



Pengembangan Sistem Seleksi Proposal Penelitian Berbasis *Web Service* Menggunakan REST API

Daru Prasetyawan^{1,*}, Prihanto Dwi Rahmanto²

¹ UIN Sunan Kalijaga Yogyakarta; daru.prasetyawan@uin-suka.ac.id

² UIN Sunan Kalijaga Yogyakarta; prihanto.rahmanto@uin-suka.ac.id

* Korespondensi: daru.prasetyawan@uin-suka.ac.id

Sitasi: Prasetyawan, D.; Rahmanto, P. D. (2024). Pengembangan Sistem Seleksi Proposal Penelitian Berbasis *Web Service* Menggunakan REST API. JTIM: Jurnal Teknologi Informasi Dan Multimedia, 6(3), 283-295. <https://doi.org/10.35746/jtim.v6i3.585>

Diterima: 30-07-2024

Direvisi: 08-09-2024

Disetujui: 12-09-2024



Copyright: © 2024 oleh para penulis. Karya ini dilisensikan di bawah Creative Commons Attribution-ShareAlike 4.0 International License. (<https://creativecommons.org/licenses/by-sa/4.0/>).

Abstract: Web service-based application development has become a vital component of the modern information technology ecosystem. The implementation of web services facilitates interoperability among diverse systems and platforms. REST API (REpresentational State Transfer Application Programming Interface) is one of the most popular architectures for creating efficient and scalable web services. This study examines the implementation of REST API in the development of web service-based applications. The case study focuses on the creation of a Web service-based Research Proposal Selection System. The REST API implementation method in this study adheres to the system development life cycle (SDLC) to ensure that the developed API meets business needs while being efficient, secure, and reliable. The analysis phase is conducted to gather both functional and non-functional REST API requirements. During the design phase, the REST API architecture is designed in detail, including how data will be transferred and processed, and how the API will interact with databases and other systems. Furthermore, in the implementation phase, the REST API is developed, ensuring that each endpoint operates correctly. In developing the Proposal Selection System, the implementation of web services integrates data from various existing systems, including employee, academic, and research data. The system encompasses various features, such as proposal import, proposal mapping, proposal review, proposal assessment, proposal selection, and reporting. The research findings indicate that implementing the REST API simplifies the development of web service applications. The test results reveal that the average response time is relatively stable, measuring 31 ms without iterations and 29 ms with five iterations.

Keywords: Selection, Research, Web Service, REST

Abstrak: Pengembangan aplikasi berbasis *web service* telah menjadi elemen krusial dalam ekosistem teknologi informasi modern. Penerapan *web service* memungkinkan interoperabilitas antara berbagai sistem dan platform. REST API (*REpresentational State Transfer Application Programming Interface*) adalah salah satu arsitektur yang populer untuk membangun *web service* yang efisien dan skalabel. Penelitian ini membahas tentang implementasi REST API, dalam pengembangan aplikasi berbasis *web service*. Studi kasus dalam penelitian ini adalah pengembangan Sistem Seleksi Proposal Penelitian berbasis *web service*. Metode implementasi REST API dalam penelitian ini mengacu pada siklus hidup pengembangan sistem (SDLC) untuk memastikan API yang dikembangkan memenuhi kebutuhan bisnis, efisien, aman, dan dapat diandalkan. Tahap analisis dilakukan untuk mengumpulkan kebutuhan fungsional dan non-fungsional REST API. Pada tahap perancangan, arsitektur REST API dirancang secara detail, termasuk bagaimana data akan ditransfer dan diproses, serta bagaimana API akan berinteraksi dengan database dan sistem lainnya. Selanjutnya dalam tahap implementasi, REST API dikembangkan dan memastikan setiap *endpoint* berfungsi dengan benar. Dalam pengembangan Sistem Seleksi Proposal, penerapan *web service* dilakukan untuk

mengintegrasikan data dari berbagai sistem yang sudah tersedia sebelumnya, diantaranya data pegawai, akademik, dan penelitian. Sistem tersebut terdiri dari berbagai fitur, diantaranya impor proposal, mapping proposal, reviu proposal, penilaian proposal, seleksi proposal, dan laporan. Hasil penelitian menunjukkan bahwa penerapan REST API dapat memberikan kemudahan dalam pengembangan aplikasi *web service*. Dari hasil pengujian menunjukkan bahwa rata-rata waktu respon yang diperlukan relatif stabil, yaitu 31 ms pada pengujian tanpa iterasi dan 29 ms dengan 5 iterasi.

Kata kunci: Seleksi, Penelitian, Web Service, REST

1. Pendahuluan

Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) merupakan unsur akademik yang mengemban tugas untuk memfasilitasi, mengkoordinasi, mengawasi, dan mengevaluasi pelaksanaan penelitian dan pengabdian kepada masyarakat. Salah satu kegiatan dalam mengelola penelitian adalah menyeleksi dan menentukan diterimanya proposal yang diajukan oleh mahasiswa, dosen, dan tenaga fungsional lainnya. Permasalahan yang ada selama ini adalah proses seleksi proposal dilakukan secara manual karena sistem yang ada saat ini hanya digunakan untuk pendaftaran proposal saja. Proses seleksi dilakukan secara manual yang terkadang mengakibatkan kesalahan, seperti proposal yang diterima lebih dari satu pada klaster yang berbeda, sehingga harus ada revisi di kemudian hari. Pemerataan terhadap bidang kajian juga menjadi pertimbangan yang harus dilakukan dalam proses seleksi menjadi sulit dilakukan secara manual. Selain itu, integrasi data dengan sistem lain juga menjadi hal yang harus pertimbangan untuk mencegah terjadinya kesalahan data.

Di sisi lain, teknologi internet berkembang semakin pesat seiring semakin banyaknya pengguna internet di Indonesia. Internet tidak hanya digunakan untuk memperoleh informasi dan hiburan semata, tetapi internet sudah menjadi sebuah kebutuhan. Perkembangan internet juga memungkinkan pertukaran data dapat berlangsung dengan cepat, sehingga integrasi data dan informasi antar sistem yang berbeda dapat berlangsung dengan cepat juga. Saat ini, sebuah sistem informasi dituntut untuk dapat berkomunikasi dengan sistem lain. Komunikasi tersebut biasanya menggunakan sebuah format yang dapat dimengerti oleh berbagai sistem, sehingga sistem-sistem tersebut dapat saling berkomunikasi meskipun memiliki platform, sistem operasi, dan bahasa pemrograman yang berbeda. Integrasi aplikasi memungkinkan beberapa aplikasi dapat saling bertukar informasi secara *realtime*. Teknologi *web service* semakin memegang peranan yang signifikan dalam perkembangan sistem informasi. *Web service* digunakan untuk mengkoordinasi berbagai aplikasi web menggunakan langkah-langkah tertentu [1]. *Web service* merupakan kombinasi antara protokol terbuka dan protokol standar yang memungkinkan komunikasi antara klien dan *server* dengan menyediakan interoperabilitas antar aplikasi [2].

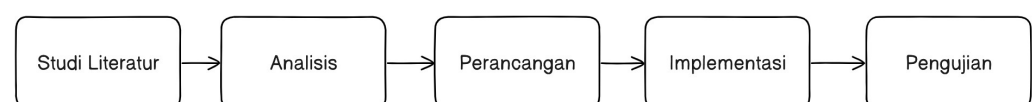
REST (*REpresentational State Transfer*) atau RESTfull merupakan sebuah arsitektur *web service* yang paling populer saat ini karena ringan dan memiliki kemampuan beradaptasi ke berbagai aplikasi web. REST merupakan arsitektur yang digunakan untuk merancang *service* yang dapat digunakan oleh berbagai *platform* untuk mendukung interoperabilitas [3]. REST memungkinkan mengandalkan URI dan *HTTP messages* untuk membangun aplikasi web yang memiliki kemampuan untuk saling bekerjasama dengan aplikasi lain [4]. REST API dapat juga disebut API (*Application Program Interface*) karena bertugas sebagai jembatan antar aplikasi untuk saling berkomunikasi. REST API dapat dipanggil dengan sebuah alamat yang dikenal dengan URI (*Uniform Resource Identifier*). API ini umumnya dapat diakses melalui HTTP (*Hypertext Transfer Protocol*) dengan menyertakan standar yang telah ditentukan seperti GET, POST, PUT, dan DELETE [5].

REST API menerapkan prinsip umum antar muka yang memungkinkan arsitektur sistem secara keseluruhan menjadi sederhana dan dapat meningkatkan visibilitas interaksi. REST API bersifat *client-server decoupling*. Aplikasi klien dan aplikasi *server* benar-benar terpisah. Aplikasi klien hanya mengetahui informasi yang dikirim dari sumber yang diminta melalui URI. Sebaliknya hanya memberikan informasi yang diminta dan tidak dapat mengubah apapun pada aplikasi klien. REST API juga harus bersifat *stateless*, artinya server tidak menyimpan informasi apapun terkait dengan permintaan klien. Oleh karena itu, klien harus menyertakan informasi yang lengkap setiap kali meminta permintaan kepada *server*.

Pada penelitian-penelitian sebelumnya, implementasi REST API telah banyak dilakukan dalam pengembangan sistem informasi. REST API telah digunakan untuk mengembangkan aplikasi penyedia informasi film dengan menyediakan berbagai macam informasi genre film dengan berbagai fitur [6]. REST API juga sering digunakan untuk membangun aplikasi *multi-platform* dalam usaha jasa yang memungkinkan pengguna dapat saling berinteraksi dengan sistem [7]. REST API juga digunakan dalam mengembangkan Sistem Informasi Terpadu untuk mengelola tugas-tugas dalam pengembangan sebuah sistem [8]. REST API juga dimanfaatkan untuk membantu dalam pengelolaan administrasi pasien klinik [9]. REST API juga dimanfaatkan sebagai *backend* untuk pembayaran *cashless* pada bisnis retail. Pada penelitian tersebut dibangun protokol data yang dapat diakses oleh aplikasi klien (*frontend*). Layanan yang disediakan oleh aplikasi *backend* antara lain sistem login, *register*, *withdraw*, *top-up balance*, transaksi antar pengguna, informasi saldo, riwayat transaksi, logout dan konfigurasi database itu sendiri [10]. Sebuah aplikasi Al-Qur'an online juga pernah dikembangkan dengan memanfaatkan REST API sebagai perantara antar aplikasi [11]. REST API dapat dikembangkan dengan berbagai bahasa pemrograman. Bahasa pemrograman berbasis Javascript sering digunakan dalam membangun REST API. Sebagai contoh, ExpressJS dapat digunakan untuk membangun REST API yang mendukung komunikasi data pada sebuah website [12]. NodeJS juga dapat digunakan untuk membangun REST API sebagai *backend* dalam mengembangkan aplikasi transaksi jasa elektronik [13]. Bahasa Go juga dapat dipakai untuk membangun REST API yang digunakan untuk mendukung aplikasi peminjaman ruangan [14]. Penelitian ini bertujuan untuk mengembangkan sebuah sistem seleksi bantuan penelitian dengan mengimplementasikan REST API dalam mengintegrasikan data, sehingga data yang sudah ada seperti data akademik dan data kepegawaian dapat digunakan kembali. Penelitian ini menghasilkan sebuah sistem informasi yang terintegrasi dengan sistem-sistem yang telah ada sebelumnya.

2. Bahan dan Metode

Metodologi yang digunakan dalam penelitian di dasarkan pada siklus hidup pengembangan perangkat lunak (*Software Development Life Cycle*) model *waterfall* yang terdiri dari tahap analisis kebutuhan, desain sistem, implementasi, serta pengujian dan evaluasi. SDLC diperlukan untuk meningkatkan kualitas perangkat lunak dan proses pengembangan perangkat lunak secara keseluruhan [15]. Tahapan penelitian dapat dilihat pada Gambar 1.

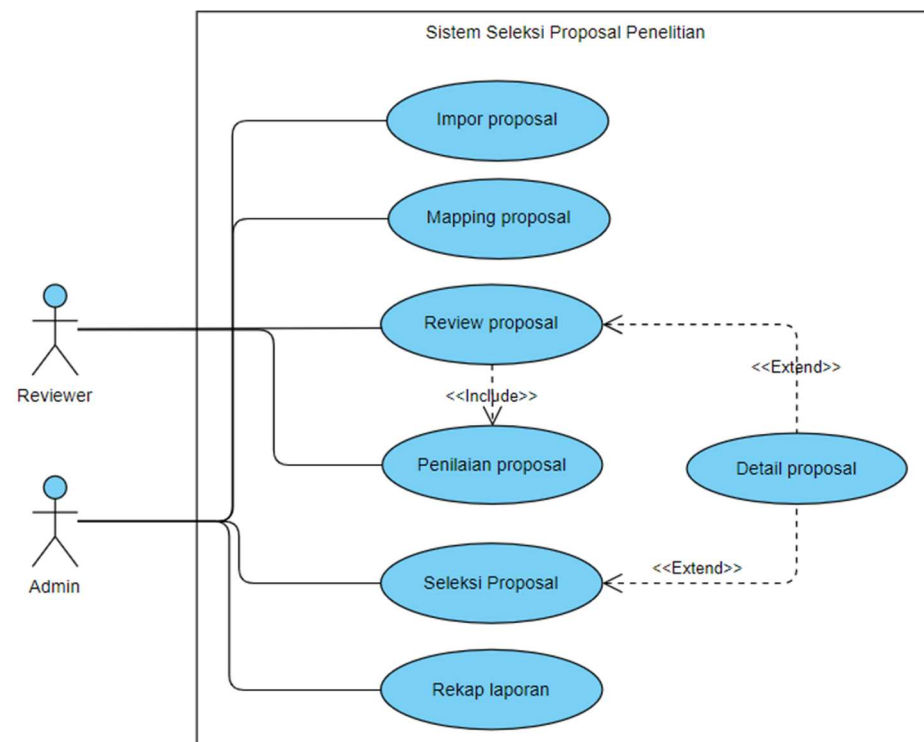


Gambar 1. Tahapan penelitian

2.1. Analisis

Tahap analisis bertujuan untuk mengumpulkan kebutuhan secara lengkap dan mendefinisikan kebutuhan yang harus dicapai oleh sistem. Tahap ini mendefinisikan

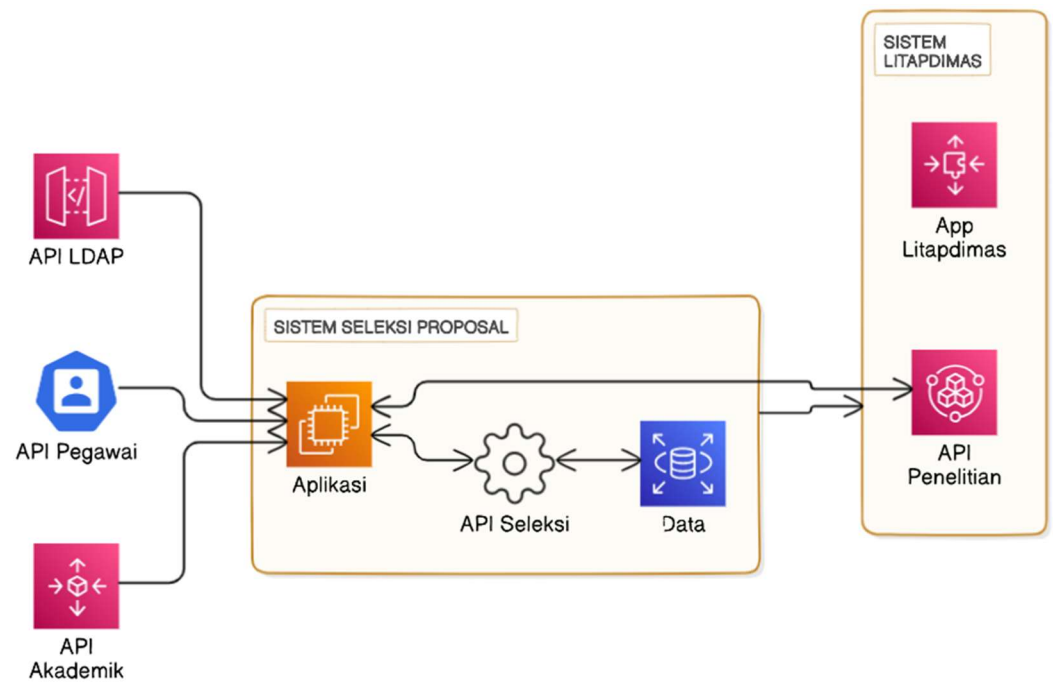
kebutuhan informasi, fungsi, perilaku, dan kinerja dari perangkat lunak [16]. Dalam tahapan analisis akan dilakukan peninjauan mengenai apa saja yang dibutuhkan dalam pengembangan sistem. Sistem Seleksi Penerimaan Proposal Bantuan Penelitian memiliki empat fitur utama, yaitu input/impor proposal, penilaian proposal, seleksi proposal, dan laporan hasil seleksi. Fitur-fitur tersebut dapat digunakan apabila pengguna sudah memiliki hak akses yang diperoleh dari API Pegawai. *Use Case* Sistem Seleksi Proposal Penelitian disajikan pada Gambar 2.



Gambar 2. Use Case Sistem Seleksi Proposal Penelitian

2.2. Perancangan Arsitektur

Perancangan arsitektur sistem digunakan untuk merepresentasikan secara konseptual komponen dan sub-komponen *software* yang akan dijalankan pada *hardware* yang ada di dalam sistem. Sistem Seleksi Proposal Penelitian menerapkan arsitektur berorientasi layanan atau dikenal dengan SOA (*Service Oriented Architecture*) dimana di dalam sistem terdiri dari berbagai komponen yang memberikan jenis layanan tertentu, seperti API LDAP, API pegawai, API Akademik, dan API Penelitian. API LDAP digunakan untuk memproses otentikasi data, sehingga data pengguna dan password serta proses validasi pengguna tidak perlu lagi dilakukan di aplikasi Seleksi Proposal Penelitian. API Akademik digunakan untuk menyediakan data kepegawaian, sehingga data yang berkaitan dengan kepegawaian tidak perlu lagi disimpan ke dalam database seleksi penelitian. API akademik digunakan untuk menyediakan data terkait akademik, seperti program studi. Sedangkan API Penelitian digunakan untuk menyediakan data tentang penelitian termasuk proposal penelitian. Arsitektur Sistem Seleksi Penerimaan Proposal Bantuan Penelitian disajikan pada Gambar 3.



Gambar 3. Arsitektur Sistem Seleksi Proposal Penelitian

2.3. Perancangan Proses

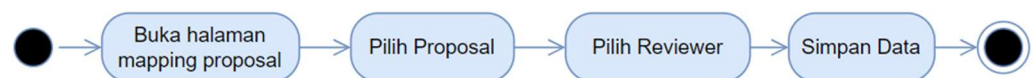
Perancangan proses menggambarkan proses yang terjadi di dalam sistem. Diagram aktivitas digunakan untuk membantu perancangan proses di dalam penelitian ini. Diagram aktivitas (*activity diagram*) digunakan untuk menggambarkan aspek dinamis dari sebuah sistem dengan memodelkan aliran dari satu aktivitas ke aktivitas lainnya.

Proses impor proposal diawali dengan membuka halaman proposal. Proposal yang sudah ada pada sistem Litapdimas (Penelitian dan Pengabdian kepada Masyarakat) dapat diintegrasikan ke dalam sistem seleksi proposal berdasarkan kluster yang telah ditentukan. Pengguna dapat mengambil data dari sistem Litapdimas melalui API Penelitian. Selanjutnya pengguna dapat memeriksa data proposal yang sudah diperoleh sebelum disimpan ke dalam basis data. Diagram aktivitas proses impor proposal penelitian dapat dilihat pada Gambar 4.



Gambar 4. Diagram Aktivitas Proses Impor Proposal

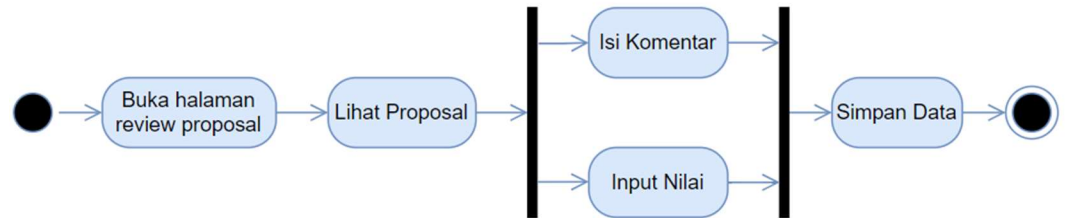
Proses *mapping* proposal adalah proses memetakan proposal dengan *reviewer* dengan tujuan untuk menentukan proposal mana saja yang akan direviu oleh reviewer tertentu. Proses tersebut dimulai dengan membuka halaman *mapping* proposal, memilih proposal dan reviewer yang ditunjuk, kemudian menyimpan data. Diagram aktivitas proses dapat dilihat pada Gambar 5.



Gambar 5. Diagram aktivitas proses mapping proposal

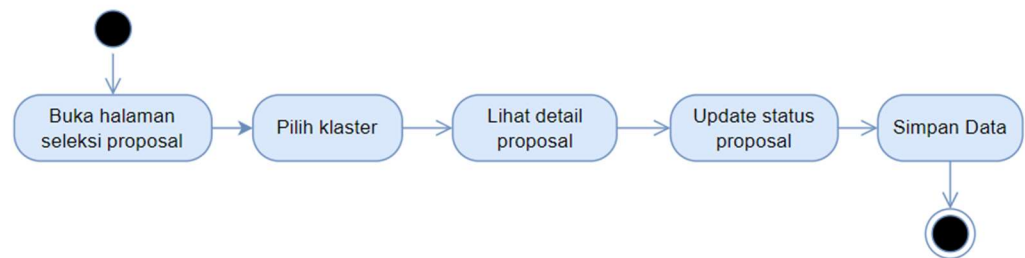
Proses reviu dan penilaian proposal merupakan proses untuk menentukan kelayakan proposal penelitian yang diajukan. Proses reviu dan penilaian dilakukan oleh reviewer

melalui halaman reviu proposal dan mereviu proposal satu per satu. Reviewer dapat memberikan penilaian dan komentar pada form penilaian. Diagram aktivitas proses penilaian proposal dapat dilihat pada Gambar 6.



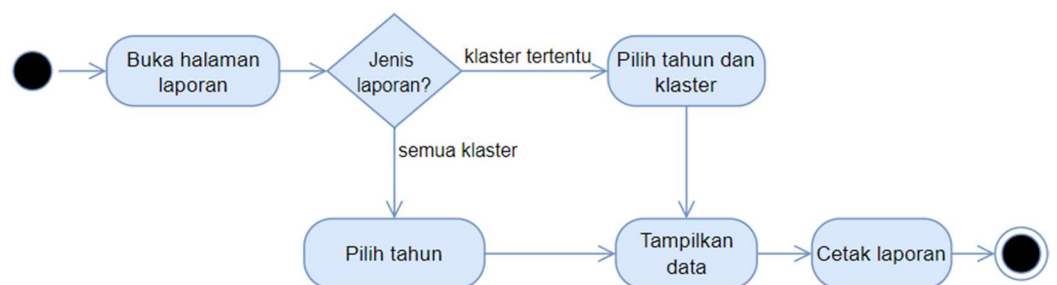
Gambar 6. Diagram aktivitas proses penilaian proposal

Proses seleksi proposal dimulai dengan membuka halaman seleksi proposal. Selanjutnya pengguna dapat memilih kluster sehingga akan menampilkan daftar proposal pada kluster tersebut. Pengguna dapat melihat secara detail informasi pada proposal tersebut seperti isi proposal dan nilai hasil reviu proposal tersebut. Untuk menentukan apakah proposal tersebut diterima atau tidak, pengguna dapat mengubah status proposal tersebut. Diagram aktivitas proses seleksi proposal dapat dilihat pada Gambar 7.



Gambar 7. Diagram aktivitas proses seleksi proposal

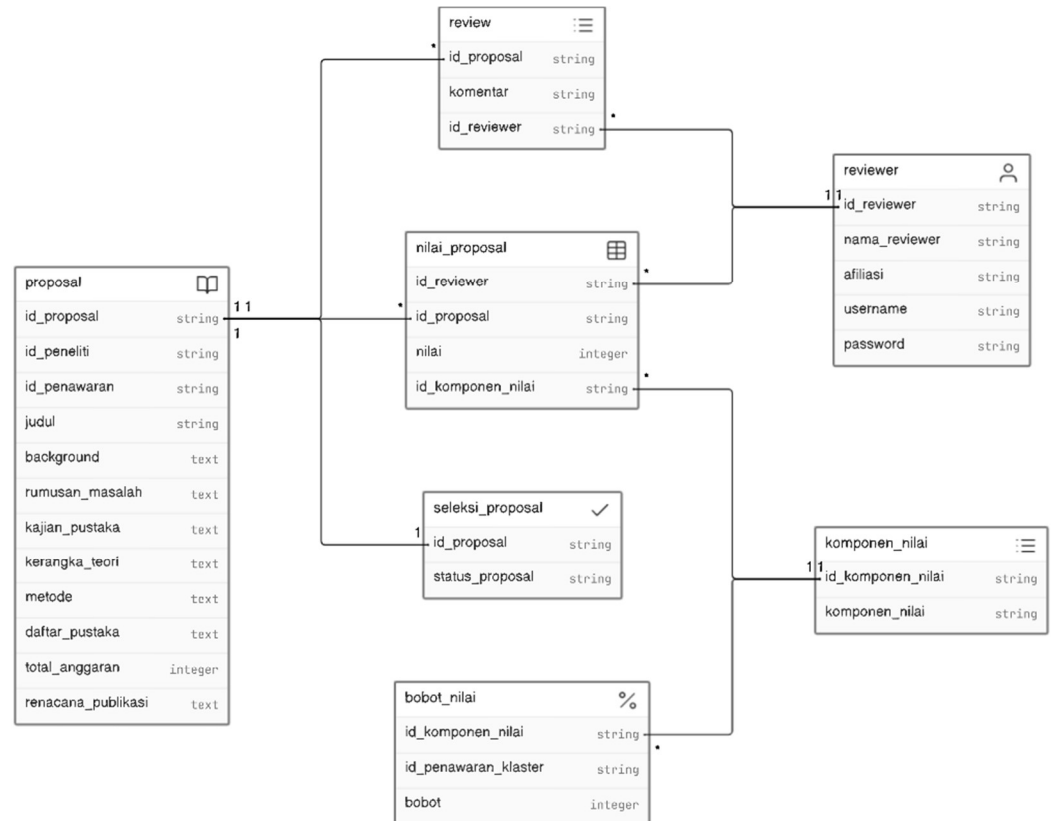
Proses rekap laporan digunakan untuk menyajikan hasil seleksi proposal penelitian. Proses rekap laporan dimulai dengan membuka halaman laporan. Selanjutnya pengguna dapat memilih jenis laporan yang diinginkan, yaitu semua proposal atau proposal dalam kluster tertentu. Jika memilih semua proposal, pengguna hanya diminta untuk memilih tahun saja, kemudian dapat menampilkan hasil seleksi proposal pada tahun tersebut. Sedangkan jika memilih proposal pada kluster tertentu, pengguna juga diharuskan memilih kluster tertentu dan menampilkan hasil seleksi proposal pada kluster yang dipilih. Selanjutnya, pengguna dapat mencetak atau mengunduh hasil seleksi dalam format pdf atau excel. Diagram aktivitas proses rekap hasil seleksi proposal dapat dilihat pada Gambar 8.



Gambar 8. Diagram Aktivitas Proses Rekap Hasil Seleksi

2.4. Pemodelan Data

Pemodelan data dilakukan untuk mempermudah dalam perencanaan penyimpanan data di dalam basis data. Model data Sistem Seleksi Proposal Penelitian dapat dilihat pada Gambar 9.



Gambar 9. Model data Sistem Seleksi Proposal Penelitian

2.5. Pengujian REST API

Pengujian dilakukan untuk mengukur performa dari REST API dan memastikan fungsionalitas dari REST API bekerja sesuai dengan yang direncanakan. Fungsionalitas API diuji untuk memastikan bahwa setiap *endpoint* API berfungsi sesuai dengan spesifikasinya. Pengujian dilakukan dengan mengirim request dengan data yang valid dan memeriksa apakah API memberikan respons yang benar. Pengujian fungsional ini juga menguji semua metode HTTP yang digunakan sesuai dengan fungsinya. Selain pengujian fungsional, pengujian juga dilakukan dengan menguji kinerja API di bawah kondisi beban yang berbeda, dengan memberikan volume permintaan yang tinggi. Pengujian performa ini dilakukan dengan cara mengirim sejumlah besar permintaan secara bersamaan untuk mengukur waktu respons dan stabilitas API.

Pengujian tersebut dilakukan menggunakan tool Postman dan Newman. API yang telah dibuat dikumpulkan dalam satu *collection* di dalam Postman. Penggunaan *collection* berguna untuk menyimpan permintaan (*requests*), pengaturan, dan skenario pengujian API dalam satu tempat yang terorganisir. Selanjutnya API yang sudah dikumpulkan dalam *collection* diuji bersama-sama menggunakan Postman. Skenario lain adalah dengan menggunakan iterasi pengujian. Pengujian dilakukan dalam waktu bersamaan dan diukur waktu respons rata-rata.

3. Hasil dan Pembahasan

3.1. Pengembangan REST API

REST API Seleksi dikembangkan menggunakan bahasa pemrograman PHP dan *framework* Codeigniter. Database yang digunakan adalah PostgreSQL. REST API Seleksi terdiri dari 5 servis, yaitu servis proposal, servis komponen nilai, servis nilai, dan servis seleksi. Setiap servis memiliki beberapa fungsi yang bertugas melayani permintaan data dari aplikasi klien, dalam hal ini adalah sistem seleksi proposal penelitian. Fungsi-fungsi yang terdapat di dalam servis-servis tersebut disajikan dalam Tabel 1.

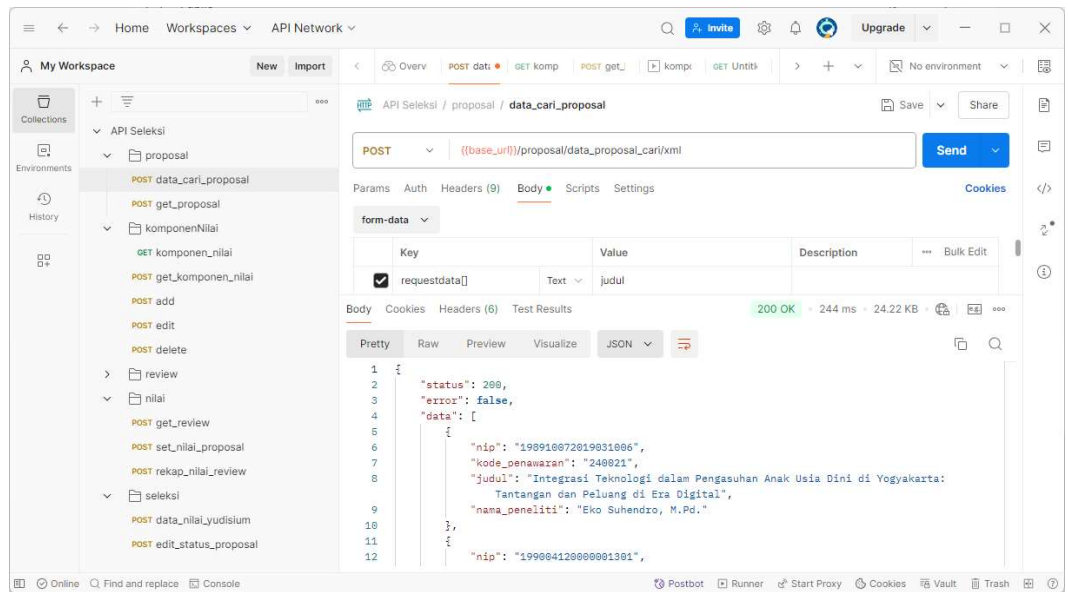
Tabel 1. Fungsi yang disediakan API Seleksi.

Servis	Fungsi	Method	Keterangan
proposal	data_cari_proposal	post	Menyediakan daftar proposal berdasarkan pencarian tertentu
	get_proposal	post	Menyediakan data proposal tertentu
	add	post	Memasukan proposal baru ke dalam data-base
komponen nilai	index	get	Menyediakan seluruh data komponen penilaian
	get_proposal	post	Menyediakan data komponen penilaian tertentu
	add	post	Memasukan komponen penilaian baru ke dalam database
	edit delete	post post	Memperbarui data komponen penilaian Menghapus data komponen penilaian
review	get_review	post	Menyediakan hasil revidi untuk proposal tertentu
	set_review	post	Menyimpan hasil revidi proposal tertentu
nilai	nilai_proposal	post	Menyediakan data nilai proposal dari reviewer tertentu
	set_nilai_proposal	post	Memasukan atau mengubah nilai proposal
	rekap_nilai_review	post	Menyediakan daftar proposal beserta nilai proposal
seleksi	data_nilai_yudisium	post	Menyediakan daftar proposal beserta nilai kumulatif
	edit_status_proposal	post	Mengubah status proposal

3.2. Implementasi dan Pengujian REST API

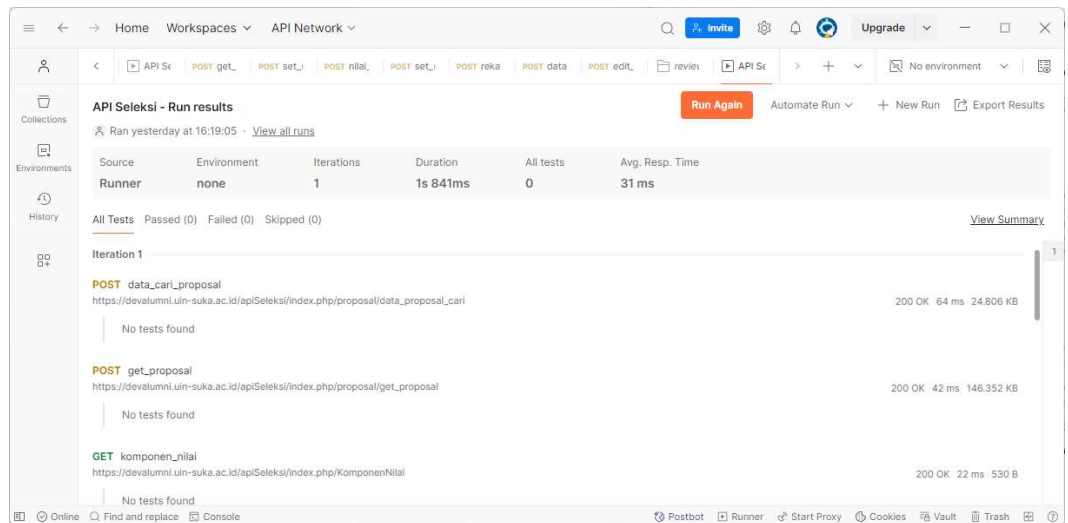
Implementasi merupakan tahapan untuk menerapkan analisis dan rancangan yang telah dibuat sebelumnya. Implementasi REST API sangat mudah dilakukan dengan menggunakan client URL (cURL). Dengan cURL, aplikasi klien dapat berinteraksi langsung dengan API. CURL akan menerjemahkan permintaan dari klien dan mengirimkannya melalui jaringan menggunakan metode yang telah ditentukan.

Pengembangan aplikasi menggunakan REST API menjadi lebih fleksibel karena setiap permintaan dari klien ke server bersifat independen dan tidak tergantung pada permintaan sebelumnya. Selain itu REST API mendukung berbagai format seperti JSON, XML, HTML, dan plain text, sehingga memungkinkan pengembang dapat memilih format yang cocok dengan aplikasi yang akan dikembangkan. Dalam pengembangan sistem seleksi proposal penelitian ini, format yang digunakan adalah JSON. JSON dipilih karena formatnya yang ringkas dan mudah dibaca oleh manusia maupun mesin. Contoh output atau respon dari API Seleksi dalam format JSON dapat dilihat pada Gambar 10.



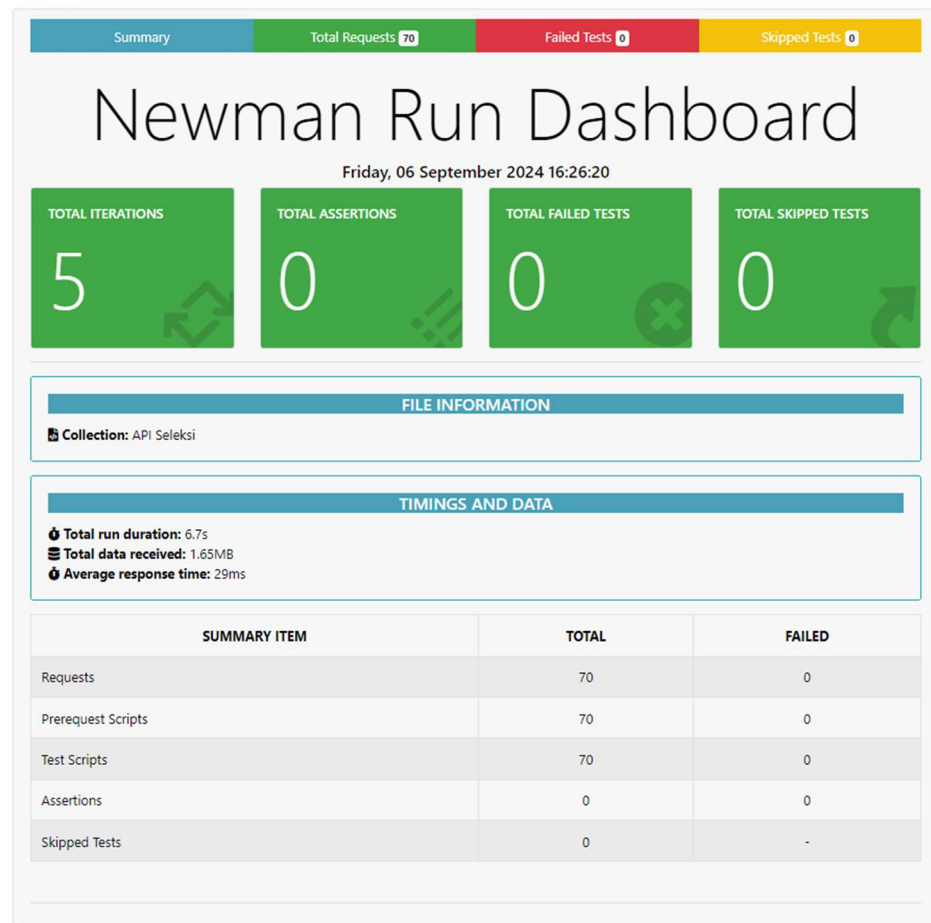
Gambar 10. Contoh output/respon dari API Seleksi dalam format JSON

Pengujian API yang dilakukan menggunakan Postman menunjukkan bahwa rata-rata waktu respon sebesar 31 ms. Hasil pengujian API menggunakan Postman dapat dilihat pada Gambar 11.



Gambar 11. Hasil pengujian API Seleksi menggunakan Postman

Selanjutnya pengujian API dilakukan dengan melakukan permintaan dengan sejumlah iterasi tertentu. Dari hasil pengujian dengan 5 iterasi menunjukkan bahwa semua API berhasil memberikan respon dengan rata-rata waktu respon 29 ms. Hasil pengujian dengan 5 iterasi menggunakan Newman ditunjukkan pada Gambar 11.



Gambar 12. Hasil pengujian API dengan Newman

3.3. Implementasi Sistem Seleksi Proposal

3.3.1. Halaman Proposal

Halaman proposal digunakan untuk menyajikan daftar proposal yang telah terdaftar pada sistem. Halaman proposal diimplementasikan menggunakan service proposal untuk menyediakan data proposal. Tampilan halaman proposal dapat disajikan pada Gambar 13.

No	Kode	Peneliti	Judul Penelitian	Jenis Penelitian	Telepon	Email	Status	Aksi
1	240122	Setia Rahmawan, M.Pd. (199306262020121005)	Journal of Tropical Chemistry Research and Education	Peningkatan Kualitas Jurnal international Bereputasi	087726961221	setia.rahmawan@uin-suka.ac.id	DIPROSES	Detail
2	240122	Ir. Muhammad Taufiq Nuruzaman, S.T. M.Eng., Ph.D. (197911182005011003)	JISKA (Jurnal Informatika Sunan Kalijaga)	Peningkatan Kualitas Jurnal international Bereputasi	087708181179	m.taufiq@uin-suka.ac.id	DIPROSES	Detail
3	240122	Dr. Muhammad Akmaluddin, M.S.I. (198912112020121007)		Peningkatan Kualitas Jurnal international Bereputasi	085878722025	muhammad.akmaluddin@uin-suka.ac.id	DIPROSES	Detail
4	240122	Prof. Dr. Hj. Na'imah, M.Hum (196104241990032002)		Peningkatan Kualitas Jurnal international	0274485840/081328163392	naimah@uin-suka.ac.id	DIPROSES	Detail

Gambar 13. Tampilan halaman proposal

3.3.2. Form Penilaian Proposal

Form komponen nilai digunakan untuk mengisikan data nilai proposal baru atau mengubah data nilai proposal yang sudah ada. Tampilan form penilaian proposal dapat dilihat pada Gambar 14.

Gambar 14. Tampilan form penilaian proposal

3.3.3. Halaman Seleksi Proposal

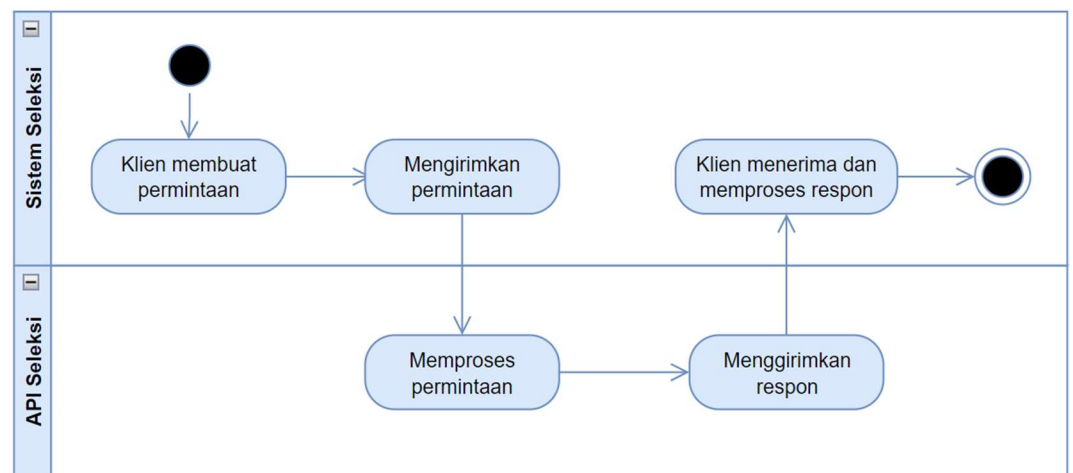
Halaman seleksi proposal digunakan untuk menentukan apakah suatu proposal diterima atau tidak. Halaman seleksi proposal diimplementasikan dengan meminta data yang disediakan oleh service seleksi. Tampilan halaman seleksi proposal dapat dilihat pada Gambar 15.

No	Peneliti	Judul	Nilai	Aksi
1	Daru Prasetyawan, S.T., M.Kom. NIP. 198610302011011005	Implementasi REST API untuk Pengembangan Sistem Seleksi Penerimaan Proposal Bantuan Penelitian	88.67	Diterima
2	Setia Rahmawan, M.Pd. NIP. 199306262020121005	Akreditasi Berbasis Web Program Studi Pendidikan Kimia UIN Sunan Kalijaga Yogyakarta	0	Terima
3	Dr. Sulistyaningih, S.Sos., M.Si. NIP. 197612242006042001	Optimalisasi Program Internasionalisasi pada Fakultas Ilmu Sosial dan Humaniora UIN Sunan Kalijaga	29.33	Terima
4	Adi Wirawan, S.Kom., M.Cs., NIP. 198603122011011009	PENGEMBANGAN LOAD BALANCING DENGAN HAPROXY UNTUK MENINGKATKAN KECEPATAN AKSES SISTEM INFORMASI AKADEMIK	0	Diterima
5	Dr. Amanah, M.A NIP. 198102140000002101	THE ROLE OF DIGITAL TECHNOLOGY AND INDONESIAN MUSLIM TRANSFORMATION: AN ANTHROPOLOGICAL STUDY	0	Terima

Gambar 15. Tampilan form penilaian proposal

3.4. Alur Pemrosesan Data

Pemrosesan data diawali dari klien yang membuat permintaan dan mengirimkannya ke server. Permintaan ini terdiri dari beberapa komponen, yaitu *endpoint* (URL), metode HTTP, *header*, dan *body*. *Endpoint* merupakan alamat lokasi API yang dituju. Metode merupakan tindakan untuk mengambil data, seperti GET, POST, PUT, dan DELETE. *Header* berisi informasi tambahan seperti otentikasi, tipe konten, dan sebagainya. *Body* berisi data yang akan dikirimkan ke server. Kemudian server memproses permintaan dari klien, seperti melakukan otentikasi dan otorisasi, validasi input, mengambil data dari database, dan penanganan kesalahan. Selanjutnya server mengirimkan respon ke klien yang berisi status HTTP dan *body*. Status HTTP mengindikasikan hasil permintaan yang dapat berupa status sukses (misal 200, 201, 202, dll) atau status error (misalnya 400, 401, 404, dll). *Body* respon berisi hasil permintaan data, pesan pemrosesan data, atau pesan kesalahan. Respon dari server selanjutnya akan diproses di klien seperti ditampilkan dalam bentuk tabel atau digunakan untuk pemrosesan lainnya. Alur pemrosesan data permintaan dari klien dapat dilihat pada Gambar 16.



Gambar 16. Alur pemrosesan data permintaan dari klien

4. Kesimpulan

Penerapan REST API memberikan kemudahan dalam pengembangan aplikasi berbasis *web service*. Dengan memanfaatkan prinsip-prinsip REST, termasuk penggunaan metode HTTP standar dan format data yang fleksibel seperti JSON, memungkinkan aplikasi dapat dengan mudah berinteraksi dengan aplikasi lain. Desain RESTful yang sederhana memungkinkan pengembang untuk membangun API yang mudah dipahami dan diimplementasikan. Hasil penelitian dan studi kasus dengan mengembangkan Sistem Seleksi Penelitian yang telah dilakukan menunjukkan bahwa penerapan REST API mempermudah dalam implementasi dengan berbagai aplikasi yang sudah ada sebelumnya. Dalam studi kasus tersebut, data proposal penelitian diperoleh dengan meminta (*request*) dari API Penelitian, sehingga untuk memperoleh data proposal, pengguna tidak perlu lagi mengisi data proposal di dalam Sistem Seleksi. Penggunaan API juga mempermudah dalam otentikasi *login* ke dalam aplikasi karena proses otentikasi dilakukan oleh API Akun dan API LDAP. Dengan demikian, REST dapat menjadi solusi ideal untuk integrasi sistem dan pengembangan aplikasi berbasis *web service*.

Salah satu tantangan dalam pengembangan aplikasi berbasis *web service* adalah masalah keamanan data yang dikirim dan diterima melalui jaringan, terutama penanganan akses yang tidak sah. Untuk itu, penelitian selanjutnya dapat memfokuskan pada keamanan data dengan mengimplementasikan protokol keamanan seperti OAuth, HTTPS, dan menggunakan token autentikasi (JWT).

Referensi

- [1] I. Ghani, W. M. N. Wan-kadir, and A. Mustafa, "Web Service Testing Techniques : A Systematic Literature Review," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 8, pp. 443–458, 2019.
- [2] A. Soni and V. Ranga, "API features individualizing of web services: REST and SOAP," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 9 Special Issue, pp. 664–671, 2019, doi: <https://doi.org/10.35940/ijitee.I1107.0789S19>
- [3] L. Li, W. Chou, W. Zhou, and M. Luo, "Design Patterns and Extensibility of REST API for Networking Applications," *IEEE Transactions on Network and Service Management*, vol. 13, no. 1, pp. 154–167, 2016, doi: <https://doi.org/10.1109/TNSM.2016.2516946>
- [4] H. Ed-douibi, J. L. Cánovas Izquierdo, and J. Cabot, "OpenAPItoUML: A Tool to Generate UML Models from OpenAPI Definitions," in *International Conference on Web Engineering*, T. Mikkonen, R. Klamma, and J. Hernández, Eds., Cham: Springer International Publishing, 2018, pp. 487–491. doi: https://doi.org/10.1007/978-3-319-91662-0_41
- [5] A. Ehsan, M. A. M. E. Abuhaliqa, C. Catal, and D. Mishra, "RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions," *Applied Sciences (Switzerland)*, vol. 12, no. 9, 2022, doi: <https://doi.org/10.3390/app12094369>
- [6] S. N. Yanti and E. Rihyanti, "Penerapan Rest API untuk Sistem Informasi Film Secara Daring," *Jurnal Informatika Universitas Pamulang*, vol. 6, no. 1, p. 195, 2021, doi: <https://doi.org/10.32493/informatika.v6i1.10033>
- [7] R. Choirudin and A. Adil, "Implementasi Rest Api Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 18, no. 2, pp. 284–293, 2019, doi: <https://doi.org/10.30812/matrik.v18i2.407>
- [8] Ardiansyah, D. Kurniawan, D. Sakheti, Bustomi, and B. Hermanto, "Web Service Sistem Informasi Terpadu (SIMIPA) Menggunakan REST API," *Jurnal Komputasi*, vol. 9, no. 2, pp. 11–26, 2021, doi: <https://doi.org/10.23960/komputasi.v9i2.2869>.
- [9] B. Adi Pranata, A. Hijriani, and A. Junaidi, "Perancangan Application Programming Interface (API) Berbasis Web Menggunakan Gaya Arsitektur Representational State Transfer (REST) Untuk Pengembangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit," *Jurnal Komputasi*, vol. 6, no. 1, pp. 33–42, 2018, doi: <https://doi.org/10.23960/komputasi.v6i1.1554>.
- [10] B. M. Adam, A. R. A. Besari, and M. M. Bachtiar, "Backend Server System Design Based on REST API for Cashless Payment System on Retail Community," in *2019 International Electronics Symposium (IES)*, 2019, pp. 208–213. doi: <https://doi.org/10.1109/ELECSYM.2019.8901668>.
- [11] I. W. Ridhoni, M. Sholeh, and U. J. Basuki, "PENGEMBANGAN APLIKASI ALQURAN ONLINE DENGAN MEMANFAATKAN REST API," *Device*, vol. 12, no. 2, pp. 51–59, 2022, doi: <https://doi.org/10.32699/device.v12i2.2861>.
- [12] Clinton Hatta Pradigi, Tintin Harlina, and Solehatin, "IMPLEMENTASI EXPRESS JS UNTUK MEMBANGUN REST API WEBSITE STIKOM PGRI BANYUWANGI," *Jikom: Jurnal Informatika dan Komputer*, vol. 9, no. 2, pp. 118–122, Jun. 2022, doi: <https://doi.org/10.55794/jikom.v9i2.68>.
- [13] I. Kurniawan and F. Rozi, "REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android," vol. 1, no. 4, pp. 127–132, 2020, doi: <https://doi.org/10.30630/jitsi.1.4.18>.
- [14] A. B. Prakoso, I. W. Santiyasa, and I. G. N. A. C. Putra, "Implementasi REST API pada Fitur User dari Aplikasi Peminjaman Ruangan," *Jurnal Pengabdian Informatika*, vol. 1, no. 1, pp. 73–82, 2022.
- [15] N. Dwivendi, D. Katiyar, and G. Goel, "A Comparative Study of Various Software Development Life Cycle (SDLC) Models," *International Journal of Research in Engineering, Science and Management*, vol. 5, no. 3, pp. 141–144, 2022. <https://journal.ijresm.com/index.php/ijresm/article/view/1881>
- [16] M. S. Supriya, S. N. Samreen, and N. Khatri-valmik, "A Comparative Study on Software Development Life Cycle Models," *International Research Journal of Engineering and Technology (IRJET)*, vol. 05, no. 02, pp. 696–700, 2018. <https://www.irjet.net/archives/V5/i2/IRJET-V5I2154.pdf>