

IMPLEMENTASI WEB SERVICE SISTEM INFORMASI AKADEMIK BERBASIS MOBILE (STUDI KASUS CYBERCAMPUS UAM)

Putri Ariatna Alia¹, Agung Teguh Setyadi², Erik Yohan Kartiko³, Rusina Widha Febriana⁴, Rony Kriswibowo⁵, Muhammad Fajrul Falah⁶.

^{1,6}Politeknik Elektronika Negeri Surabaya, Departemen Teknik Multimedia Kreatif, Teknologi Game

²Politeknik Elektronika Negeri Surabaya, Departemen Teknik Informatika dan Komputer. Teknik Informatika PSDKU Lamongan

³Universitas Jember, Fakultas Ilmu Komputer, Informatika

^{4,5}Universitas Anwar Medika, Fakultas Sains dan Teknologi, Sistem Informasi

ariatna@pens.ac.id¹, agungteguhsetyadi@pens.ac.id²

erikyohan@unej.ac.id³, rusina.widha@uam.ac.id⁴

rkriswibowo@gmail.com⁵, falah@pens.ac.id⁶

*coresponding author : ariatna@pens.ac.id

Received: 28-04- 2026

Revised: 20-05-2026

Approved: 26-05-2026

ABSTRAK

Perkembangan teknologi informasi menuntut institusi pendidikan menyediakan layanan akademik yang cepat, akurat, dan mudah diakses. Universitas Anwar Medika memerlukan sistem informasi akademik berbasis mobile yang mampu meningkatkan fleksibilitas akses bagi mahasiswa dan dosen. Penelitian ini bertujuan mengimplementasikan RESTful Web Service sebagai media integrasi antara sistem Cybercampus berbasis web dengan aplikasi mobile sehingga layanan akademik dapat diakses secara real-time tanpa melakukan perubahan pada sistem backend yang telah ada. Penelitian menggunakan metode pengembangan Waterfall yang meliputi tahap analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan evaluasi. Web Service dikembangkan menggunakan framework Laravel dengan arsitektur REST dan format pertukaran data JSON, sedangkan aplikasi mobile dibangun menggunakan platform Kodular. Pengujian dilakukan melalui black-box testing, beta testing menggunakan skala Likert, serta pengujian kinerja sistem yang meliputi response time, throughput, uptime, dan kompatibilitas aplikasi mobile. Hasil penelitian menunjukkan bahwa sistem berhasil mengintegrasikan layanan akademik ke dalam aplikasi mobile dengan baik. Pengujian kinerja menunjukkan response time sebesar 280 ms, page load time 1,82 detik, uptime 99,87%, throughput 168 request/detik, serta error rate sebesar 0,14%. Selain itu, hasil beta testing memperoleh nilai rata-rata di atas 4,30 pada seluruh indikator, yang menunjukkan bahwa aplikasi diterima dengan baik oleh pengguna. Dengan demikian, implementasi Web Service berbasis REST mampu meningkatkan aksesibilitas layanan akademik secara efektif, responsif, dan andal.

Kata Kunci — REST API, Mobile Application, Web Service, Kodular, Sistem Informasi Akademik.

PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi telah mendorong transformasi digital di berbagai sektor, termasuk pendidikan tinggi. Perguruan tinggi dituntut untuk meningkatkan kualitas layanan akademik melalui pemanfaatan sistem informasi yang efektif, efisien, dan mudah diakses [1]. Sistem informasi akademik tidak hanya berfungsi sebagai media pengelolaan data akademik, tetapi juga menjadi sarana utama penyampaian informasi kepada mahasiswa, dosen, dan masyarakat. Seiring meningkatnya penggunaan

perangkat bergerak (mobile devices), pengguna mengharapkan akses informasi yang cepat, fleksibel, dan dapat dilakukan kapan saja. Penelitian [2] menunjukkan bahwa kualitas interaktivitas website perguruan tinggi berpengaruh signifikan terhadap kecepatan pengguna dalam menemukan informasi. Website dengan struktur navigasi dan interaktivitas yang baik mampu meningkatkan efisiensi pencarian informasi serta memberikan pengalaman pengguna yang lebih baik [2]. Hasil tersebut menunjukkan bahwa peningkatan kualitas layanan digital pada perguruan tinggi tidak hanya ditentukan oleh kelengkapan informasi, tetapi juga oleh kemudahan akses dan desain sistem yang mendukung kebutuhan pengguna.

Beberapa penelitian telah memanfaatkan teknologi Representational State Transfer (REST) Web Service sebagai media integrasi antara sistem backend dan aplikasi client. RESTful API diimplementasikan pada sistem informasi akademik perguruan tinggi untuk mendukung pengembangan aplikasi yang mampu mengakses data akademik secara terintegrasi [3]. Hasil penelitian menunjukkan bahwa penggunaan REST API mampu mempermudah pertukaran data antarplatform serta meningkatkan interoperabilitas sistem, namun penelitian tersebut masih berfokus pada pengembangan layanan akademik tanpa mengevaluasi kinerja layanan secara menyeluruh [4]. Selanjutnya, [5] mengembangkan integrasi antara Sistem Informasi Akademik dengan platform Moodle menggunakan REST API sehingga proses sinkronisasi data akademik dapat dilakukan secara otomatis. Meskipun demikian, penelitian tersebut lebih berorientasi pada integrasi sistem e-learning dan belum membahas implementasi aplikasi mobile berbasis no-code [6]. Penelitian lain mengenai implementasi REST API pada sistem informasi juga menunjukkan bahwa pendekatan berbasis layanan mampu meningkatkan fleksibilitas komunikasi antara frontend dan backend, mempermudah proses pengembangan aplikasi, serta mendukung interoperabilitas antarplatform [7],[8].

Universitas Anwar Medika (UAM) telah mengimplementasikan sistem informasi akademik berbasis web bernama Cybercampus untuk mendukung aktivitas akademik dan administrasi. Namun, sistem tersebut masih memiliki keterbatasan dalam hal mobilitas karena hanya dapat diakses secara optimal melalui browser pada perangkat desktop maupun perangkat mobile. Kondisi ini menyebabkan mahasiswa dan dosen belum memperoleh pengalaman penggunaan yang optimal, terutama ketika membutuhkan akses layanan akademik secara cepat dan real-time melalui aplikasi mobile. Selain itu, pengembangan aplikasi mobile secara konvensional umumnya memerlukan perubahan pada arsitektur backend, penambahan sumber daya pengembang, serta waktu implementasi yang relatif panjang sehingga kurang efisien bagi institusi yang telah memiliki sistem informasi akademik yang berjalan dengan baik.

Berdasarkan kajian penelitian terdahulu, sebagian besar penelitian berfokus pada pembangunan aplikasi baru atau integrasi dengan platform tertentu [9], [10], [11]. Sementara itu, penelitian mengenai pemanfaatan Web Service sebagai media integrasi antara sistem informasi akademik yang telah berjalan dengan aplikasi mobile tanpa melakukan rekayasa ulang terhadap backend masih relatif terbatas. Kondisi tersebut menunjukkan adanya research gap, yaitu belum optimalnya pemanfaatan sistem akademik eksisting untuk

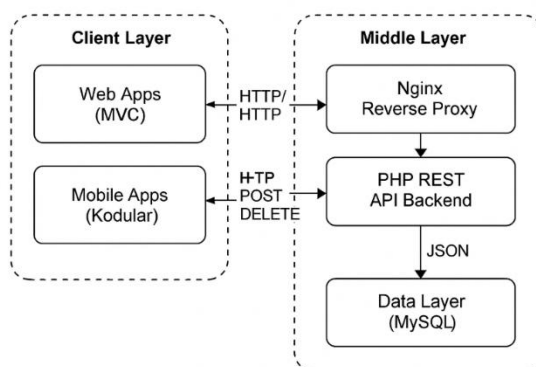
mendukung transformasi layanan akademik berbasis mobile secara efisien, cepat, dan berkelanjutan.

Untuk mengatasi permasalahan tersebut, penelitian ini mengusulkan implementasi Web Service berbasis REST sebagai lapisan integrasi antara sistem backend Cybercampus dan aplikasi mobile. Web Service memungkinkan pemisahan antara lapisan presentasi (frontend) dan logika bisnis (backend) sehingga layanan yang telah tersedia pada sistem web dapat digunakan kembali melalui mekanisme REST API tanpa perlu melakukan pengembangan ulang sistem secara menyeluruh. Aplikasi mobile dikembangkan menggunakan platform Kodular yang menerapkan konsep no-code development, sehingga proses pengembangan menjadi lebih cepat dan efisien.

Kebaruan (novelty) penelitian ini terletak pada penerapan REST API sebagai lapisan integrasi adaptif antara sistem Cybercampus yang telah beroperasi dengan aplikasi mobile berbasis no-code tanpa melakukan perubahan terhadap arsitektur backend yang telah digunakan sebelumnya. Selain mengimplementasikan integrasi layanan akademik, penelitian ini juga melakukan evaluasi kinerja sistem berdasarkan parameter response time, throughput, uptime, dan error rate, serta mengevaluasi tingkat penerimaan pengguna melalui beta testing. Berdasarkan hal tersebut, penelitian ini bertujuan mengimplementasikan Web Service berbasis REST sebagai media integrasi antara sistem informasi akademik Cybercampus dan aplikasi mobile, serta mengevaluasi kinerja dan penerimaan pengguna terhadap sistem yang dihasilkan sehingga dapat menjadi solusi yang efektif dalam mendukung transformasi layanan akademik berbasis mobile di lingkungan perguruan tinggi [12].

METODE PENELITIAN

Pemilihan metode merupakan salah satu tahapan penting dalam pengembangan sistem karena harus disesuaikan dengan karakteristik permasalahan dan tujuan penelitian. Kesesuaian metode dengan karakteristik data memberikan pengaruh terhadap kualitas hasil yang diperoleh [13]. Oleh karena itu, penelitian ini menggunakan pendekatan pengembangan sistem dengan metode client server architecture yang mengintegrasikan sistem informasi akademik berbasis web dengan aplikasi mobile. Sistem yang dikembangkan terdiri dari dua komponen utama, yaitu aplikasi pada sisi klien dan aplikasi pada sisi server [14]. Objek penelitian dalam studi ini adalah sistem informasi akademik Cybercampus Universitas Anwar Medika, sedangkan subjek penelitian adalah pengguna sistem yang terdiri dari mahasiswa dan dosen yang terlibat dalam proses pengujian aplikasi mobile. Arsitektur sistem yang dikembangkan mengadopsi konsep layered architecture yang terdiri dari tiga lapisan utama, yaitu client layer, middle layer, dan data layer. Pada client layer, aplikasi mobile dikembangkan menggunakan platform no-code Kodular berbasis Android yang berfungsi sebagai antarmuka pengguna [15]. Aplikasi ini memungkinkan pengguna untuk mengakses layanan akademik seperti jadwal kuliah, nilai, dan informasi lainnya secara real-time.



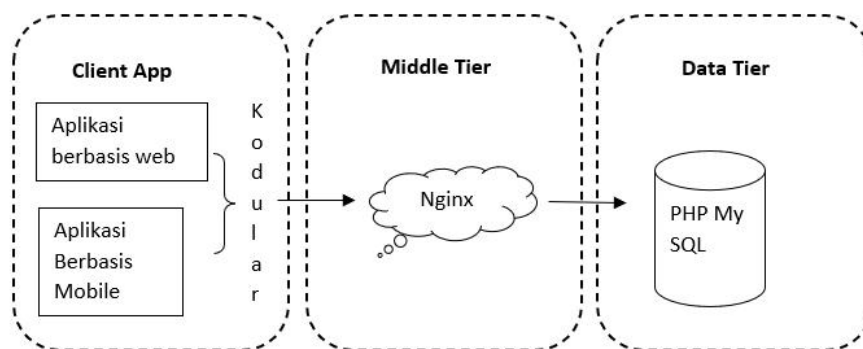
Gambar 2.1 Arsitektur Aplikasi

Pada gambar 2.1 Arsitektur Aplikasi pada middle layer, sistem menggunakan REST API yang dikembangkan menggunakan bahasa pemrograman PHP dengan framework Laravel dan dijalankan di atas Nginx sebagai web server sekaligus reverse proxy. REST API berfungsi sebagai penghubung antara aplikasi client dan sistem backend, serta menangani proses komunikasi data menggunakan metode HTTP seperti GET, POST, PUT, dan DELETE dalam format JSON. Sementara itu, data layer menggunakan database MySQL yang merupakan basis data utama dari sistem Cybercampus. Layer ini menyimpan seluruh data akademik seperti data mahasiswa, nilai, kehadiran, dan KRS. Proses pengolahan data dilakukan melalui skrip backend yang terintegrasi dengan REST API untuk memastikan data dapat diakses secara aman dan efisien. Untuk implementasi sistem, server ditempatkan pada lingkungan berbasis Linux menggunakan layanan VPS (Virtual Private Server). Nginx dikonfigurasi untuk mendukung keamanan melalui protokol SSL/TLS serta meningkatkan performa sistem melalui mekanisme reverse proxy dan load balancing. Tahapan penelitian dilakukan melalui beberapa langkah, diantaranya Analisis kebutuhan sistem, dengan mengidentifikasi kebutuhan integrasi antara sistem web Cybercampus dan aplikasi mobile. Perancangan sistem, meliputi desain arsitektur client server dan perancangan REST API. Implementasi sistem, yaitu pengembangan REST API dan aplikasi mobile menggunakan Kodular. Pengujian sistem, dilakukan dengan mengukur kinerja sistem menggunakan parameter response time, throughput, dan uptime, serta pengujian penerimaan pengguna (beta testing) [16], [17]. Evaluasi sistem, dilakukan berdasarkan hasil pengujian kinerja dan tingkat kepuasan pengguna.

Dengan pendekatan ini, sistem Cybercampus yang telah ada dapat diintegrasikan ke dalam platform mobile tanpa perlu melakukan rekayasa ulang backend secara menyeluruh, sehingga lebih efisien dari sisi waktu dan sumber daya. Sistem tetap menggunakan basis data yang ada saat ini dalam CyberCampus, seperti MySQL atau SQL Server. Layer ini menyimpan semua informasi penting seperti data mahasiswa, nilai, kehadiran, KRS, dan lainnya. Pada bagian middle layer ini menyediakan layanan REST API yang akan digunakan untuk mengakses data dan logika bisnis. REST API dikembangkan menggunakan teknologi yang sesuai (seperti PHP, Node.js, atau ASP.NET Core),

dan dijalankan di atas Nginx. Nginx berperan sebagai web server dan juga reverse proxy yang meneruskan permintaan ke backend sesuai kebutuhan, serta mengelola keamanan dan performa layanan. Client Layer yang digunakan adalah menggunakan aplikasi android berbasis Kodular, yang menjadi antarmuka pengguna untuk mahasiswa dan dosen. Aplikasi ini memungkinkan pengguna login, melihat jadwal, nilai, dan layanan akademik lainnya. Kodular akan memanfaatkan API endpoint dari server Nginx untuk mengambil dan mengirim data.

Untuk hosting dan pengelolaan server, sistem dapat ditempatkan di layanan cloud seperti VPS (Virtual Private Server) atau container (Docker) dengan sistem operasi Linux agar optimal saat menjalankan Nginx sebagai web server [18], [19]. Dengan konfigurasi Nginx yang tepat, sistem dapat menangani berbagai permintaan client secara efisien, cepat, dan aman. Dengan pendekatan ini, Universitas Anwar Medika tidak perlu membangun ulang sistem CyberCampus secara penuh. Cukup dengan mengembangkan antarmuka mobile melalui Kodular dan menyediakan REST API di server Nginx, maka layanan akademik dapat dijangkau lebih luas melalui perangkat seluler. Pada sistem ini terdapat pembagian layer seperti pada gambar 2.2.



Gambar 2.2 Pembagian layer

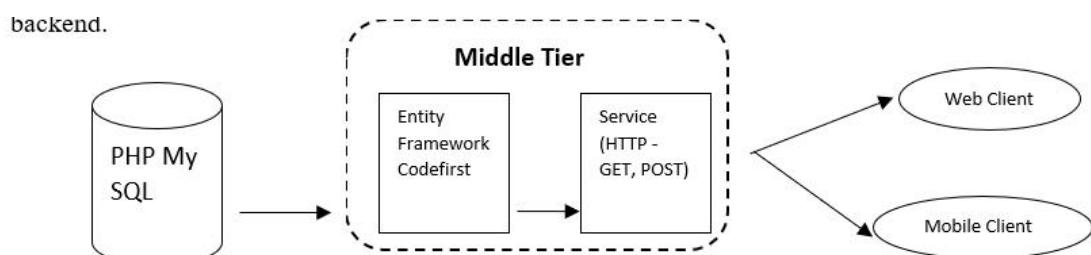
Pada arsitektur sistem yang dikembangkan, lapisan Client App memegang peran penting sebagai antarmuka utama antara pengguna dengan sistem. Aplikasi client berfungsi untuk menampilkan data, menerima masukan dari pengguna, serta mengirimkan permintaan ke server melalui jaringan internet. Dalam implementasi ini, digunakan dua jenis aplikasi client, yaitu aplikasi berbasis web dan aplikasi berbasis mobile. Aplikasi web dikembangkan untuk memberikan akses dari peramban (browser), sedangkan aplikasi mobile dibangun menggunakan platform Kodular, yang memungkinkan pembuatan aplikasi Android secara visual tanpa perlu menulis kode pemrograman secara manual. Kedua jenis aplikasi ini terhubung dengan sistem backend melalui layanan web, sehingga mampu menampilkan dan memperbarui data secara dinamis berdasarkan interaksi pengguna. Lapisan Middle Tier berfungsi sebagai jembatan antara aplikasi client dengan server backend. Middle tier menggunakan Nginx, sebuah web server sekaligus reverse proxy yang menerima permintaan dari client (baik dari aplikasi web maupun aplikasi mobile Kodular) dan meneruskannya ke server backend untuk diproses. Nginx juga dapat dikonfigurasi untuk meningkatkan keamanan sistem, seperti

mengaktifkan SSL/TLS agar komunikasi antar layer terenkripsi serta membatasi akses ke endpoint tertentu menggunakan firewall [18]. Selain itu, Nginx juga mendukung load balancing jika sistem dikembangkan lebih lanjut untuk mendistribusikan beban permintaan ke beberapa server backend secara efisien [20].

Sementara itu, lapisan Data Tier terdiri dari skrip PHP sebagai logika backend dan database MySQL sebagai tempat penyimpanan data. Ketika client mengirimkan permintaan (seperti login, simpan data, atau ambil data), Nginx akan meneruskannya ke skrip PHP. Skrip tersebut kemudian akan memproses permintaan tersebut, termasuk melakukan validasi dan manipulasi data sesuai kebutuhan. Seluruh data yang dibutuhkan atau diubah akan diambil dari atau disimpan ke dalam database MySQL yang telah dirancang dengan skema terstruktur dan relasi antar tabel yang sesuai. Misalnya, saat pengguna melakukan login melalui aplikasi Kodular, data username dan password akan dikirim ke endpoint tertentu, lalu Nginx meneruskan permintaan tersebut ke skrip PHP. Selanjutnya, PHP akan mencocokkan informasi login tersebut dengan data pada tabel users di MySQL. Jika data cocok, maka PHP akan mengirimkan respons kembali ke aplikasi melalui Nginx dalam format JSON untuk ditampilkan ke pengguna. Proses ini memastikan data dapat diproses dan dikembalikan secara real-time dan aman.

Perancangan Lapisan Penghubung Antara Client Dan Server

Perancangan lapisan penghubung antara client dan server merupakan tahap penting dalam arsitektur sistem yang bertugas menjembatani komunikasi data antara aplikasi pengguna (client) dan backend (server). Pada sistem ini, lapisan penghubung berfungsi untuk menerima permintaan dari client seperti login atau pengambilan data, lalu meneruskannya ke skrip backend (misalnya PHP) yang akan memproses permintaan tersebut dan berinteraksi dengan database. Peran lapisan ini sangat penting untuk memastikan aliran data dari dan ke client dapat berjalan secara efisien, aman, dan terstruktur. Dengan adanya lapisan penghubung ini, pengguna dapat mengakses informasi secara real-time tanpa perlu berinteraksi langsung dengan komponen server di sisi backend.



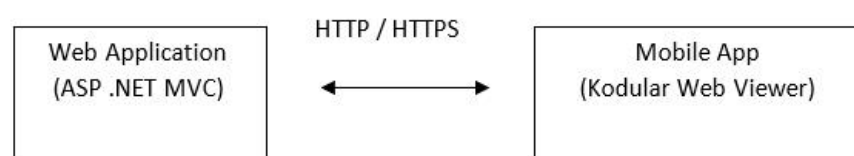
Gambar 2.3 Perancangan Arsitektur penghubung client dan server

Bagan tersebut menggambarkan arsitektur sistem tiga lapis (three-tier architecture) yang terdiri dari lapisan data (data tier), lapisan tengah (middle tier), dan lapisan pengguna (client tier). Pada lapisan data, digunakan kombinasi PHP dan MySQL sebagai fondasi backend, di mana seluruh data sistem seperti informasi pengguna, konten, dan transaksi disimpan dan dikelola. Data tersebut kemudian diakses melalui lapisan tengah yang berperan

sebagai penghubung antara database dan aplikasi client. Dalam middle tier, terdapat komponen yang menyerupai Entity Framework CodeFirst yang berfungsi sebagai pemetaan data antara struktur database dan objek aplikasi (dalam konteks PHP bisa digantikan oleh ORM seperti Eloquent). Selanjutnya, data dikirim atau diambil melalui layanan web (service) yang mendukung protokol HTTP seperti GET dan POST. Layanan ini bertugas menangani permintaan dari klien baik dalam bentuk pengambilan maupun pengiriman data. Di sisi client, terdapat dua jenis pengguna, yaitu web client dan mobile client, yang dapat mengakses layanan tersebut melalui antarmuka aplikasi mereka. Alur proses dimulai dari client yang mengirimkan permintaan ke layanan API, kemudian diteruskan ke database melalui lapisan penghubung, dan hasilnya dikirim kembali ke client dalam format yang dapat ditampilkan, seperti JSON. Arsitektur ini dirancang untuk memastikan komunikasi antar komponen sistem berjalan aman, efisien, dan terstruktur.

Perancangan Aplikasi Client.

Pada sistem ini, terdapat dua jenis aplikasi client yang dirancang untuk mengakses layanan REST Web Services. Aplikasi client pertama merupakan aplikasi berbasis web yang dikembangkan menggunakan teknologi ASP.NET MVC, sedangkan aplikasi client kedua adalah aplikasi berbasis mobile yang dikembangkan menggunakan Kodular, sebuah platform pemrograman visual berbasis blok yang memudahkan pembuatan aplikasi Android tanpa harus menulis kode secara manual. Untuk aplikasi web, proses komunikasi dengan Web Services dilakukan melalui Web API Client Library, yang memungkinkan pengiriman permintaan HTTP seperti GET dan POST. Sementara itu, aplikasi mobile yang dibuat dengan Kodular memanfaatkan komponen seperti Web Component untuk mengirim permintaan HTTP ke server dan menerima respons dalam format JSON. Data yang diterima kemudian diproses menggunakan blok-blok pemrosesan teks di Kodular untuk menampilkan informasi secara dinamis di dalam aplikasi. Pendekatan ini memungkinkan kedua aplikasi, baik web maupun mobile, berinteraksi secara real-time dengan sistem backend melalui antarmuka layanan web yang telah disediakan.



Gambar 2.4 Arsitektur Perancangan Aplikasi Dengan Client

Arsitektur perancangan aplikasi dengan client menggambarkan bagaimana aplikasi dibangun dengan membagi peran antara komponen client, middle tier, dan server (data tier). Dalam arsitektur ini, aplikasi client bertugas sebagai antarmuka yang digunakan langsung oleh pengguna, baik melalui perangkat mobile maupun web. Komponen client akan mengirimkan permintaan (request) ke middle tier yang berfungsi sebagai perantara antara client dan database. Middle tier biasanya dibangun menggunakan layanan web (web service) berbasis arsitektur REST, yang menangani permintaan data dari

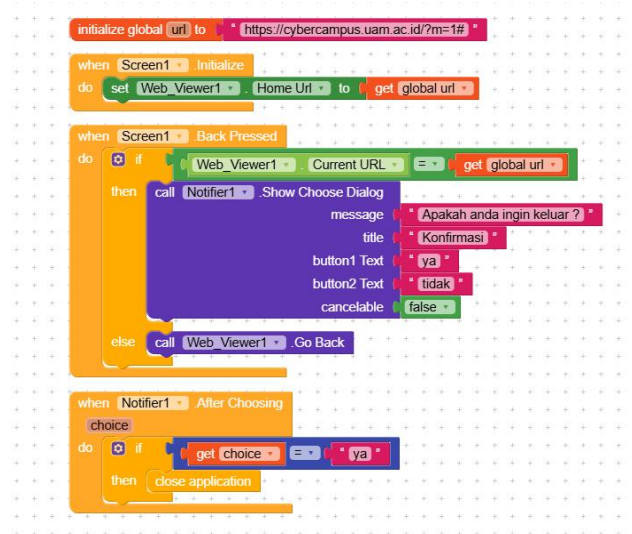
client melalui metode HTTP seperti GET, POST, PUT, dan DELETE. Selanjutnya, middle tier akan meneruskan permintaan tersebut ke server, yang berisi database dan logic backend (misalnya PHP-MySQL). Setelah diproses, hasil data dikembalikan ke client melalui format ringan seperti JSON. Dengan pendekatan ini, arsitektur sistem menjadi lebih modular, mudah dikembangkan, serta memungkinkan akses data secara real-time tanpa harus membebani client dengan logika bisnis atau manajemen data kompleks. Pendekatan ini umum digunakan dalam aplikasi modern untuk memastikan skalabilitas, efisiensi, dan keamanan komunikasi data.

HASIL DAN PEMBAHASAN

Setelah melalui tahapan analisis kebutuhan dan perancangan sistem, bagian ini akan membahas implementasi dari sistem yang telah dikembangkan. Pembahasan difokuskan pada komponen utama yang membentuk arsitektur aplikasi, mulai dari desain client hingga integrasi dengan web service melalui lapisan penghubung (middle layer). Setiap komponen dijelaskan berdasarkan fungsinya dalam alur komunikasi data antara aplikasi mobile dan server, serta bagaimana Kodular digunakan untuk membangun antarmuka mobile yang terhubung secara dinamis dengan data dari server. Pembahasan ini juga mencakup struktur data, alur proses, serta teknologi yang digunakan dalam setiap lapisan sistem.

Penerapan Web Service

Dalam penerapan Web Service pada Kodular, proses komunikasi antara aplikasi mobile dan server dilakukan menggunakan komponen Web yang tersedia di dalam Kodular. Komponen ini memungkinkan aplikasi melakukan permintaan (request) ke API atau server melalui protokol HTTP menggunakan metode seperti GET dan POST. Fungsi Web Service sangat penting dalam sistem ini karena menjadi jembatan antara aplikasi mobile dan server penyedia data. Controller seperti pada ASP.NET Web API, di Kodular kita memanfaatkan blok-blok pemrograman visual untuk menentukan URL tujuan, parameter yang dikirim, serta bagaimana menangani respons dari server. Setelah server memberikan respons dalam format JSON, data tersebut kemudian diolah menggunakan komponen JSON Text Decode untuk menampilkan informasi yang diinginkan ke dalam tampilan aplikasi. Untuk menampilkan data mahasiswa dari server, kita cukup mengatur URL endpoint API ke komponen Web, lalu menggunakan blok Web1.Get atau Web1.PostText sesuai kebutuhan. Respons dari server kemudian diuraikan menggunakan blok JSON agar dapat ditampilkan secara dinamis di aplikasi mobile. Dengan pendekatan ini, Kodular mampu menjadi platform mobile yang ringan namun tetap bisa terintegrasi dengan Web Service secara efisien. Seperti gambar 3.1



Gambar 3.1 Blok pada kodular

Inisialisasi Url

Proses menetapkan atau menyimpan alamat sebuah website ke dalam variabel pada awal program, agar alamat tersebut bisa digunakan secara berulang dalam aplikasi tanpa harus diketik ulang setiap kali dibutuhkan. Dalam konteks aplikasi mobile, misalnya di Kodular, URL (Uniform Resource Locator) digunakan untuk menentukan alamat halaman web yang akan ditampilkan dalam komponen Web Viewer.



Gambar 3.2 Inisialisasi URL.

Blok ini digunakan untuk menyimpan alamat website CyberCampus ke dalam sebuah variabel global bernama url. Variabel global ini berfungsi sebagai tempat penyimpanan data yang dapat digunakan berulang kali di seluruh bagian aplikasi. Dengan mendefinisikan URL pada satu tempat, Anda tidak perlu menulis ulang alamat tersebut setiap kali ingin mengaksesnya—cukup dengan memanggil get global url.

Inisialisasi Layar Awal

Inisialisasi layar awal merupakan proses yang dijalankan secara otomatis saat aplikasi pertama kali dibuka. pada tahap ini, aplikasi akan mengeksekusi perintah untuk menampilkan halaman utama yang telah disimpan dalam variabel global url ke dalam komponen web viewer. dengan demikian, pengguna langsung diarahkan ke tampilan halaman web yang diinginkan, tanpa perlu melakukan tindakan manual. proses ini bertujuan untuk memberikan pengalaman pengguna yang efisien dan praktis sejak awal penggunaan aplikasi.

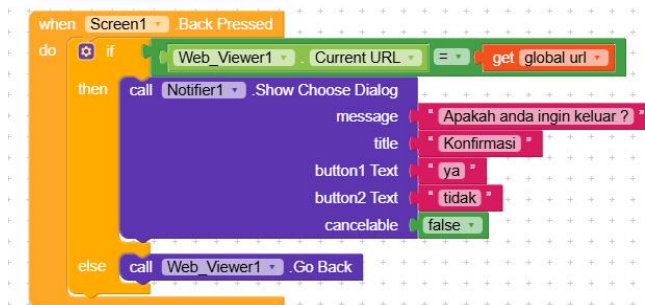


Gambar 3.3 Blok Inisialisasi Layar Awal.

Blok kode when Screen1.Initialize do set Web_Viewer1.HomeUrl to get global url digunakan untuk mengatur tampilan awal aplikasi saat pertama kali dijalankan. Ketika layar utama (Screen1) dimulai, perintah ini secara otomatis mengarahkan komponen Web Viewer untuk membuka halaman web yang telah ditentukan sebelumnya dalam variabel global url. Dalam konteks ini, URL yang dimaksud adalah alamat situs CyberCampus, sehingga begitu aplikasi dibuka, pengguna langsung melihat tampilan halaman CyberCampus dalam format mobile tanpa perlu mengetik alamat atau menavigasi secara manual.

Implementasi Web Service

Web Viewer digunakan untuk membuka dan menampilkan konten dari URL tertentu, seperti halaman website kampus, portal akademik. komponen dalam platform pengembangan aplikasi (seperti Kodular, MIT App Inventor, dan sejenisnya) yang berfungsi untuk menampilkan halaman web langsung di dalam aplikasi mobile. Komponen ini bekerja mirip seperti browser mini yang tertanam dalam aplikasi.

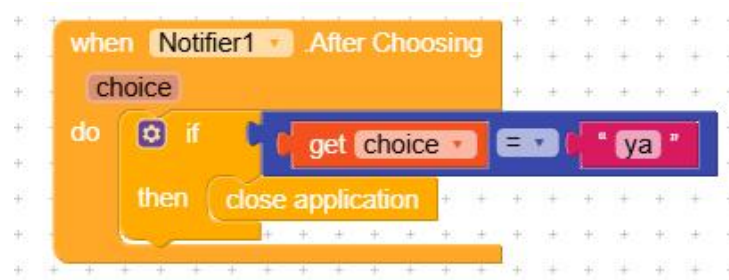


Gambar 3.4 Blok Implementasi Webviewer Untuk Memasukkan ke Mobile.

Blok ini menangani aksi ketika tombol kembali ditekan. Terdapat dua kemungkinan. Jika user berada di halaman utama (Current URL = global url), akan muncul konfirmasi keluar aplikasi. Jika tidak, maka user diarahkan ke halaman sebelumnya dalam Web Viewer (fungsi Go Back). Pemberitahuan ketika tekan pilihan kembali akan muncul notifier “Apakah anda ingin keluar?”.

Implementasi Navigasi Dan Konfirmasi Keluar Aplikasi

Implementasi navigasi dan konfirmasi keluar aplikasi merupakan bagian penting dalam desain antarmuka pengguna agar memberikan pengalaman penggunaan yang lebih ramah dan terkontrol. Dalam konteks aplikasi berbasis Kodular, navigasi dikendalikan melalui event Back Pressed pada Screen1. Saat pengguna menekan tombol kembali, sistem akan memeriksa apakah halaman yang sedang ditampilkan adalah halaman awal (berdasarkan URL). Jika iya, maka aplikasi tidak langsung ditutup, melainkan menampilkan dialog notifikasi konfirmasi menggunakan komponen Notifier.



Gambar 3.5 Implementasi Navigasi dan Konfirmasi Keluar Aplikasi

Blok program di atas merupakan bagian dari pengendalian aksi setelah pengguna memilih opsi dari dialog notifikasi (blok Notifier1.After Choosing). Fungsi utama dari blok ini adalah untuk menangani konfirmasi keluar aplikasi. Saat pengguna menekan tombol "kembali" di layar utama (Screen1), akan muncul dialog konfirmasi melalui komponen Notifier yang menampilkan pilihan "ya" atau "tidak". Blok ini mengecek apakah pilihan yang dipilih pengguna adalah "ya". Jika benar, maka perintah close application akan dijalankan, yang artinya aplikasi akan ditutup.

Hasil Aplikasi Mobile Dengan Fitur Web Service Di Kodular



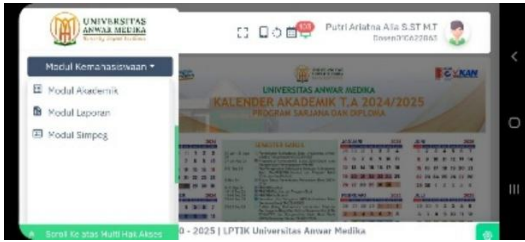
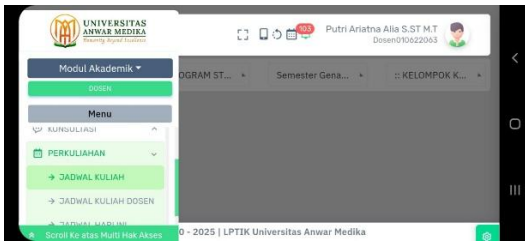

Hasil penelitian ini adalah berupa aplikasi mobile dengan nama UAM1 yang merupakan hasil implementasi Web Service dari sistem informasi berbasis web www.cybercampus.uam.ac.id milik Universitas Anwar Medika. Aplikasi ini dikembangkan dengan pendekatan arsitektur layanan terbuka, di mana fitur-fitur utama dari website diubah menjadi layanan modular yang dapat diakses langsung oleh aplikasi mobile melalui RESTful API. Gambar 3.6 ini menjadi bukti bahwa pendekatan Web Service memungkinkan transformasi sistem informasi kampus menjadi lebih fleksibel, terintegrasi, dan mudah diakses oleh pengguna melalui perangkat bergerak, tanpa harus membangun ulang sistem dari awal.



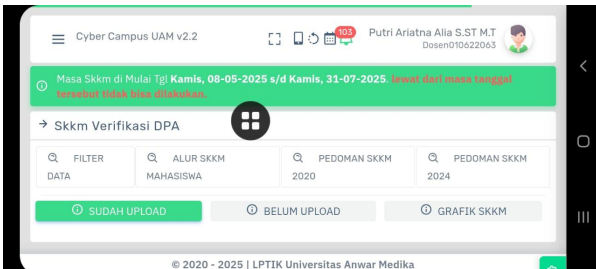


Gambar 3.6 Mobile App UAM1

Hasil dari aplikasi mobile ini menunjukkan bahwa layanan-layanan kampus yang sebelumnya hanya tersedia dalam versi web kini dapat diakses secara praktis melalui perangkat mobile. Implementasi Web Service memungkinkan integrasi data secara real-time, sehingga pengguna dapat mengakses informasi akademik dengan cepat dan efisien. Kinerja dan fungsionalitas dari masing-masing fitur dalam aplikasi ini telah diuji untuk memastikan kehandalan sistem. Selengkapnya, hasil pengujian terhadap tiap fitur akan dijelaskan pada Tabel 3.8.

Tabel 3.7 Test Plan

No	Fitur	Tampilan	Keterangan
1.	Tampilan awal aplikasi		Aplikasi siap di install untuk digunakan pada versi mobile.
2.	Tampilan awal login		Aplikasi telah berhasil ter install, lalu masuk ke halaman awal untuk login Cybercampus.
3.	Halaman Dashboard		Dashboard mobile website ini terdiri dari 3 fitur, diantaranya modul akademik, modul laporan dan laporan simpeg.
4.	Fitur Modul Akademik		Isi pada modul akademik fitur untuk melakukan pengaturan jadwal kuliah dosen, jadwal kuliah dan hari ini.
5.	Fitur Jadwal Kuliah		Pengaturan jadwal kuliah dengan menambahkan jumlah pertemuan.

No	Fitur	Tampilan	Keterangan
6.	Fitur Jadwal Dosen		Menampilkan jadwal kuliah dosen utuh sesuai dengan surat tugas yang diinput bagian akademik.
7.	Fitur Modul Kemahasiswaan		Modul kemahasiswaan mengatur persetujuan kegiatan kemahasiswaan terhadap dosen pengampu.
8	Fitur SKKM		Mahasiswa mengajukan surat pendukung mahasiswa, lalu di approve oleh dosen pembimbing akademik.

Pada Tabel 3.7 menampilkan dokumentasi antarmuka dan fitur dari aplikasi mobile berbasis web “Cybercampus” yang digunakan dalam sistem informasi akademik. Tampilan dimulai dari halaman login, di mana pengguna masuk ke sistem. Setelah login, pengguna diarahkan ke halaman dashboard yang terdiri dari tiga fitur utama: modul akademik, laporan, dan simpeg. Modul akademik mencakup fitur seperti pengisian KRS, jadwal kuliah, dan data ujian. Fitur jadwal kuliah memungkinkan pengaturan dan peninjauan jadwal perkuliahan, sedangkan fitur jadwal dosen menampilkan informasi detail jadwal mengajar dosen. Terakhir, modul kepegawaian menyajikan data kepegawaian dosen dan staf yang terhubung dengan sistem simpeg. Seluruh fitur dirancang untuk mempermudah pengelolaan akademik dan administratif di lingkungan kampus.

Tabel 3.8 BETA TESTING

No	Pertanyaan	Score					Total	Hasil
		1	2	3	4	5		
1.	Setelah menggunakan Cybercampus mobile, apakah lebih memudahkan ? (Karyawan)	0	0	12	21	37	70	4.357
2.	Setelah menggunakan Cybercampus mobile, apakah lebih memudahkan ? (Mahasiswa)	10	29	120	232	506	897	4.3034
3.	Aplikasi berhasil menampilkan halaman login tanpa kendala.	0	10	45	336	512	903	4.50
4.	Saya dapat login dengan mudah menggunakan akun saya.	0	9	34	309	529	903	4.54

5.	Semua fitur di dashboard dapat diakses dengan baik.	2	10	40	200	652	902	4.65
6.	Jadwal kuliah saya ditampilkan dengan akurat.	5	10	100	408	380	903	4.27
7.	Jadwal dosen ditampilkan lengkap dan sesuai.	8	15	140	293	446	902	4.28
8.	Aplikasi merespons dengan cepat saat saya berpindah menu.	6	20	30	109	738	903	4.72
9.	Saya ingin menggunakan aplikasi ini kembali di masa mendatang.	5	40	53	209	596	903	4.50
10.	Navigasi antar menu terasa mudah dan intuitif.	4	54	110	200	535	903	4.34

Berdasarkan hasil survei yang dilakukan terhadap sepuluh pertanyaan menggunakan skala Likert terhadap aplikasi Cybercampus Mobile, dapat disimpulkan bahwa aplikasi ini secara umum telah memberikan pengalaman penggunaan yang sangat memuaskan bagi mayoritas penggunanya, baik dari kalangan mahasiswa maupun karyawan. Rata-rata skor keseluruhan yang diperoleh berada di atas 4, yang mencerminkan bahwa mayoritas responden memberikan penilaian "Setuju" hingga "Sangat Setuju" terhadap berbagai aspek fungsionalitas dan kenyamanan penggunaan aplikasi. Salah satu indikator paling menonjol adalah pada pertanyaan mengenai kecepatan respons aplikasi saat berpindah menu, yang memperoleh skor tertinggi sebesar 4.72. Hal ini menunjukkan bahwa kecepatan dan performa sistem menjadi kekuatan utama Cybercampus Mobile dalam mendukung aktivitas akademik dan administrasi pengguna. Selain itu, fitur-fitur utama seperti kemudahan login, kelengkapan menu dashboard, dan aksesibilitas antarmuka juga memperoleh skor yang sangat tinggi, yang berarti sistem ini sudah cukup matang dari segi teknis dan desain antarmuka pengguna (user interface).

Meskipun begitu, ada beberapa aspek yang menunjukkan peluang untuk pengembangan lebih lanjut. Fitur jadwal kuliah dan jadwal dosen, meskipun memperoleh skor cukup baik (masing-masing 4.27 dan 4.28), cenderung berada di bawah rata-rata skor fitur lainnya. Ini mengindikasikan adanya potensi masalah minor seperti ketidakakuratan data, keterlambatan pembaruan informasi, atau tampilan yang kurang informatif bagi pengguna. Dari sisi kepuasan berdasarkan jenis pengguna, terlihat bahwa karyawan sedikit lebih merasa terbantu dengan adanya aplikasi ini dibandingkan mahasiswa, meskipun perbedaannya tidak terlalu signifikan. Yang menarik, lebih dari 4.900 responden memilih skor maksimal (5 – Sangat Setuju), yang mengukuhkan bahwa penerimaan terhadap aplikasi ini sangat positif.

Tabel 3.9 Hasil Pengujian Kinerja Website dan REST API CyberCampus

No	Parameter Pengujian	Deskripsi	Hasil Pengujian	Keterangan
1	Response Time	Waktu respons server saat permintaan GET ke endpoint utama (homepage & login).	280 ms	Sangat cepat, memenuhi standar < 500 ms.

2	Page Load Time	Waktu yang dibutuhkan halaman web (desktop & mobile) untuk dimuat sepenuhnya.	1.82 detik	Stabil dan responsif, memenuhi standar < 3 detik.
3	Availability (Uptime)	Persentase ketersediaan server selama 7 hari pengujian.	99.87%	Website berjalan lancar dan dapat diakses hampir tanpa gangguan.
4	Throughput	Jumlah permintaan yang dapat ditangani setiap detik.	168 request/s	Cukup tinggi untuk kebutuhan sistem akademik internal.
5	Error Rate	Persentase kesalahan HTTP (404, 500, 502) saat pengujian beban ringan.	0.14%	Hampir tanpa error—stabil dan dapat diandalkan.
6	Concurrent User Test	Simulasi 50 pengguna aktif mengakses menu dashboard secara bersamaan.	Berjalan lancar tanpa penurunan performa signifikan	Sistem mampu menangani beban normal pengguna harian.
7	API Performance (REST)	Pengujian kecepatan endpoint KHS, Jadwal Kuliah, dan Profil.	0.32 s (rata-rata)	API responsif dan cepat saat diakses aplikasi mobile.
8	Mobile Compatibility	Pengujian tampilan dan respons pada mobile browser dan mobile view.	Berjalan normal, layout adaptif	Siap diintegrasikan dengan aplikasi Kodular.

Hasil pengujian kinerja website menunjukkan bahwa sistem CyberCampus berjalan dengan stabil dan responsif ketika diakses melalui perangkat desktop maupun mobile. Rata-rata waktu respons berada pada angka 280 ms, yang menunjukkan performa server yang cepat. Uptime selama 7 hari pengujian mencapai 99,87%, dengan tingkat error yang sangat rendah yaitu 0,14%. Pengujian beban menggunakan simulasi 50 pengguna aktif menunjukkan bahwa website tetap berjalan lancar tanpa adanya penurunan performa signifikan. Selain itu, performa REST API yang digunakan untuk integrasi dengan aplikasi mobile Kodular juga menunjukkan hasil yang baik dengan rata-rata waktu respons sebesar 0,32 detik. Secara keseluruhan, hasil ini mengonfirmasi bahwa website CyberCampus siap digunakan sebagai backend utama dalam implementasi aplikasi mobile berbasis Web Service. Secara keseluruhan, aplikasi Cybercampus Mobile dinilai telah berhasil memenuhi ekspektasi pengguna dalam menunjang kebutuhan layanan akademik dan administrasi secara digital. Dengan mempertahankan kecepatan respon, kemudahan akses, dan kelengkapan fitur yang sudah ada, serta melakukan perbaikan minor pada aspek-aspek yang masih kurang maksimal, Cybercampus Mobile memiliki potensi besar

untuk terus menjadi sistem informasi terdepan dalam mendukung transformasi digital di lingkungan pendidikan tinggi.

KESIMPULAN

Penelitian ini bertujuan mengimplementasikan Web Service berbasis REST sebagai media integrasi antara sistem informasi akademik Cybercampus dan aplikasi mobile. Berdasarkan hasil implementasi dan pengujian, tujuan penelitian telah berhasil dicapai. Web Service yang dikembangkan mampu menghubungkan aplikasi mobile dengan sistem backend sehingga layanan akademik, seperti proses login, jadwal perkuliahan, nilai, dan informasi akademik lainnya, dapat diakses secara real-time tanpa memerlukan perubahan pada sistem Cybercampus yang telah ada.

Hasil pengujian menunjukkan bahwa sistem memiliki kinerja yang baik dengan response time sebesar 280 ms, page load time 1,82 detik, uptime 99,87%, dan error rate sebesar 0,14%. Selain itu, hasil beta testing memperoleh nilai rata-rata di atas 4,30, yang menunjukkan bahwa aplikasi diterima dengan baik oleh pengguna. Dengan demikian, implementasi Web Service berbasis REST menjadi solusi yang efektif untuk memperluas akses layanan akademik melalui perangkat mobile sekaligus mempertahankan sistem backend yang telah digunakan sebelumnya.

Dengan demikian, implementasi Web Service berbasis REST terbukti mampu mengintegrasikan sistem Cybercampus dengan aplikasi mobile secara efektif, meningkatkan aksesibilitas layanan akademik, serta dapat menjadi alternatif pengembangan aplikasi mobile bagi institusi pendidikan yang telah memiliki sistem informasi akademik berbasis web.

DAFTAR PUSTAKA

- [1] P. A. Alia, "LITERATURE REVIEW: LUNG DISEASE DETECTION BASED ON X-RAY USING ARTIFICIAL INTELLIGENCE," *Jurnal Rekayasa Sistem Informasi dan Teknologi*, vol. 1, no. 2, pp. 50–53, Nov. 2023, doi: 10.59407/JRSIT.V1I2.152.
- [2] A. T. Setyadi, P. A. Alia, E. Y. Kartiko, R. W. Febriana, and R. Kriswibowo, "The Impact of Website Interactivity on Users' Speed in Finding Information," *International Journal of Computer and Information System (IJCIS)*, vol. 6, no. 3, pp. 235–241, Jul. 2025, doi: 10.56327/ijcis.v6i3.245.
- [3] M. A. Novianto and S. Munir, "Analisis dan Implementasi RESTful API guna Pengembangan Sistem Informasi Akademik pada Perguruan Tinggi," *Jurnal Informatika Terpadu*, vol. 8, no. 1, pp. 47–61, 2022.
- [4] P. A. Alia, J. S. Prayogo, R. Kriswibowo, and A. T. Setyadi, "Implementation Open Artificial Intelligence ChatGPT Integrated With Whatsapp Bot," *Advance Sustainable Science, Engineering and Technology*, vol. 6, no. 1, p. 02401019, Jan. 2024, doi: 10.26877/ASSET.V6I1.17909.
- [5] A. Y. Pratama and J. A. Razaq, "Integrasi Sistem Informasi Akademik dan Elearning Moodle dengan REST API," *Jurnal Manajemen Informatika dan Sistem Informasi*, vol. 6, no. 1, pp. 26–38, 2023.
- [6] R. Kriswibowo, J. S. Prayogo, R. W. Febriana, and P. A. Alia, "Implementasi Black Box Testing dan Acceptance Testing Fitur SKKM pada Cybercampus.uam.ac.id Universitas Anwar Medika," *Jurnal Informatika*

- Universitas Pamulang*, vol. 8, no. 4, pp. 561–567, Dec. 2023, doi: 10.32493/INFORMATIKA.V8I4.36904.
- [7] rony kriswibowo, P. A. Alia, J. S. Prayogo, and R. W. Febriana, “Implementation of Text Processing Techniques on Citizen Opinions Regarding Floods in Surabaya,” *ELECTRON Jurnal Ilmiah Teknik Elektro*, vol. 5, no. 1, pp. 30–36, May 2024, doi: 10.33019/ELECTRON.V5I1.148.
- [8] P. Ariatna Alia, R. Widha Febriana, J. Suryo Prayogo, R. Kriswibowo, and C. Author, “Implementation Chatbot on Whatsapp Using Artificial Intelligence With Natural Language Processing Method,” *ELECTRON Jurnal Ilmiah Teknik Elektro*, vol. 5, no. 1, pp. 8–14, May 2024, doi: 10.33019/ELECTRON.V5I1.134.
- [9] H. Herfandi, M. Julkarnain, and M. Hanif, “Desain dan Implementasi RESTful Web Services untuk Integrasi Data dan Aplikasi,” *Jurnal Informatika Teknologi dan Sains*, vol. 4, no. 4, 2022.
- [10] R. Simbulan and J. Aryanto, “Implementasi REST API Web Services pada Aplikasi Sumber Daya Manusia,” *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, vol. 5, no. 1, 2024.
- [11] D. R. Salsabilla, “Implementasi REST-API untuk Sistem Informasi Administrasi di Redha Wedding Jakarta,” *Journal of Information Technology and Computer Science (INTECOMS)*, vol. 7, no. 2, pp. 319–325, 2024, doi: 10.31539/intecom.v7i2.7169.
- [12] P. A. Alia, D. K. Sari, N. Azis, B. G. Sudarsono, and P. A. Sucipto, “Implementation Artificial Intelligence with Natural Language Processing Method to Improve Performance of Digital Product Sales Service,” *Advance Sustainable Science, Engineering and Technology (ASSET)*, vol. 6, no. 3, pp. 0240301–0240301, Jun. 2024, doi: 10.26877/ASSET.V6I3.521.
- [13] E. Y. Kartiko, A. T. Setyadi, R. A. Atmoko, J. S. Prayogo, and P. A. Alia, “Comparative Study of Normalization Methods for DEMNAS Elevation Data to Grayscale Representation for 3D Terrain Visualization,” *International Journal of Multidisciplinary Sciences and Arts*, vol. 4, no. 3, pp. 149–155, Jul. 2025, doi: 10.47709/ijmdsa.v4i3.6717.
- [14] P. A. Alia, D. K. Sari, N. Azis, B. G. Sudarsono, and P. A. Sucipto, “Implementation Artificial Intelligence with Natural Language Processing Method to Improve Performance of Digital Product Sales Service,” *Advance Sustainable Science Engineering and Technology*, vol. 6, no. 3, pp. 0240301–0240301, Jun. 2024, doi: 10.26877/ASSET.V6I3.521.
- [15] P. Ariatna Alia, J. Suryo Prayogo, R. Kriswibowo, A. budi setyawan, R. Widha Febriana, and U. Anwar Medika, “PENGEMBANGAN KETERAMPILAN DESAIN INTERAKTIF DAN SERBAGUNA DALAM ERA SOCIETY 5.0 DENGAN MENGGUNAKAN CANVA,” *Jurnal Pengabdian Kolaborasi dan Inovasi IPTEKS*, vol. 2, no. 3, pp. 977–982, Jun. 2024, doi: 10.59407/JPKI2.V2I3.864.
- [16] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed. New York, NY, USA: McGraw-Hill, 2020.
- [17] I. Sommerville, *Software Engineering*, 10th ed. Boston, MA, USA: Pearson, 2016.
- [18] F5, “NGINX Reverse Proxy,” NGINX Documentation. [Online]. Available: <https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/>.

- [Accessed: Jul. 4, 2026].
- [19] V. George, "How to Deploy NGINX Reverse Proxy in Docker on Ubuntu 20.04," Atlantic.Net, Jun. 21, 2021. [Online]. Available: <https://www.atlantic.net/dedicated-server-hosting/how-to-deploy-nginx-reverse-proxy-in-docker-on-ubuntu-20-04/>. [Accessed: Jul. 4, 2026].
- [20] F5, "HTTP Load Balancing," NGINX Documentation. [Online]. Available: <https://docs.nginx.com/nginx/admin-guide/load-balancer/http-load-balancer/>. [Accessed: Jul. 4, 2026].