yang terdaftar dan terotorisasi. Sistem login akan memerlukan username dan password, dan otentikasi akan memverifikasi identitas pengguna. Ini akan memastikan bahwa hanya admin atau pihak yang berwenang yang dapat mengakses dan memanipulasi data guru, sehingga meningkatkan keamanan sistem secara keseluruhan dan melindungi informasi sensitif. Penambahan fitur otorisasi (misalnya, peran pengguna yang berbeda dengan hak akses berbeda) juga dapat dipertimbangkan di kemudian hari.

Penambahan fitur-fitur ini diharapkan akan menjadikan aplikasi *Teacher Management* lebih lengkap, aman, dan sesuai dengan kebutuhan administrasi data guru yang lebih kompleks di institusi pendidikan mana pun. Pengembangan ini akan dilakukan secara iteratif, sesuai dengan metodologi prototyping, di mana setiap fitur baru akan diimplementasikan, diuji, dan disempurnakan berdasarkan umpan balik pengguna.

4. KESIMPULAN

Aplikasi Teacher Management yang dikembangkan merupakan solusi efektif untuk pengelolaan data guru. Dengan pendekatan model prototyping, aplikasi ini mendukung fungsi CRUD (Create, Read, Update, Delete) secara lengkap dan dilengkapi fitur pencarian yang memudahkan akses data secara cepat dan akurat. Antarmuka yang sederhana dan user-friendly menjadikan aplikasi mudah digunakan oleh admin tanpa memerlukan keahlian teknis tinggi. Dari sisi teknologi, aplikasi dibangun dengan React Native di sisi frontend, memungkinkan pengembangan lintas platform dengan struktur modular yang rapi. Di sisi backend, PHP Native digunakan untuk menangani operasi CRUD melalui API yang terstruktur, memastikan komunikasi lancar antara sistem dan database. Integrasi kedua teknologi ini menunjukkan efisiensi dan keselarasan dalam membangun aplikasi mobile. Ke depan, pengembangan akan difokuskan pada penambahan fitur penting seperti validasi data yang lebih kuat, sistem notifikasi, konfirmasi penghapusan data, fitur ekspor, dan pagination. Yang paling penting, sistem login dan otentikasi akan diterapkan untuk meningkatkan keamanan dan memastikan hanya pengguna yang berwenang yang dapat mengakses data. Dengan pengembangan berkelanjutan, aplikasi ini siap menjadi alat penting dalam manajemen data guru di institusi pendidikan.

UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada pihak-pihak yang telah mendukung terlaksananya penelitian ini.

REFERENCES

- [1] A. Berkati, Licantik, N. Nugrahaningsih, A. Lestari, and F. Sylviana, "Rancang Bangun Aplikasi Absensi Guru Dan Staf Tu Dengan Penerapan Geolocation Dan Fingerprint Berbasis Android Di Smk Gke Mandomai," *J. Teknol. Inf. J. Keilmuan dan Apl. Bid. Tek. Inform.*, vol. 18, no. 1, pp. 61–74, 2024, doi: 10.47111/jti.v18i1.12590.
- [2] E. Gunawan, S. Angkasa, H. M. Junaidi, H. Yunita, and A. T. Fihawiano, "Sistem informasi pendataan guru dan siswa pada SD Negeri Keraton 3 Martapura berbasis web," *J. Pendidik. dan Konseling*, vol. 4, no. 6, pp. 306–318, 2022.
- [3] E. Asoka, L. Rahmi, Y. Hapsari, and S. Sulistiyanto, "Penggunaan Framework React Native Dalam Perancangan Aplikasi Penjualan Goodday Garden," *Int. Res. Big-Data Comput. Technol. I-Robot*, vol. 8, no. 2, pp. 19–23, 2024, doi: 10.53514/ir.v8i2.569.
- [4] M. A. Karim and A. R. Adriansyah, "Analisis dan Perancangan Aplikasi Mobile untuk Donasi menggunakan Metode Hybrid berbasis React Native," *J. Inform. Terpadu*, vol. 8, no. 1, pp. 26–34, 2022, doi: 10.54914/jit.v8i1.394.
- [5] H. A. CHANDRA, H. Hoiriyah, and F. Ekawati, "Metode Cepat Pembuatan Aplikasi Web Bootstrap Akademik Pada SMK Kota Banjarmasin Berbasis Codeigniter," *Technol. J. Ilm.*, vol. 15, no. 3, p. 597, 2024, doi: 10.31602/tji.v15i3.15494.
- [6] M. Alda, "Pengembangan Aplikasi Pengolahan Data Siswa Berbasis Android Menggunakan Metode Prototyping," *J. Manaj. Inform.*, vol. 13, no. 1, pp. 11–23, 2023, doi: 10.34010/jamika.v13i1.8216.
- [7] M. Baharu, "Minda Baharu, Volume 5, No 2 Desember, 2021," vol. 5, no. 2, pp. 89–94, 2021.
- [8] A. H. Nasyuha, Z. Zulkifli, I. Purnama, A. Sidabutar, A. Karim, and M. Mesran, "Sistem Pendukung Keputusan Penentuan Kerani Timbang Lapangan Terbaik Menerapkan Metode Operational Competitiveness Rating Analysis (OCRA)," *J. Media Inform. Budidarma*, vol. 6, no. 1, p. 355, 2022, doi: 10.30865/mib.v6i1.3475.
- [9] A. Pramana, R. Watrianthos, and I. Purnama, "Sistem Informasi Pendaftaran Mahasiswa Baru Berbasis Android," *J. Inform. Upgris*, vol. 5, no. 2, pp. 121–125, 2019, doi: 10.26877/jiu.v5i2.3807.