

Implementasi Web Service Restful Dengan Autentikasi Json Web Token Berbasis Web dan Android

Ahmad Pudoli¹, Yulianawati², Iqbal Suwandi³

^{1,3} Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur

² Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Universitas Budi Luhur

Email: ¹ahmad.pudoli@budiluhur.ac.id, ²yulianawati@budiluhur.ac.id, ³1911501219@student.budiluhur.ac.id

Abstrak - Sistem pembelajaran saat ini mendapatkan pengaruh sangat besar oleh perkembangan teknologi, dalam melakukan pemanfaatan pembelajaran ini memudahkan antara murid dan guru untuk tidak harus bertemu secara langsung. Pada sistem pembelajaran ini adalah untuk memberikan kemudahan kepada pengguna. Pada pembuatan aplikasi sistem pembelajaran ini membutuhkan bagian CMS (Content Management System) yang berguna untuk mengelola dibagian belakang dari user interface agar mudah di kelola dan sangat user friendly. Oleh karena itu perlu dibuat suatu aplikasi sistem pembelajaran untuk meningkatkan proses kinerja pada yang semula manual menjadi terkomputerisasi. Pada pembuatan program aplikasi ini dikembangkan dengan bahasa JavaScript. Sementara untuk integrasi dengan sistem yang sudah ada diperlukan web service sebagai backend system sehingga sistem pembelajaran dapat diakses oleh berbagai platform. Arsitektur yang digunakan pada web service menggunakan REST API, namun masih ada beberapa masalah pada REST API yaitu mengenai keamanan pada proses otentikasi. Pada arsitektur REST API diperlukan metode otentikasi, salah satunya dapat menggunakan JSON Web Token. Hasil penelitian ini menunjukkan bahwa penggunaan JSON Web Token Authentication pada web service aplikasi CMS (Content Management System) dapat membentuk sistem yang sangat skalabilitas, mampu berinteraksi multiplatform. Setelah tahap penelitian pengembangan sistem dibuat, dan dilakukan proses pengujian dengan blackbox serta usability semua pengujian yang berhubungan dengan fungsionalitas aplikasi di android dan web dapat diterima serta dapat dijalankan sesuai yang diharapkan. Dari hasil pengujian usability, diperoleh tingkat keberhasilan dalam sistem dalam menyimpan data di flatform android sebesar 97% , keberhasilan dalam menyimpan di website 98%, jika data yang diinput terdapat kesalahan pada jenis tipe data maupun panjang karakternya oleh karena itu aplikasi ini dapat dinyatakan memiliki kualifikasi sangat baik dan berhasil.

Kata Kunci - Web service, JSON, REST

Abstract - *The current learning system is greatly influenced by technological developments, in making use of this learning it makes it easier for students and teachers not to have to meet in person. In this learning system is to provide convenience to the user. In making this learning system application requires a CMS (Content Management System) section which is useful for managing the back of the*

user interface so that it is easy to manage and very user friendly. Therefore it is necessary to make a learning system application to improve the performance process from being manual to computerized. In making this application program developed with the JavaScript language. Meanwhile, for integration with existing systems, a web service is needed as a backend system so that the learning system can be accessed by various platforms. The architecture used in the web service uses the REST API, but there are still some problems with the REST API, namely regarding security in the authentication process. The REST API architecture requires an authentication method, one of which can use JSON Web Token. The results of this study indicate that the use of JSON Web Token Authentication on a CMS (Content Management System) application web service can form a highly scalable system, capable of multi-platform interaction. After the system development research stage is made, and the testing process is carried out with blackbox and the usability of all tests related to application functionality on android and web is acceptable and can be run as expected. From the results of usability testing, the success rate of the system in storing data on the Android platform is 97%, success in saving on the website is 98%, if the input data contains errors in the data type and character length, this application can be declared qualified. very good and successful.

Keywords - Web service, JSON, REST

I. PENDAHULUAN

Inovasi yang terus bergerak maju seiring dengan perkembangan zaman menjadi bagian dalam kebutuhan dalam mencukupi alat bantu manusia dalam melakukan pengolahan data menjadi suatu informasi yang berharga[1]. Kebutuhan akan lengkapnya mengakses data atau informasi dari antarmuka web, maupun android menjadi salah satu indikator sebuah instansi dikatakan memiliki manajemen data yang baik. Penggunaan sebuah situs web tak hanya digunakan sebagai media informasi, hal ini dapat juga dapat digunakan dalam semua bidang ilmu, salah satunya adalah E-learning[2].

Sebuah situs website bisa dikatakan statis yaitu dimana salah satunya dalam web tersebut tidak melakukan perubahan isi ataupun konten pada saat terjadi request data ke server[3]. Tampilan web tersebut hanya akan berubah apabila pihak admin membuat aktivitas update dalam isi website secara manual.

Web Service menurut pengertian adalah rangkaian standar serta metode dalam pemrograman untuk berbagi data antara aplikasi perangkat lunak yang berbeda, kemudian melakukan

pendistribusian layanan/service melalui layanan internet yang mendukung interoperabilitas sistem[4]. Trend era digitalisasi ini yaitu adanya penggunaan aplikasi berbasis android yang dapat memberikan tampilan yang menarik dan interaktif[4].

Pembuatan website berbasis Content Management System (CMS) sebagai tools atau alat bantu yang memudahkan kepada user dalam melakukan pengelolaan, maintenance, mengedit konten dan modul serta membuat perubahan isi konten dan memperbarui data yang ada untuk ditampilkan pada sebuah website[5].

Dalam membangun sebuah website berbasis *Content Management System (CMS)* yang digunakan sebagai tools atau alat bantu untuk dapat memudahkan user dalam membuat pengelolaan, pemeliharaan, perubahan isi atau konten dan juga modul serta dapat memperbarui data yang ada untuk dapat ditampilkan pada sebuah situs. Hal ini dapat dilakukan oleh developer maupun penggunaannya[5].

Integrasi dengan sistem yang ada membutuhkan web service sebagai sistem backend agar layanan donor darah dapat tersedia dari platform yang berbeda. Arsitektur yang digunakan pada web service menggunakan teknologi REST, namun REST masih memiliki kendala terkait keamanan proses autentikasi. Metode autentikasi stateless diperlukan dalam arsitektur REST, salah satunya dapat menggunakan token web JSON. Hasil penelitian ini menunjukkan bahwa penggunaan autentikasi token web JSON dalam layanan web Blood Donors dan sistem backend dapat membentuk sistem yang sangat terukur, aman, multi-interaktif, dan dapat diterima keakuratannya[6]. Permasalahan yang akan dihadapi adalah bagaimana membuat dan mengetahui bahwa web service yang efisien dan tidak rentan. Keamanan tidak luput dari istilah yang disebut kriptografi. Maka kriptografi merupakan suatu hal yang penting terutama pada bagian transfer data dalam suatu aplikasi. Kriptografi yang digunakan adalah metode AES, dan dengan arsitektur RESTFUL API pada bagian web service. Juga dengan menggunakan token autentikasi untuk akun sehingga tidak sembarang akun dapat mengakses API. Tujuan akhir dari penelitian ini adalah menganalisis kinerja layanan yang diberikan dan juga membandingkan hasil waktu respon yang keluar saat menggunakan metode enkripsi antara dengan tidak menggunakannya[7].

Absensi merupakan salah satu indikator tingkat kinerja, khususnya kedisiplinan seorang pegawai. Berdasarkan pendeteksian masalah, diperlukan website Absensi Karyawan. Tentunya aplikasi ini harus seefektif perangkat sidik jari tradisional di kantor. Aplikasi ini memiliki fitur validasi yang menggunakan penandaan lokasi, sidik jari, dan kontrol perangkat untuk meminimalkan penipuan saat karyawan berpartisipasi. Penelitian ini mengimplementasikan fitur keamanan RESTful API pada web service menggunakan algoritma HMAC SHA-256 berbasis JSON Web Token (JWT). Semua langkah implementasi diuji menggunakan metode Black Box dan terbukti bahwa JWT dapat mengamankan proses autentikasi, melakukan proses request dan response serta melindungi data. Selain itu, fungsi verifikasi dapat memberikan informasi kehadiran dengan akurasi 90,9%.[8]. Inputan yang diberikan dari penelitian atas

pengujian model implementasi JSON Web Token dengan menggunakan penerapan algoritma SHA-512 di aplikasi BatikKita yaitu didapatkan hasil kecepatan menjadi meningkat dengan nilai rata-rata diantara 138,8 milisecond pada SOAP serta 122,7 milisecond pada REST jika dikomparasi pada algoritma HMAC SHA-256 dan algoritma HMAC SHA-384. Adapun untuk jenis ukuran token yang diperlukan jauh lebih tinggi yaitu antara 2,13 kb untuk SOAP serta 2,11 kb pada REST jika dikomparasi dengan algoritma HMAC SHA-256 dan algoritma HMAC SHA-384[9]. Dalam konstruksi dan proses integrasi akan didapatkan sebuah kesimpulan yaitu cara menggunakan data produk pada CV. XYZ, menjadi sebuah sistem Inventory yang dikonstruksi dengan memakai teknologi web service penggunaan pengelolaan data API barang yang berjenis data JSON di Server Rest agar bisa dipakai pada Client Rest. Pada riset tersebut mendapatkan Web service dengan tahapan dan proses ujicoba/testing via service yang berbeda dengan angka total data yang diminta/request dan diterima yaitu 62.48 kb dan total waktu Response dari pengambilan data 2.2 detik dengan waktu rata-rata eksekusi Request data sebanyak 177 mili detik[10]. Proses otentikasi menggunakan JWT (JSON Web Token) dapat menentukan hak akses pada fungsi serta sumber daya yang tersedia didalam back-end server. Adapun hasil yang didapatkan yaitu aplikasi yang dibuat berbasis web yang sudah memenuhi seluruh product backlog yang ditentukan oleh product owner. Hasil dari penelitian ini adalah metode yang diterapkan mampu mengembangkan fitur di tengah proses pengembangan aplikasi tanpa mempengaruhi pengembangan fitur utama dan metode ini efektif digunakan bagi lingkungan pengembang aplikasi yang berbeda [11].

Pembelajaran daring terus dilihat sebagai paradigma baru dalam kegiatan belajar mengajar dimana kegiatan antara mahasiswa dan pembimbing profesional tidak perlu hadir di dalam kelas[12]. Siswa hanya bergantung pada koneksi internet untuk menyelesaikan proses belajar mengajar, dan proses ini dapat dilakukan dari jarak jauh dan kapan saja. Karena kemudahan dan kepraktisan yang ditawarkan oleh PT. Mojadi Aplikasi Indonesia, tidak heran jika banyak siswa yang lebih memilih sistem virtual atau pembelajaran online. Pembelajaran online dapat dilakukan di mana saja ada siswa dan mentor profesional. Namun, muncul pertanyaan apakah fungsi dan psikologi pembelajaran daring memiliki pengertian yang sama, atau setidaknya fungsi dan psikologi pembelajaran tatap muka.

REST (*Representational State Transfer*) adalah metode arsitektur untuk membuat teknologi web service yang mengimplementasikan konsep peralihan antar keadaan dengan menggambarkannya sebagai browser yang meminta halaman web sambil meminta keadaan halaman web saat ini di sisi server yang dikirimkan browser[13]. Ide dasar dibalik REST adalah berfokus terutama pada interaksi sumber daya dan beralih antar status alih-alih berfokus pada pengiriman dan penerimaan pesan seperti web service berbasis SOAP (Simple Object Access Protocol). Prinsip dari metode REST yaitu identifikasi resource dengan URI (Uniform Resource Identifier), keseragaman resource (setiap resource dalam REST ditangani oleh empat operasi yaitu GET, PUT, POST, DELETE create, read, update, delete) self - Describe Message

(akses ke beberapa format seperti PDF, HTML, JPEG, XML, JSON, teks), hyperlink untuk interaksi spasial[14].

PT. Mojadi Aplikasi Indonesia adalah platform pertama dan terlengkap di Indonesia dengan pendekatan psikologis dan fitur ahli untuk memudahkan semua orang memilih jurusan dan mencapai tujuan impiannya. PT. Mojadi Aplikasi Indonesia didirikan pada tahun 2020, PT. Mojadi Aplikasi Indonesia berawal dari ketertarikan untuk berkontribusi dalam kemajuan dunia pendidikan di Indonesia. PT. Mojadi Aplikasi Indonesia hadir sebagai platform untuk mendukung dunia pendidikan guna mempersiapkan pelajar Indonesia menuju masa depan yang cerah. PT. Mojadi Aplikasi Indonesia bertujuan untuk meningkatkan kualitas pembelajaran dan berfungsi sebagai platform untuk mendukung pendidikan yang terjangkau dan berkualitas internasional, dapat diakses oleh semua siswa Indonesia (baik yang kurang mampu maupun yang mampu/baik perkotaan maupun pedesaan).

PT. Mojadi Aplikasi Indonesia menawarkan beberapa *video* dari mentor profesional yang ahli di bidangnya dan pembelajaran online untuk menawarkan solusi kepada siswa agar tidak perlu khawatir tentang masa depan mereka. Pembelajaran daring terus dilihat sebagai paradigma baru dalam kegiatan belajar mengajar dimana kegiatan antara mahasiswa dan pembimbing profesional tidak perlu hadir di dalam kelas. Siswa hanya bergantung pada koneksi internet untuk menyelesaikan proses belajar mengajar, dan proses ini dapat dilakukan dari jarak jauh dan kapan saja. Karena kemudahan dan kepraktisan yang ditawarkan oleh PT. Mojadi Aplikasi Indonesia, tidak heran jika banyak siswa yang lebih memilih sistem virtual atau pembelajaran online. Pembelajaran online dapat dilakukan di mana saja ada siswa dan mentor profesional. Namun, muncul pertanyaan apakah fungsi dan psikologi pembelajaran daring memiliki pengertian yang sama, atau setidaknya fungsi dan psikologi pembelajaran tatap muka

Tujuan dari teknik implementasi adalah merancang sistem dengan arsitektur REST yang dapat digunakan dengan berbagai klien seperti aplikasi mobile dan program web untuk membantu PT. Mojadi Aplikasi Indonesia akan dialihkan dari sistem manual ke sistem komputer untuk meningkatkan kinerja PT. Mojadi Aplikasi Indonesia, namun hanya aplikasi berbasis web dan mobile yang digunakan dalam pengujian aplikasi. Untuk memudahkan dan mempercepat siswa dalam mengajar dan belajar kursus online. Namun sejak siswa PT. Mojadi Aplikasi Indonesia bersifat serbaguna dan memiliki perangkat yang berbeda-beda, sehingga diperlukan suatu aplikasi yang memfasilitasi sistem yang dibuat antar aplikasi pada platform yang berbeda. Untuk mengatasi hal tersebut diperlukan sebuah web service dengan arsitektur RESTful API dan autentikasi keamanan dengan JSON (Json Web Token). Web service adalah mekanisme interaktif antar sistem yang mendukung interoperabilitas dan mempromosikan integrasi data, serta dapat diakses oleh banyak pihak melalui internet menggunakan perangkat berbeda yang dimiliki oleh setiap pengguna. Melalui layanan online, dimungkinkan untuk bertukar informasi dari satu sistem ke sistem lainnya, bahkan ketika menggunakan platform dan bahasa pemrograman yang berbeda pada waktu yang bersamaan. Kemudian bagaimanakah cara membuat website yang awalnya bersifat

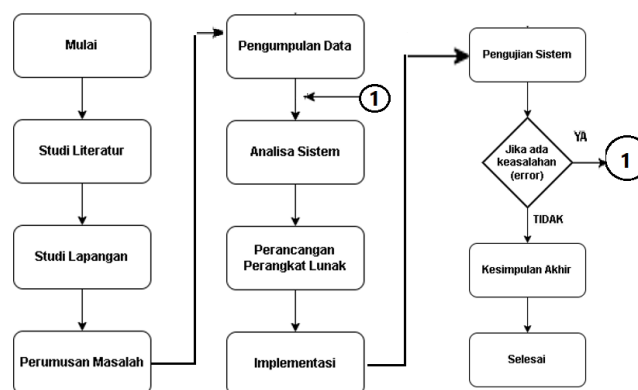
statis namun dapat terintegrasi dengan website lainnya. Saat mengimplementasikan web service, banyak kekurangan yang muncul jika penggunaannya tidak menyertakan autentikasi saat *login*. Dengan menggunakan autentikasi yang artinya token ini menggunakan JSON (JavaScript *Object Notation*) berbentuk *string* panjang yang sangat acak, token ini memungkinkan kita mengirimkan data yang dapat diverifikasi oleh dua pihak atau lebih. Seperti *password*, server memberikan token setelah *login* berhasil. Kemudian, *user* menyimpan token di penyimpanan *local* atau *cookie* di browser, dan saat *user* ingin mengakses halaman tertentu, mereka harus menyertakan token tersebut. Karena itu, *user* mengembalikan pengidentifikasi yang ditentukan sebelumnya sebagai bukti bahwa *user* ini *login*. Bagaimana membuat suatu aplikasi CMS yang dapat menghubungkan aplikasi web dan mobile secara cepat dan aman agar bisa di akses secara terpusat? Bagaimana membuat web service yang dapat digunakan oleh beberapa aplikasi client?

II. METODE PENELITIAN

Data yang akan digunakan dalam penelitian ini adalah file data siswa dan administrator pada tahun 2022 yang ada di PT. Mojadi Aplikasi Indonesia.

A. Penerapan Metode

Metode penelitian ini digunakan sebagai pedoman untuk melakukan penelitian agar hasil yang diperoleh tidak menyimpang dari tujuan yang telah dicapai sebelumnya. Gambar 1. menunjukkan langkah-langkah penerapan metode penelitian yang diterapkan dalam penelitian ini :



Gambar 1. Tahapan penelitian

B. Analisis Penerapan Algoritma

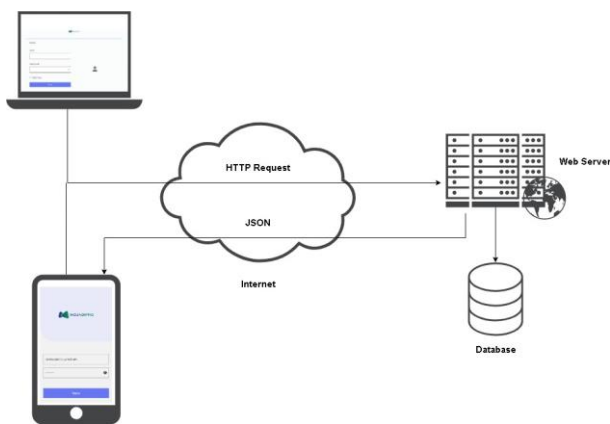
Setelah tahap pengumpulan data dan observasi operasi sistem, dilakukan analisa penerapan algoritma. Analisa penerapan algoritme menjelaskan langkah-langkah implementasi JSON dan metode yang ditambahkan adalah algoritme Base64. Algoritme Base64 adalah salah satu algoritme untuk *encoding* dan *decoding* data ASCII yang didasarkan pada bilangan dasar 64 atau bisa dikatakan salah satu metode yang digunakan melakukan penyandian (*encoding*) data biner[15]. Implementasi REST yang digunakan pada sistem dimaksudkan untuk mengolah data yang nantinya akan disimpan dalam *database*. JSON

digunakan untuk melindungi data yang disimpan dalam *database*. Oleh karena itu, menyimpan data dalam *database* memerlukan modul untuk melakukan *hashing* pada data tersebut. Mesin pengkodean berada di dalam aplikasi, yang dipanggil saat pengguna ingin melihat data.

C. Perancangan Perangkat Lunak

Ini Pada fase ini, perencanaan didasarkan pada hasil analisis sistem, terutama desain metode REST dan modul pendukung lainnya yang nantinya akan diintegrasikan ke dalam aplikasi, serta desain antarmuka pengguna.

Metode *Waterfall* digunakan dalam pengembangan perangkat lunak. Model ini mengharuskan satu langkah diselesaikan sepenuhnya sebelum melanjutkan ke langkah berikutnya, dan hasil dari setiap langkah harus didokumentasikan dengan baik. Gambar 2 menunjukkan alur perancangan perangkat lunak.

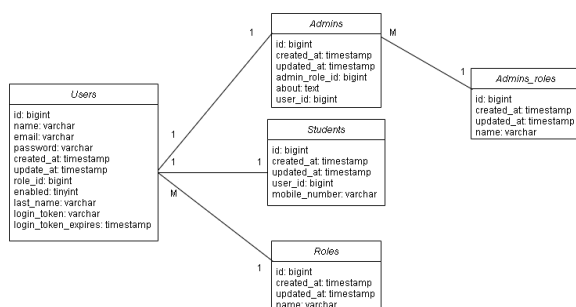


Gambar 2. Alur Perancangan Perangkat Lunak

Pada proses ini, implementasi terjadi dengan menerjemahkan modul-modul yang dirancang pada tahap desain ke dalam bahasa pemrograman tertentu. Dalam hal ini aplikasi yang digunakan adalah: 1) Perangkat lunak yang digunakan dalam implementasi integrasi web service ini menggunakan bahasa pemrograman Express JS serta DBMS-nya adalah PHPMyAdmin. 2) Perangkat keras yang digunakan meliputi Processor AMD Ryzen 5 2500U, RAM 12.0 GB, SSD 118 GB

D. Class Diagram

Class diagram adalah deskripsi dari struktur dan hubungan objek aplikasi. Struktur berisi atribut dan metode yang ada di setiap kelas. Hubungan pada pada setiap kategori yang ada ditunjukkan pada Gambar 3.

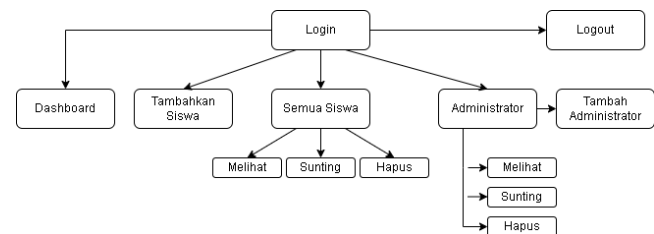


Gambar 3. Class Diagram

E. Rancangan Menu

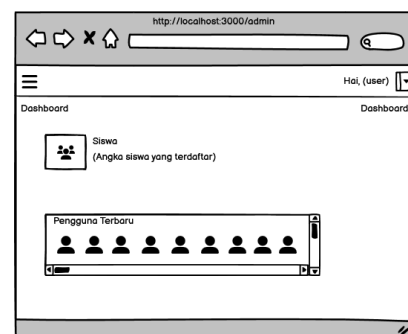
Beberapa halaman yang dibuat dalam program ini antara lain *form login*, halaman *dashboard*, *form* tambahkan siswa, halaman siswa, halaman administrator dan *form* menambah administrator.

Pada *form login*, pengguna hanya perlu memasukkan alamat *email* dan kata sandi sebelum melanjutkan aplikasi. Setelah *form login* berhasil, maka akan muncul halaman *dashboard* yang hanya menampilkan informasi siswa, siswa aktif dan *username*. Halaman *dashboard* ada menu *sidebar* yang menampilkan *form* tambahkan siswa, halaman semua siswa, halaman administrator dan *form* menambah administrator. Saat pengguna pertama kali mengklik *form* tambahkan siswa, halaman formulir akan terbuka dimana pengguna dapat menambahkan siswa dan langsung masuk ke halaman semua siswa. Halaman semua siswa berisi tabel tempat pengguna dapat melihat informasi siswa dan menu tempat pengguna dapat melihat semua informasi siswa, mengedit informasi siswa, dan menghapus akun siswa secara permanen. Semua halaman aplikasi ini memiliki menu *sidebar* di pojok kiri atas. Ketika pengguna mengklik *sidebar*, halaman administrator ditampilkan. Pada halaman administrator sama seperti semua siswa hanya menampilkan informasi administrator dalam bentuk tabel, menu yang memungkinkan untuk melihat informasi administrator secara otomatis, mengganti dan menghapus informasi administrator. Terdapat menu untuk menambahkan akun administrator dan administrator secara permanen. Pada *form* menambah administrator, pengguna harus mengisi formulir pendaftaran untuk menambahkan administrator dan langsung masuk ke halaman administrator.



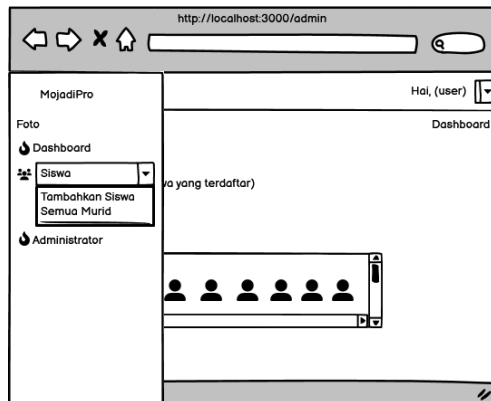
Gambar 4. Rancangan Menu Web

Rancangan layar halaman *dashboard* adalah tampilan setelah pengguna masuk. Tampilan ini memiliki menu jumlah siswa yang terdaftar, menu pengguna terbaru dan dapat ditampilkan pada Gambar 5.



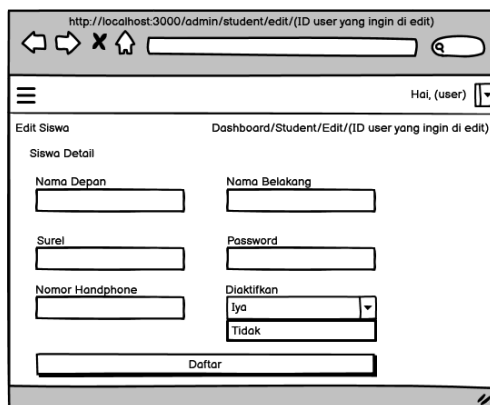
Gambar 5. Rancangan Layar Halaman Dashboard Web

Seperti yang ditunjukkan pada Gambar 5, ada tiga garis pada sudut kiri atas. Tiga garis tersebut disebut *sidebar*, yang muncul menampilkan menu-menu seperti halaman *dashboard*, siswa dan administrator. Dapat ditunjukkan pada Gambar 6



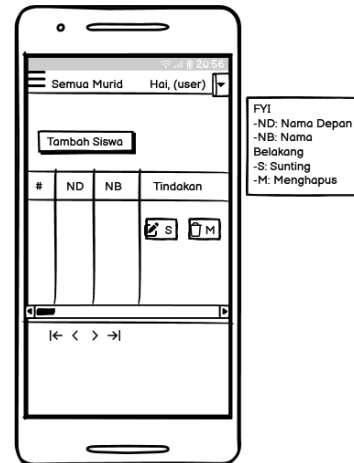
Gambar 6. Rancangan Layar Halaman *Sidebar* Web

Rancangan Layar *form edit* data siswa merupakan tampilan *form* untuk *edit* data siswa seperti kata sandi, nama depan, nama belakang, *email*, nomor *handphone*. Rancangan ini ditunjukkan pada Gambar 7.



Gambar 7. Rancangan Layar *Form Edit* Data Siswa Web

Rancangan layar halaman semua siswa adalah tampilan dimana semua siswa saat ini yang sudah didaftarkan oleh administrator. Tampilan ini berbentuk tabel, terdapat tombol *action* untuk menambah siswa, mengganti data siswa dan menghapus akun siswa. Rancangan ini ditunjukkan pada Gambar 8.



Gambar 8. Rancangan Layar Halaman Semua Siswa *Mobile*

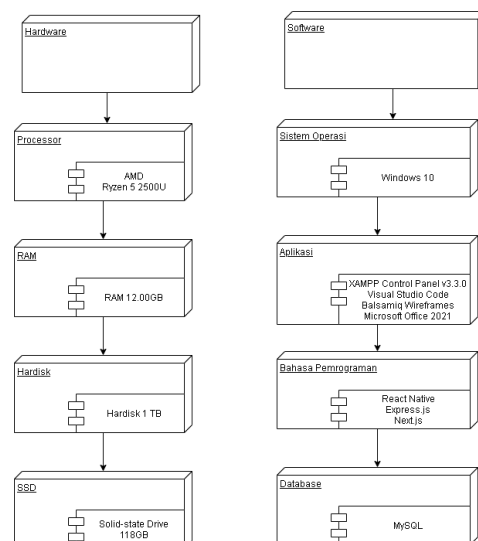
F. Spesifikasi Hardware dan Software

Software yang digunakan untuk mengimplementasikan aplikasi ini adalah sebagai berikut:

- 1) Sistem Operasi Windows 10.
- 2) XAMPP Control Panel v3.3.0.
- 3) Microsoft Office 2021.
- 4) Balsamiq Wireframes.
- 5) Visual Studio Code.
- 6) React Native 18.1.0.
- 7) Express.js 4.18.2.
- 8) Next.js 13.0.6.
- 9) MySQL.

Hardware yang digunakan untuk mengimplementasikan aplikasi ini adalah sebagai berikut:

- a) Processor AMD Ryzen 5 2500U.
- b) RAM 12.00GB.
- c) Hardisk 1 TB.
- d) Solid-State Drive 118GB



Gambar 9. Spesifikasi *Software* dan *Hardware*

III. HASIL DAN PEMBAHASAN

Beberapa langkah yang diambil selama tahap implementasi metode, termasuk yang berikut:

A. Proses Metode GET

Fungsi *read-only* yang digunakan untuk meminta informasi spesifik ke server dalam bentuk permintaan, properti operasi GET bersifat idempoten dan *aman*, tidak peduli berapa kali prosedur ini dilakukan, hasilnya tetap sama. Sebaliknya, *secure* artinya saat fungsi ini ditarik, status server belum berubah.

B. Proses Metode POST

Metode yang digunakan dalam kelompok HTTP ini adalah membuat objek atau *resource* terbaru. Kelompok metode ini biasanya tidak mengembalikan *output* yang disebut *procedure*

C. Proses Metode PUT

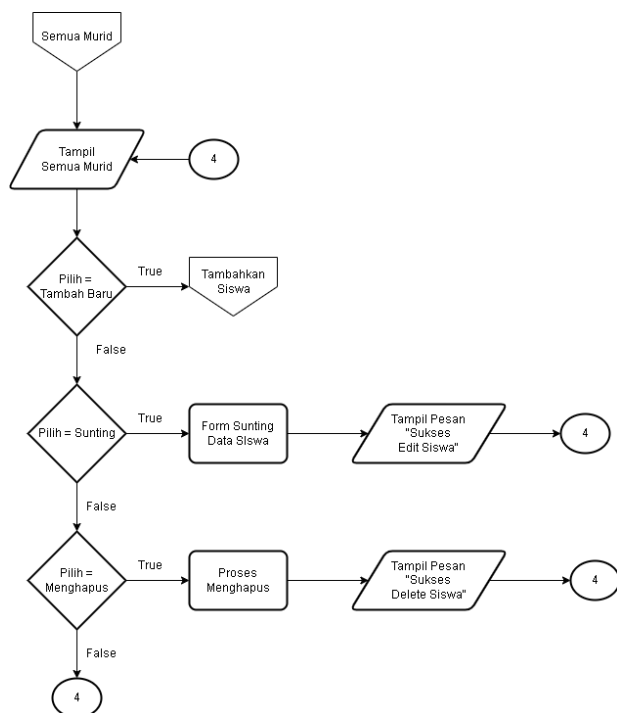
Metode yang digunakan dalam kelompok HTTP ini adalah memperbarui *item/resource* yang telah ada

D. Proses Metode DELETE

Penggunaan metode untuk kelompok HTTP ini adalah untuk menghapus *object/resource* yang sudah ada sebelumnya

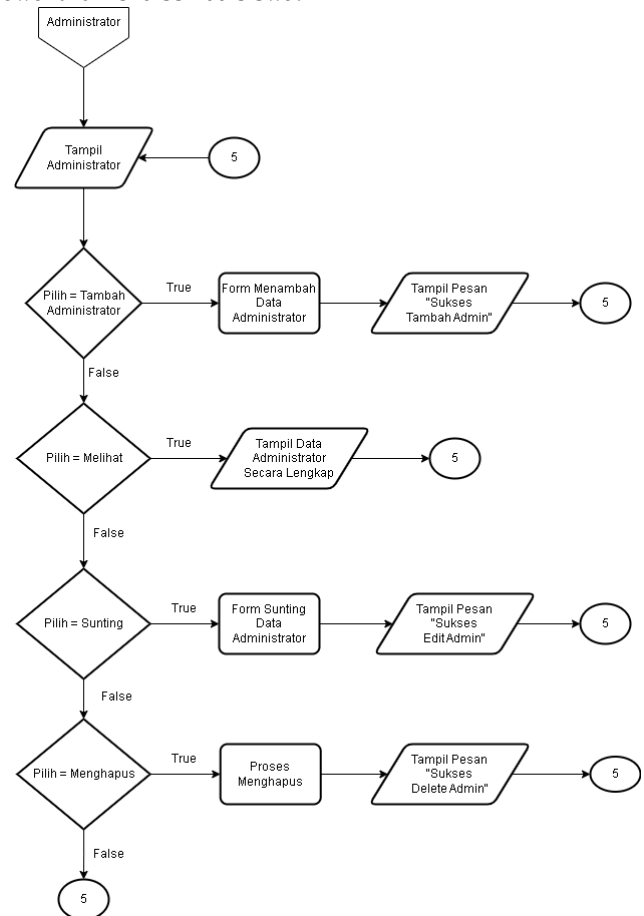
E. Flowchart

Saat menggambarkan urutan proses aplikasi, *flowchart* digunakan untuk menunjukkan alur proses. Dibawah ini adalah beberapa *flowchart* untuk setiap proses di tombol pada setiap *form*. *Flowchart form login* merupakan proses untuk menuju menu *dashboard*. Pertama dimulai isi kolom *email* dan *password*, kemudian setelah diisi akan diproses lagi untuk konfirmasi, jika belum diisi harus terlebih dahulu mengisi kolom, jika alamat *email* atau *password* kosong maka perlu diisi kembali jika *login* berhasil maka akan muncul menu *dashboard*.



Gambar 10. Flowchart Menu Semua Siswa Mobile

Flowchart menu semua siswa adalah proses menjalankan program pada menu semua siswa, yang berisi menu dan sub-menu yang dapat dikontrol oleh pengguna. Di bawah ini *flowchart* menu semua siswa.

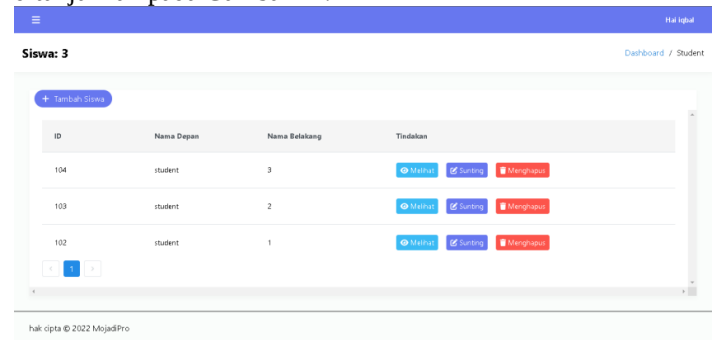


Gambar 11. Flowchart Menu Administrator Web

Flowchart menu administrator adalah proses menjalankan program dalam menu administrator yang berisi menu dan sub-menu yang dapat dikontrol oleh pengguna

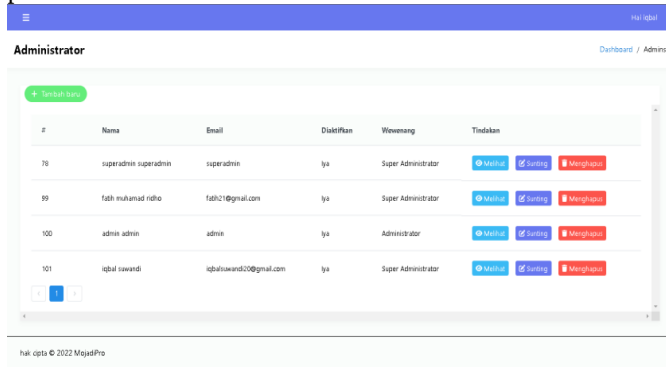
F. Tampilan Layar Halaman Siswa

Tampilan layar halaman semua siswa adalah tampilan dimana semua siswa saat ini yang sudah didaftarkan oleh administrator. Tampilan ini berbentuk tabel, terdapat tombol *action* untuk melihat data siswa secara lengkap, mengganti data siswa dan menghapus akun siswa. Rancangan ini ditunjukkan pada Gambar 12.



Gambar 12. Tampilan Layar Halaman Semua

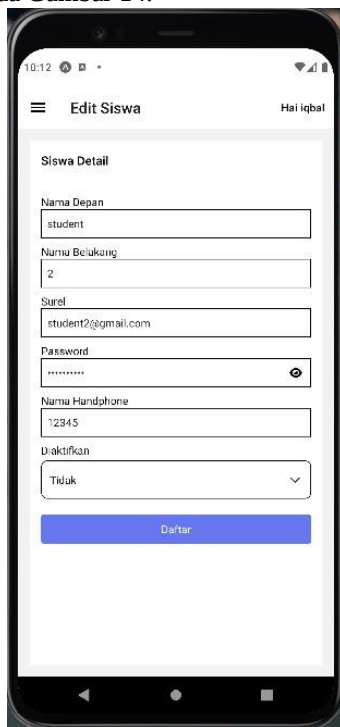
Tampilan layar halaman administrator adalah tampilan dimana semua administrator saat ini yang sudah di daftarkan. Tampilan ini dalam format tabel dan memiliki tombol *action* untuk menambah baru administrator, melihat informasi lengkap data administrator, mengganti data administrator dan menghapus administrator. Rancangan ini di tunjukkan pada Gambar 13.



Gambar 13. Tampilan Layar Halaman Administrator Web

G. Tampilan Layar Aplikasi Android

Tampilan layar *form edit* data siswa merupakan tampilan *form* untuk *edit* data siswa seperti kata sandi, nama depan, nama belakang, *email*, nomor *handphone*. Rancangan ini ditunjukkan pada Gambar 14.



Gambar 14. Tampilan Layar *Form Edit* Data Siswa Mobile

H. Penqujian Program

Untuk memastikan keberhasilan pengoperasian aplikasi ini, perlu untuk menguji aplikasi yang sudah selesai. Jika tidak mencobanya terlebih dahulu, aplikasi mungkin gagal. Tujuan dari pengujian ini adalah untuk mengetahui performa dari hasil web service yang dapat digunakan oleh beberapa aplikasi dengan bahasa pemrograman yang berbeda dan hasil dari

token yang digunakan untuk mengakses aplikasi tersebut. Tabel yang menunjukkan hasil web service yang diimplementasikan untuk aplikasi web.

Tabel 1. Tabel Pengujian Layanan Web service

Nama Layanan	Tampilan Berhasil
Login User	<pre>{ "date": { "_id": "60c8e9d7f000000000000000", statusText: "OK", headers: AxiosHeaders, config: { "_id": "..." } }, "config": { "> adapters: (1) [http, http] > baseUrl: "https://localhost:4000/api/v1" > data: {"email":"admin@gmail.com","password":"superadmin"} > env: { FormData: f, Blob: f } > Headers: AxiosHeaders {accept: application/json, text/plain: */*, Content-Type: application/json} > bodyLength: -1 > maxContentLength: -1 > maxContentLength: -1 > method: "post" > timeout: 0 > transformRequest: [f] > transformResponse: [f] > transitional: {silentJSONParsing: true, forcedJSONParsing: true, clarifyJSONError: false} url: "http://localhost:4000/api/v1/login" > validateStatus: f validateStatus(status) xhrOptions: {} responseType: "json" [Symbol]: function() {} }, "data": { "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkbSI6ImFkbWUiLCJyb2I6ImFkbWUiLCJpdiI6IjEiLCJpcyI6ImFkbWUiLCJhdWQiOiJkaWQzIiwiaWF0IjoxNjYxMjA0MDAwfQ.", "enabled": true, "date": { "_id": "60c8e9d7f000000000000000", statusText: "OK", headers: AxiosHeaders, config: { "_id": "..." } } }, "request": XMLHttpRequest {readyStateString: null, readyState: 4, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, ...} status: 200 statusText: "OK" }</pre>
Create Akun Siswa	<pre>{ "date": { "_id": "60c8e9d7f000000000000000", statusText: "OK", headers: AxiosHeaders, config: { "_id": "..." } }, "config": { "> adapters: (1) [http, http] > baseUrl: "https://localhost:4000/api/v1" > data: {"email":"student@gmail.com","password":"123456","role_id":1,"enabled":{"_id":"60c8e9d7f000000000000000","password":"superadmin"},"env":{FormData:f,Blob:f}} > env: { FormData: f, Blob: f } > Headers: AxiosHeaders {accept: application/json, text/plain: */*, Content-Type: application/json} > bodyLength: -1 > maxContentLength: -1 > method: "post" > timeout: 0 > transformRequest: [f] > transformResponse: [f] > transitional: {silentJSONParsing: true, forcedJSONParsing: true, clarifyJSONError: false} url: "http://localhost:4000/api/v1/register" > validateStatus: f validateStatus(status) xhrOptions: {} responseType: "json" [Symbol]: function() {} }, "data": { "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkbSI6ImFkbWUiLCJyb2I6ImFkbWUiLCJpdiI6IjEiLCJpcyI6ImFkbWUiLCJhdWQiOiJkaWQzIiwiaWF0IjoxNjYxMjA0MDAwfQ.", "enabled": true, "date": { "_id": "60c8e9d7f000000000000000", statusText: "OK", headers: AxiosHeaders, config: { "_id": "..." } } }, "request": XMLHttpRequest {readyStateString: null, readyState: 4, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, ...} status: 200 statusText: "OK" }</pre>
Data Administrat or Terdaftar	<pre>{ "name": "admin", email: "adminster@gmail.com", password: "superadmin", role_id: 1, enabled: "1", ...} about: "halo admin" email: "adminster@gmail.com" enabled: "1" last_name: "test" name: "admin" password: "superadmin" role_id: 1 }</pre>
Edit Data Administrat or	<pre>{ "success": Array(1), message: { "_id": "..." } "data": Array(7) > 0: { _id: 78, name: "superadmin superadmin", email: "superadmin", enabled: true, role: "Super Administrator" } > 1: { _id: 99, name: "fatih muhammad ridho", email: "fatih2@gmail.com", enabled: true, role: "Super Administrator" } > 2: { _id: 100, name: "admin admin", email: "admin", enabled: true, role: "Administrator" } > 3: { _id: 101, name: "andi jaward", email: "andiawardi@gmail.com", enabled: true, role: "Super Administrator" } > 4: { _id: 106, name: "halo amirah", email: "haloamir@gmail.com", enabled: false, role: "Super Administrator" } > 5: { _id: 108, name: "admin test", email: "admintest1@gmail.com", enabled: true, role: "Super Administrator" } > 6: { _id: 110, name: "admin test", email: "admintest@gmail.com", enabled: true, role: "Super Administrator" } length: 7 [[Prototype]]: Array(0) meta: {totalData: 8 pages: 7, totalData: 7, pages: 1, totalPages: 1} }</pre>
Delete Data Administrat or	<pre>{ "name": "superadmin", last_name: "superadmin", email: "superadmin", password: "superadmin", admin_role_id: 1, ...} about: "superadmin" admin_role_id: 1 email: "superadmin" enabled: 1 last_name: "superadmin" name: "superadmin" password: "superadmin" }</pre>
Edit Data Siswa	<pre>{ "date": { "_id": "60c8e9d7f000000000000000", statusText: "OK", headers: AxiosHeaders, config: { "_id": "..." } }, "config": { "> adapters: (2) [xhr, http] > baseUrl: "https://localhost:4000/api/v1" > data: undefined > env: { FormData: f, Blob: f } > Headers: AxiosHeaders {accept: application/json, text/plain: */*, Content-type: null} > bodyLength: -1 > maxContentLength: -1 > method: "delete" }, "data": { "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJkbSI6ImFkbWUiLCJyb2I6ImFkbWUiLCJpdiI6IjEiLCJpcyI6ImFkbWUiLCJhdWQiOiJkaWQzIiwiaWF0IjoxNjYxMjA0MDAwfQ.", "enabled": true, "date": { "_id": "60c8e9d7f000000000000000", statusText: "OK", headers: AxiosHeaders, config: { "_id": "..." } } }, "request": XMLHttpRequest {readyStateString: null, readyState: 4, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload, ...} status: 200 statusText: "OK" }</pre>

Tabel 2. menunjukkan hasil dari pengujian pada enkripsi data pengguna dengan memodifikasi data teks menggunakan metode Base64.

Plaintext	Hasil Encode	Hasil Decode
"id": 101, "name": "iqbal suwandi", "email": "iqbalsuwandi 20@gmail.co m", "enabled ":true, "role": "Super Administrator"	Id: ImlkJjogMT AxLA== Name: Im5hbWUiOiAiaXFhYwgc3V3YW5kaSIs Email: ImVtYWlsIjogImlxYmFsc3V3YW5kaTlwQGdtYWlsLmNvbSIs Enabled: ImVuYWJsZWQiOiB0cnVLA== Role: InJvbGUiOiAiU3VwZXIgc3V3YW5kaTlwYXRvcilI=	"id": 101, "name": "iqbal suwandi", "email": "iqbalsuwa ndi 20@gmail. co m", "enabl ed": true, "role": "Super Administrator "

Aplikasi web service ini diimplementasikan pada PT. Mojadi Aplikasi Indonesia. Setelah menganalisis hasil pengujian aplikasi, ditemukan beberapa kelebihan dan kekurangan dari aplikasi ini adalah sebagai berikut:

A. Kelebihan Aplikasi

- 1) Aplikasi ini menggunakan JWT (JSON Web Token) untuk mengamankan data pada aplikasi ini.
- 2) Dengan menggunakan web service, aplikasi ini *client* tidak langsung mengakses *database*, tetapi terlebih dahulu mengakses web service.
- 3) *Friendly user interface*.
- 4) Dapat menggunakan data siswa dan administrator dalam satu aplikasi.
- 5) Pengecekan data siswa dan administrator lebih cepat.
- 6) Tingkat keberhasilan dalam sistem dalam menyimpan data di platform android sebesar 97% , keberhasilan dalam menyimpan di website 98%, jika data yang diinput terdapat kesalahan pada jenis tipe data maupun panjang karakternya.

B. Kekurangan Aplikasi

- 1) Fungsi aplikasi ini tidak menyederhanakan semua proses seluruh bisnis PT. Mojadi Aplikasi Indonesia hanya menyangkut pada *Content Management System*.
- 2) Dari masing-masing aplikasi seperti web belum di

push ke web server dan pada *mobile* belum di *push* ke Google Play Store.

IV. KESIMPULAN DAN SARAN

Berdasarkan analisis pengumpulan data terhadap pengembangan aplikasi inidan percobaan yang dilakukan pada program aplikasi web service, dapat diambil kesimpulan dan saran yang harus diperhatikan untuk mendukung aplikasi tersebut di fase selanjutnya.

A. Kesimpulan

Berdasarkan perancangan, pembuatan, serangkaian uji coba, dan analisis perangkat lunak dari aplikasi ini, dapat ditarik beberapa kesimpulan yaitu:

- 1) Semua web service yang dibuat dapat diimplementasikan dengan baik pada setiap aplikasi seperti web dan *mobile*.
- 2) Hal ini membuat PT. Mojadi Aplikasi Indonesia lebih mudah dioperasikan dengan menggunakan web service yang memiliki kemampuan interoperabilitas sehingga dapat digunakan oleh aplikasi yang berjalan pada platform yang berbeda.
- 3) Menampilkan data dengan mudah hanya memerlukan parameter URL yang dibuat di web service.

B. Saran

Saran yang mungkin diperlukan agar aplikasi ini bekerja lebih baik lagi antara lain:

- 1) Spesifikasi kebutuhan aplikasi harus dipenuhi agar aplikasi berfungsi dengan baik dan lancar.
- 2) Dalam penelitian selanjutnya, diharapkan adanya fitur yang memungkinkan terhadap sistem belajar online terhadap customer.
- 3) Pada penelitian selanjutnya diharapkan dapat menampilkan user interface yang menarik

DAFTAR PUSTAKA

- [1] R. Choirudin and A. Adil, "Implementasi Rest Api Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 18, no. 2, pp. 284–293, 2019, doi: 10.30812/matrik.v18i2.407.
- [2] T. Choiri, N. Dengen, and Islmiah, "Web Service untuk Web Profil SMP Negeri 2 Rantepao Toraja Utara," vol. 2, no. 1, 2017.
- [3] S. Laugi, "Sistem Informasi berbasis Web dalam Penyelenggaraan Lembaga Pendidikan," *Shautut Tarb.*, vol. 24, no. 1, p. 109, 2018, doi: 10.31332/str.v24i1.939.
- [4] I. Yuslin Anugrah, R. Reza El Akbar, A. Rahmatulloh, J. Siliwangi Nomor, and J. Barat, "SISTEMASI: Jurnal Sistem Informasi Monitoring Nilai Akademik Siswa Sekolah Dasar Berbasis Android dengan Restful Web Service Monitoring Elementary School Students Achievement using Android-based RESTful Web Service," vol. 11, pp.

- 84–96, 2022, [Online]. Available: <http://sistemasi.ftik.unisi.ac.id>
- [5] Rio Bayu Sentosa, “MEMBANGUN WEB KONTEN MANAJEMEN SISTEM SECARA DINAMIS DENGAN BAHASA PEMROGRAMAN PHP FRAMEWORK CODEIGNITER DENGAN DATABASE MARIADB,” *Intecom J. Inf. Technol. Comput. Sci.*, vol. 1, no. 2, pp. 212–223, 2018, [Online]. Available: <https://journal.ipm2kpe.or.id/index.php/INTECOM/article/download/295/184/>
- [6] R. Gunawan and A. Rahmatulloh, “Gunawan Rahmatulloh, 2019,” vol. 5, no. 1, pp. 74–79, 2019.
- [7] R. C. Saputra and Y. D. Setianto, “Web Service Security System Analysis with Rest Architecture Using The Aes Method with JWT,” *Praxis (Bern. 1994)*, vol. 3, no. 1, p. 76, 2020, doi: 10.24167/praxis.v3i1.2826.
- [8] A. Umarjati and A. Wibowo, “Implementasi JWT pada Aplikasi Presensi dengan Validasi Fingerprint,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 1, no. 10, pp. 1085–1091, 2021, [Online]. Available: <https://doi.org/10.29207/resti.v4i6.2650>
- [9] A. Setiawan and A. I. Purnamasari, “Implementasi JSON Web Token Berbasis Algoritma SHA-512 untuk Otentikasi Aplikasi BatikKita,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 6, pp. 4–10, 2020, doi: 10.29207/resti.v4i6.2533.
- [10] D. Trianggara, W. Witanti, and H. Ashaury, “Integrasi Data Barang Pada Sistem Inventory Cv. Xyz Menggunakan Web Service,” *JIKA (Jurnal Inform.)*, vol. 6, no. 3, p. 324, 2022, doi: 10.31000/jika.v6i3.6840.
- [11] I. Tahyudin and Zidni Iman Sholihati, “Pengembangan Aplikasi Tiga-Tingkat Menggunakan Metode Scrum pada Aplikasi Presensi Karyawan Glints Academy,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 1, pp. 169–176, 2022, doi: 10.29207/resti.v6i1.3793.
- [12] H. Putria, L. H. Maula, and D. A. Uswatun, “Analisis Proses Pembelajaran dalam Jaringan (DARING) Masa Pandemi Covid- 19 Pada Guru Sekolah Dasar,” *J. Basicedu*, vol. 4, no. 4, pp. 861–870, 2020, doi: 10.31004/basicedu.v4i4.460.
- [13] W. G. Wardhana, I. Arwani, and B. Rahayudi, “Implementasi Teknologi Restful Web Service Dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis Website (Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya),” *J. Pengemb. Teknol. Inf. dan Ilmu Komputer; Vol 4 No 2*, vol. 4, no. 2, pp. 680–689, 2020, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/7024%0Ahttp://j-ptiik.ub.ac.id>
- [14] A. N. Septandy and F. A. Fauzi, “Sejarah Web Service dan Contoh Penggunaannya,” *Coll. Student*, no. March, 2020.
- [15] A. Hidayat and Pristiwanto, “Implementasi Algoritma Base64 Untuk Verifikasi Qr Code Login Jaringan Wifi Berbasis Android,” *J. Sist. Komput. dan Inform. Hal*, vol. 2, no. 1, pp. 25–30, 2020, doi: 10.30865/json.v2i1.2468.