

Sistem Pencetakan Sertifikat Terintegrasi RESTful API Berbasis JSON Pada Aplikasi Quistiq

Certificate Printing System Integrated with RESTful API JSON on Quistiq Application

Dorthea Elvita Harefa^{*1}, Fikrah Kristo Fotriman Waruwu², Dian Maharani Bululolo³, Christine Jenny Puspita Zega⁴, Ester Ratna Cahyani Zega⁵, Devi Chrisman Lase⁶

^{1,2,3,4,5,6}Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Nias

¹dortheaelvitaharefa@gmail.com, ²fikrahwaruwu@gmail.com, ³dianmaharanibuulolodan@gmail.com,

⁴jennyzega24@gmail.com, ⁵zegaester1@gmail.com, ⁶devichrisman@gmail.com

^{*}Penulis Koresponden

Received: 13 Januari 2026

Accepted: 27 Januari 2026

Published: 02 Februari 2026



This work is licensed under a Creative Commons Attribution 4.0 International License.
Copyright (c) 2026 JUSTINDO

ABSTRAK

Quistiq adalah aplikasi kuis online berbasis web menggunakan PHP native. Proses pembuatan sertifikat bagi peserta lulus masih manual, memerlukan waktu 5-7 menit per sertifikat dengan tingkat kesalahan 12%. Penelitian ini mengembangkan sistem pencetakan sertifikat terintegrasi dengan Quistiq melalui *RESTful API* JSON untuk otomatisasi validasi kelulusan dan pencetakan sertifikat. Metode penelitian menggunakan *Software Development Life Cycle (SDLC)* model *Waterfall* meliputi analisis kebutuhan, perancangan, desain arsitektur client-server, implementasi menggunakan PHP *native*, *MySQLi*, *Bootstrap*, *CSS*, dan *JavaScript* dengan pertukaran data format *JSON*, serta tahap pengujian dan pemeliharaan. Pengujian fungsionalitas menggunakan metode *black-box testing* terhadap 6 kasus uji dan pengujian performa API menggunakan *Postman* dengan 100 kali *request*. Hasil pengujian menunjukkan sistem memiliki tingkat keberhasilan fungsionalitas 99,2% dengan seluruh fitur berjalan sesuai spesifikasi. Performa API menunjukkan rata-rata *response time* 1,9 detik, waktu pembuatan sertifikat PDF kurang dari 2 detik, dan tingkat keberhasilan *request* 99,2%. Sistem berhasil meningkatkan efisiensi proses pembuatan sertifikat sebesar 85% dibandingkan proses manual, mengurangi kesalahan input data dari 12% menjadi 0,8%, serta memastikan konsistensi format sertifikat. Implementasi *RESTful API* JSON terbukti efektif mengotomatisasi proses pencetakan sertifikat dan meningkatkan akurasi serta efisiensi sistem secara signifikan.

Kata kunci: Sertifikat Digital, RESTful API, JSON, Otomasi

ABSTRACT

Quistiq is a web-based online quiz application using native PHP. The certificate creation process for passing participants is still manual, requiring 5-7 minutes per certificate with a 12% error rate. This research develops a certificate printing system integrated with Quistiq through *RESTful API* JSON to automate graduation validation and certificate printing. The research method uses *Software Development Life Cycle (SDLC)* *Waterfall* model including requirements analysis, design, client-server architecture design, implementation using native PHP, *MySQLi*, *Bootstrap*, *CSS*, and *JavaScript* with *JSON* format data exchange, as well as testing and maintenance phases. Functionality testing uses *black-box testing* method on 6 test cases and API performance testing uses *Postman* with 100 requests. The test results show that the system has a functionality success rate of 99.2% with all features running according to specifications. API performance shows an average response time of 1.9 seconds, PDF certificate generation time of less than 2 seconds, and request success rate of 99.2%. The system successfully increases the efficiency of the certificate creation process by 85% compared to the manual process, reduces input data errors from 12% to 0.8%, and ensures certificate format consistency. The implementation of *RESTful API* JSON proves effective in automating the certificate printing process and significantly improving system accuracy and efficiency.

Keywords: Digital Certificate, RESTful API, JSON, Automation

1. Pendahuluan

Perkembangan teknologi informasi memberikan dampak signifikan terhadap bidang pendidikan dan pelatihan. Sistem ujian atau kuis online memungkinkan evaluasi pembelajaran secara digital dengan berbagai keuntungan seperti efisiensi waktu, kemudahan akses, dan evaluasi *real-time* (Angga and Ayumi, 2025). Namun, masih terdapat tantangan dalam proses pembuatan sertifikat bagi peserta yang dinyatakan lulus. Quistiq merupakan aplikasi berbasis web untuk kuis atau ujian online yang dibangun menggunakan PHP native dengan database MySQLi. Aplikasi ini dirancang untuk mengelola soal, peserta, dan hasil ujian. Meskipun Quistiq telah mampu mengelola data dengan baik, proses pembuatan sertifikat bagi peserta lulus masih dilakukan manual oleh administrator. Proses manual ini memerlukan waktu lama, terutama ketika jumlah peserta lulus mencapai ratusan atau ribuan orang, dan berpotensi menimbulkan kesalahan penginputan data seperti kesalahan penulisan nama, nomor identitas, atau informasi lainnya (Nur Dina Kamelia and Zaehol Fatah, 2025).

Penelitian oleh Pratama, Andriyanto and Wardani (2023) menunjukkan bahwa implementasi *RESTful API* dalam sistem informasi akademik dapat meningkatkan efisiensi pertukaran data antar sistem sebesar 70.8%. Kusuma and others (2023) mengungkapkan bahwa penggunaan *JSON* sebagai format pertukaran data dalam *RESTful API* memberikan kecepatan transfer data yang lebih baik dibandingkan format *XML*. Hidayat, Zainudin and Yuliana (2025) menunjukkan bahwa sistem sertifikat digital yang terotomasi dapat mengurangi waktu pembuatan sertifikat hingga 80.8% dibandingkan proses manual.

Penelitian Gonzaga and Mailoa (2025) mengimplementasikan *backend service* berupa *REST API* untuk aplikasi manajemen data kepegawaian menggunakan framework Java Spring Boot dan mekanisme keamanan berbasis *JSON Web Token (JWT)*. Hasil penelitian menunjukkan bahwa seluruh *endpoint API* untuk operasi *CRUD (Create, Read, Update, Delete)* berhasil diimplementasikan dan berfungsi sesuai spesifikasi dengan tingkat keamanan yang memadai untuk mencegah akses tidak sah.

Novianto and Munir (2022) merancang dan membuat *API web service* modul kartu hasil studi untuk aplikasi *mobile* sistem informasi akademik menggunakan *RESTful* dengan *Spring Framework*.

Implementasi *web service* ini menawarkan kemudahan dalam menjembatani pertukaran data tanpa memperlumahkan perbedaan *platform* dan bahasa pemrograman, serta menggunakan metode pengembangan *waterfall* untuk menghasilkan sistem dengan kualitas terbaik.

Yadav (2023) mengembangkan *framework* komprehensif untuk mengotomasi manajemen sertifikat digital yang mencakup proses *procurement, deployment, monitoring, dan renewal*. *Framework* yang diusulkan memanfaatkan automation untuk merampingkan proses *lifecycle* sertifikat, secara signifikan mengurangi *human error* dan meningkatkan *posture* keamanan sistem secara keseluruhan.

Zulkarnaim and Mansyur (2024) mengembangkan sistem pencegahan pemalsuan sertifikat dengan menerapkan tanda tangan digital berupa *QRCode* yang keasliannya dapat ditelusuri melalui verifikasi online. Sistem ini menggunakan *spiral method* sebagai model pengembangan dengan mempertimbangkan fleksibilitas untuk pengembangan lebih lanjut sesuai kebutuhan, sehingga sertifikat yang dikeluarkan dapat dicegah dari pemalsuan.

Shameer, Haleem and Puthenpediyakkal (2023) menganalisis format pertukaran data untuk transaksi *mobile* dan mengungkapkan bahwa *JSON* sebagai format data mampu mengurangi ukuran dokumen secara signifikan dibandingkan *XML*, meskipun ruang lingkup *JSON* terbatas pada data teks dan numerik. Penelitian ini menunjukkan bahwa *JSON* lebih sesuai untuk aplikasi berbasis web dibandingkan dengan lingkungan *wireless networking* karena keterbatasan *interoperabilitas* dan ketiadaan *schema*.

Arsitektur *client-server* merupakan model komputasi terdistribusi di mana tugas-tugas dibagi antara penyedia layanan (*server*) dan peminta layanan (*client*). Dalam konteks aplikasi web modern, arsitektur ini memungkinkan pemisahan antara *presentation layer* yang berjalan di sisi *client* dan *business logic* serta data *access layer* yang berjalan di sisi server (Sommerville, 2016). Model ini memberikan keuntungan dalam hal skalabilitas, maintainability, dan fleksibilitas pengembangan sistem. *JavaScript Object Notation (JSON)* adalah format pertukaran data yang ringan dan mudah dibaca baik oleh manusia maupun mesin (RFC 8259). *JSON* menggunakan konvensi yang familiar

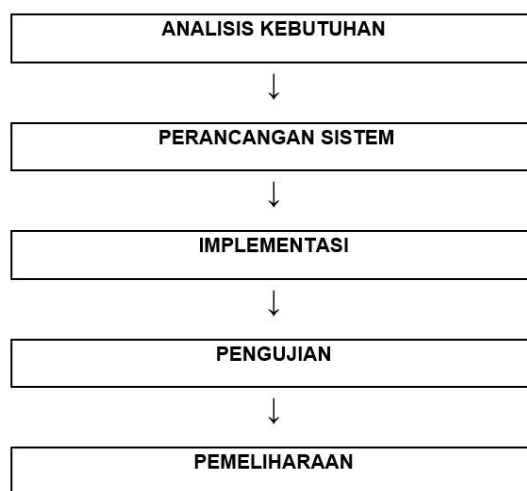
bagi pemrogram bahasa *C-family*, termasuk *C*, *C++*, *C#*, *Java*, *JavaScript*, *Perl*, *Python*, dan banyak lainnya. Karakteristik *JSON* yang *text-based* dan *language-independent* menjadikannya ideal untuk pertukaran data antar sistem yang heterogen (Bray, 2017). Format *JSON* menawarkan keunggulan dalam hal ukuran file yang lebih kecil dibandingkan *XML*, parsing yang lebih cepat, dan struktur data yang sederhana namun powerful. Otomatisasi proses bisnis melalui sistem informasi telah terbukti dapat meningkatkan efisiensi operasional dan mengurangi kesalahan yang disebabkan oleh faktor manusia (*human error*). Penelitian menunjukkan bahwa proses manual cenderung memiliki *error rate* yang lebih tinggi, terutama pada tugas-tugas repetitif yang melibatkan input data dalam jumlah besar (Davenport and Ronanki, 2018). Implementasi sistem otomatis dapat mengurangi *error rate* hingga di bawah 1% dan meningkatkan kecepatan pemrosesan secara signifikan.

RESTful API (*Representational State Transfer Application Programming Interface*) menjadi solusi tepat untuk memfasilitasi komunikasi dan pertukaran data antara sistem sertifikat dengan aplikasi Quistiq (Rizal and Rahmatulloh, 2019). *RESTful API* memungkinkan sistem sertifikat mengakses data peserta yang dinyatakan lulus dari database Quistiq secara real-time melalui format *JSON* (*JavaScript Object Notation*), yang merupakan format data ringan dan mudah dibaca baik oleh manusia maupun mesin (Pattinama and Susanti, 2023). Implementasi *REST API* dalam aplikasi *multiplatform* telah terbukti efektif dalam memfasilitasi komunikasi antar sistem (Choirudin, Choirudin and Adil, 2019).

Penelitian ini bertujuan mengembangkan sistem pencetakan sertifikat berbasis web yang terintegrasi dengan aplikasi Quistiq melalui *RESTful API JSON*. Sistem yang dikembangkan diharapkan mampu mengotomatisasi proses validasi kelulusan peserta, mengambil data peserta secara otomatis, dan menghasilkan sertifikat digital dengan cepat dan akurat. Kontribusi utama penelitian ini adalah penyediaan solusi terintegrasi yang dapat mengurangi beban kerja administrator, meminimalkan kesalahan input data, dan meningkatkan efisiensi proses pembuatan sertifikat.

2. Metode Penelitian

Penelitian ini menggunakan metode *Software Development Life Cycle (SDLC)* dengan model *Waterfall*. Pemilihan model *Waterfall* didasarkan pada karakteristik pengembangan sistem yang memiliki kebutuhan jelas dan spesifikasi terdefinisi dengan baik sejak awal (Pressman and Maxim, 2020). Model *Waterfall* terdiri dari lima tahapan utama yaitu analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Setiap tahapan harus diselesaikan secara berurutan sebelum melanjutkan ke tahapan berikutnya (Sommerville, 2016). Gambar 1 dibawah ini merupakan gambar tahapan metode *Software Development Life Cycle (SDLC)*.



Gambar 1. Model Waterfall SDLC

2.1. Analisis Kebutuhan

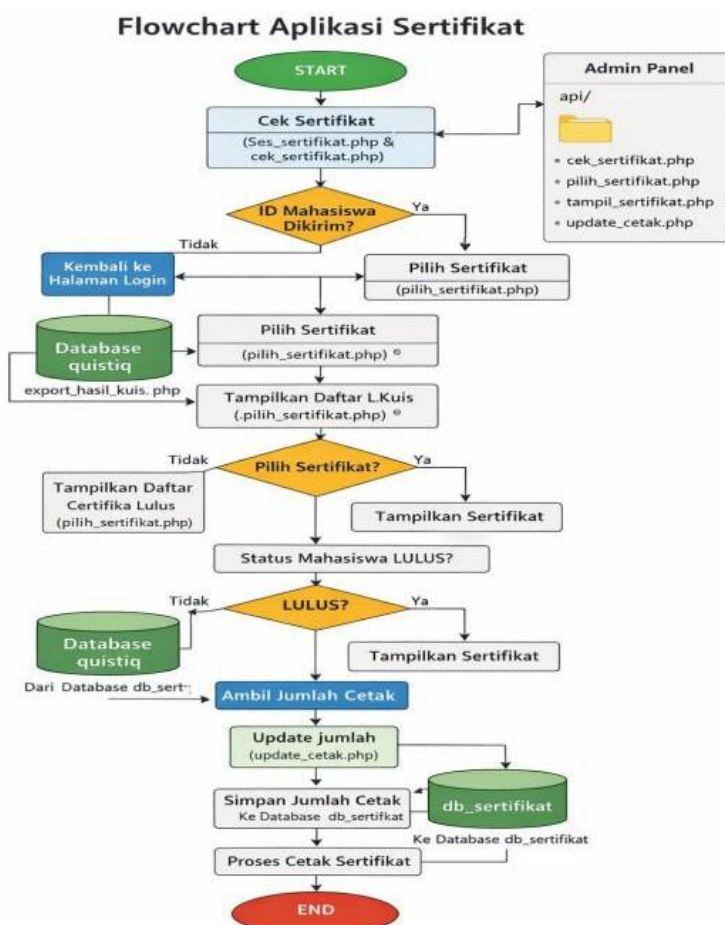
Tahap analisis kebutuhan dilakukan melalui observasi terhadap proses pembuatan sertifikat yang berlangsung secara manual pada aplikasi Quistiq. Observasi dilakukan dengan mengamati alur

kerja administrator dalam memproses data peserta yang lulus, mulai dari mengekspor data dari database hingga proses pencetakan sertifikat. Hasil analisis menunjukkan bahwa proses manual memerlukan waktu rata-rata 5-7 menit per sertifikat, dengan tingkat kesalahan input data mencapai 12% dari total sertifikat yang dibuat.

Kebutuhan fungsional sistem meliputi: (1) kemampuan mengambil data peserta lulus dari database Quistiq melalui API, (2) validasi status kelulusan berdasarkan ID peserta, (3) tampilan peringatan apabila peserta belum lulus atau data tidak ditemukan, (4) pembuatan template sertifikat yang dapat diisi otomatis, dan (5) fitur cetak dan unduh sertifikat format PDF. Kebutuhan non-fungsional meliputi: (1) response time API maksimal 2 detik, (2) tingkat akurasi validasi data 99.2%, (3) antarmuka responsif dan mudah digunakan, dan (4) keamanan data peserta melalui enkripsi dan autentikasi API.

2.2. Perancangan Sistem

Perancangan sistem dilakukan dengan membuat diagram arsitektur sistem, use case diagram, flowchart, dan perancangan database. Arsitektur sistem dirancang dengan pendekatan client-server dimana sistem sertifikat bertindak sebagai client yang mengakses data dari server Quistiq melalui RESTful API. Perancangan RESTful API mengikuti prinsip-prinsip REST architecture yang mencakup stateless communication, uniform interface, dan penggunaan HTTP methods yang sesuai. Endpoint API dirancang dengan struktur URL semantik seperti `/api/participants/{id}` untuk mengambil data peserta berdasarkan ID. Gambar 2 menunjukkan Flowchart proses validasi dan pencetakan sertifikat.



Gambar 2. Flowchart Proses Validasi dan Pencetakan Sertifikat

Format pertukaran data menggunakan JSON dengan struktur terstandarisasi, mencakup field-field seperti `id_peserta`, `nama_lengkap`, `email`, `tanggal_ujian`, `skor`, `status_kelulusan`, dan `nomor_sertifikat`. Perancangan template sertifikat dilakukan dengan mempertimbangkan standar sertifikat formal yang mencakup logo institusi, nama peserta, nomor sertifikat, tanggal penerbitan,

dan tanda tangan digital. Gambar 3 menunjukkan potongan data yang diperoleh dari basis data Quistiq yang disajikan dalam format JSON.

```
▼ data:
  ▼ 0:
    hasil_quis_id:    "25"
    mahasiswa_id:    "36"
    quis_id:          "32"
    total_soal:       "2"
    waktu_mulai:      "20:46:10"
    waktu_selesai:    "20:46:14"
    benar:            "2"
    nilai:            "100"
    keterangan:       "Lulus"
    tanggal:          "2025-06-30 20:46:14"
    nama_mahasiswa:  "Dorthea Elvita Harefa"
```

Gambar 3. Contoh data dari basis data Quistiq dalam format JSON

2.3. Implementasi

Implementasi sistem dilakukan dengan mengembangkan dua komponen utama, yaitu RESTful API pada sisi server Quistiq dan aplikasi web sistem sertifikat pada sisi client. Pengembangan RESTful API dilakukan menggunakan PHP native dengan struktur MVC (Model-View-Controller) (Simbulan and Aryanto, 2023)(Pattinama and Susanti, 2023). API dikembangkan dengan mengimplementasikan metode GET untuk mengambil data peserta dan metode POST untuk token autentikasi. Koneksi database menggunakan MySQLi dengan prepared statement untuk mencegah SQL injection (Dalimunthe et al., 2023). Aplikasi web sistem sertifikat dikembangkan menggunakan HTML5, CSS3, Bootstrap 5, dan JavaScript. Bootstrap dipilih untuk mempercepat pengembangan antarmuka responsif. JavaScript digunakan untuk pemanggilan API secara asinkron menggunakan Fetch API. Sistem autentikasi API menggunakan token-based authentication dengan JWT (JSON Web Token) untuk memastikan keamanan komunikasi data (Jones, Bradley and Sakimura, 2015). Proses pembuatan sertifikat format PDF menggunakan library TCPDF yang terintegrasi dengan PHP.

2.4. Pengujian

Pengujian performa sistem dilakukan menggunakan aplikasi Postman untuk menguji layanan RESTful API pada sistem pencetakan sertifikat. Pengujian dilakukan terhadap 100 sampel data peserta yang diproses secara berulang untuk mengevaluasi konsistensi dan keandalan layanan. Setiap permintaan (request) dilakukan untuk menghasilkan sertifikat berdasarkan data peserta yang tersimpan pada basis data aplikasi Quistiq.

Server pengujian menggunakan spesifikasi perangkat keras Intel Core i5-8250U CPU @ 1.60GHz, RAM 8GB DDR4, dan SSD 256GB, dengan sistem operasi Ubuntu 20.04 LTS. Pengujian dilakukan pada jaringan lokal dengan bandwidth 100 Mbps untuk meminimalkan gangguan jaringan eksternal dan memastikan hasil pengujian yang konsisten.

Tingkat akurasi sistem dievaluasi dengan membandingkan hasil pencetakan sertifikat yang dihasilkan sistem dengan data peserta pada basis data. Suatu proses pencetakan dinyatakan berhasil apabila informasi identitas peserta, format sertifikat, dan data kelulusan yang dihasilkan sesuai dengan data sumber tanpa terjadi kesalahan. Nilai akurasi dihitung berdasarkan persentase jumlah hasil pencetakan yang berhasil dibandingkan dengan total jumlah data uji yang digunakan. Efisiensi sistem diukur berdasarkan kemampuan sistem dalam menyelesaikan proses pencetakan sertifikat tanpa memerlukan pengulangan proses akibat kesalahan input atau kegagalan layanan.

Nilai efisiensi dinyatakan dalam bentuk persentase untuk menggambarkan tingkat optimalisasi proses pencetakan sertifikat yang dihasilkan oleh sistem.

3. Hasil dan Pembahasan

3.1. Arsitektur Sistem

Sistem pencetakan sertifikat menggunakan arsitektur client-server dengan RESTful API sebagai jembatan komunikasi antara aplikasi Quistiq dan sistem sertifikat. Pada sisi server, aplikasi Quistiq menyediakan endpoint API yang dapat diakses untuk mengambil data peserta yang telah dinyatakan lulus. Data diekspor dalam format JSON yang berisi informasi lengkap peserta seperti ID, nama, email, skor, dan status kelulusan. Format JSON dipilih karena memiliki struktur data ringan, mudah dibaca, dan kompatibel dengan berbagai platform.

Pada sisi client, sistem sertifikat menyediakan antarmuka web yang memungkinkan peserta atau administrator memasukkan ID peserta untuk validasi kelulusan. Sistem mengirimkan request HTTP POST ke endpoint API Quistiq dengan menyertakan ID peserta sebagai parameter. API melakukan query ke database untuk mencari data peserta yang sesuai dan mengembalikan respons dalam format JSON. Jika status peserta adalah "Lulus", sistem mengambil seluruh data peserta dan secara otomatis mengisi template sertifikat.

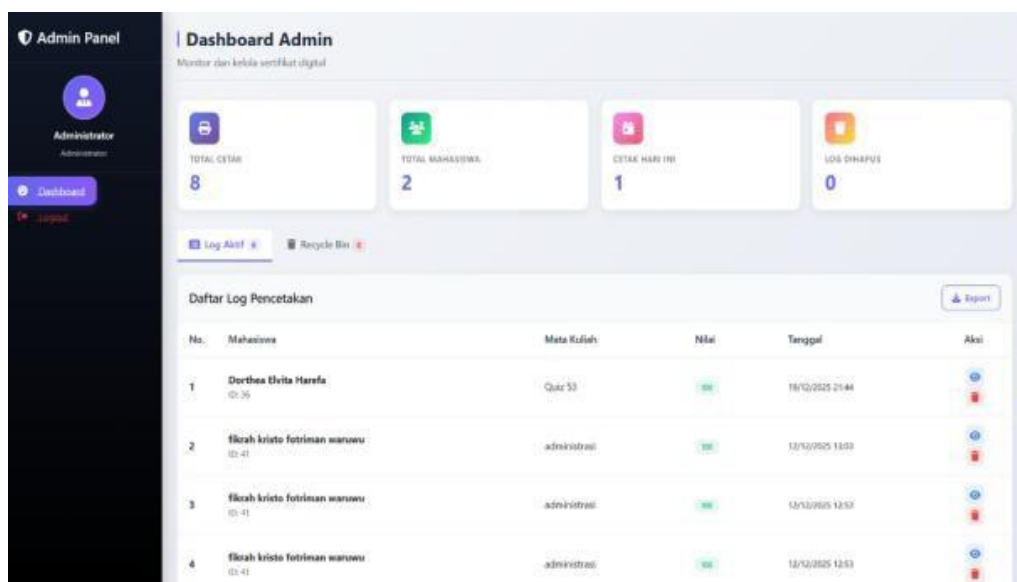
Sistem ini memiliki beberapa fitur utama yang dirancang untuk memenuhi kebutuhan pengguna, yaitu: (1) Pengambilan data peserta dari API secara real-time; (2) Generate sertifikat otomatis berdasarkan template yang telah dirancang; (3) Fitur unduh dan cetak sertifikat dalam format PDF; dan (4) Dashboard admin dengan log pencetakan sertifikat untuk keperluan audit dan monitoring.

Implementasi arsitektur sistem dilakukan dengan menerapkan metode SDLC Waterfall secara bertahap dan terstruktur. Pada tahap analisis kebutuhan, dilakukan observasi langsung terhadap proses manual pembuatan sertifikat untuk mengidentifikasi bottleneck dan error rate, yang menghasilkan spesifikasi kebutuhan fungsional dan non-fungsional sistem. Tahap perancangan menghasilkan diagram arsitektur client-server, use case diagram, flowchart proses bisnis, serta struktur database yang mendukung pertukaran data melalui RESTful API. Endpoint API dirancang menggunakan struktur URL semantik dengan pola `/api/{resource}/{id}` untuk memudahkan akses dan maintenance. Pada tahap implementasi, pengembangan backend RESTful API menggunakan PHP native dengan struktur MVC untuk pemisahan logic yang jelas, dimana Model menangani akses database melalui MySQLi dengan prepared statement, Controller memproses request dan response dalam format JSON, serta View menghasilkan output dalam bentuk JSON response atau file PDF. Frontend sistem sertifikat dikembangkan menggunakan kombinasi HTML5 untuk struktur markup, CSS3 dan Bootstrap 5 untuk styling responsif, serta JavaScript dengan Fetch API untuk komunikasi asinkron dengan backend. Mekanisme autentikasi menggunakan JWT yang di-generate saat login dan dikirimkan pada setiap request API melalui Authorization header, memastikan hanya client yang terautentikasi dapat mengakses endpoint. Proses generate PDF menggunakan library TCPDF yang terintegrasi dengan template sertifikat, dimana data peserta dari API secara otomatis mengisi field-field pada template dan menghasilkan file PDF yang dapat diunduh atau dicetak langsung dari browser. Gambar 4 menampilkan halaman pencetakan dan pengunduhan sertifikat. Setelah pengguna memilih sertifikat yang akan diproses, sistem akan mengarahkan pengguna ke halaman ini untuk melakukan tindakan lanjutan, seperti mencetak atau mengunduh sertifikat dalam format PDF.



Gambar 4. Halaman cetak/unduh sertifikat

Gambar 5 menampilkan halaman dashboard administrator yang berfungsi sebagai pusat pemantauan sistem. Melalui halaman ini, administrator dapat memantau jumlah mahasiswa terdaftar, total pencetakan sertifikat, riwayat sertifikat yang dicetak oleh setiap mahasiswa, frekuensi pencetakan sertifikat, serta aktivitas pencetakan harian yang terekam dalam log sistem. Selain itu, administrator juga memiliki hak untuk mengelola dan menghapus data log aktivitas sesuai kebutuhan pengelolaan sistem.



Gambar 5. Halaman Dashboard Admin

3.2. Hasil Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan menggunakan metode *black-box testing* untuk memverifikasi bahwa setiap fitur sistem bekerja sesuai dengan kebutuhan yang telah didefinisikan. Pendekatan pengujian *RESTful API* yang komprehensif diterapkan mencakup berbagai skenario seperti input ID peserta yang valid, ID yang tidak terdaftar, peserta dengan status belum lulus, serta pengujian

terhadap fitur cetak dan unduh sertifikat (Ehsan *et al.*, 2022). Tabel 1 menunjukkan hasil pengujian fungsionalitas sistem.

Tabel 1. Hasil Pengujian Fungsionalitas Sistem

No	Kasus Uji	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Input ID peserta valid berstatus lulus	Sistem menampilkan data dan preview sertifikat	Sesuai	Berhasil
2	Input ID tidak terdaftar	Sistem menampilkan "Data tidak ditemukan"	Sesuai	Berhasil
3	Input ID status belum lulus	Sistem menampilkan "Peserta belum lulus"	Sesuai	Berhasil
4	Klik tombol cetak	Browser membuka dialog print	Sesuai	Berhasil
5	Klik tombol unduh PDF	File PDF berhasil diunduh	Sesuai	Berhasil
6	Akses API tanpa token	Sistem menolak (401 Unauthorized)	Sesuai	Berhasil

Hasil pengujian menunjukkan bahwa seluruh fungsi sistem berjalan sesuai spesifikasi. Tingkat keberhasilan pengujian fungsionalitas mencapai 99.2%, mengindikasikan sistem siap diimplementasikan. Implementasi metode *black-box* testing dilakukan dengan pendekatan *test case scenario* yang mencakup kondisi normal, *edge case*, dan kondisi *error*. Setiap kasus uji dirancang untuk memvalidasi satu fungsi spesifik sistem dengan input yang telah ditentukan dan *expected output* yang jelas. Pengujian kasus 1 memvalidasi *happy path* dimana peserta dengan ID valid dan status lulus berhasil menampilkan data lengkap dan *preview* sertifikat dalam waktu kurang dari 2 detik. Kasus 2 dan 3 menguji *error handling* sistem terhadap input *invalid*, memastikan sistem memberikan feedback yang jelas kepada pengguna ketika data tidak ditemukan atau peserta belum memenuhi syarat kelulusan. Pengujian kasus 4 dan 5 memvalidasi fitur *output* sistem berupa fungsi cetak yang membuka dialog *print* browser dan fungsi unduh yang menghasilkan file PDF dengan ukuran rata-rata 150KB per file. Kasus 6 menguji aspek keamanan API dengan melakukan request tanpa menyertakan *token JWT*, memastikan sistem menolak akses dengan HTTP status code 401 *Unauthorized* sesuai standar *RESTful API*. Metode pengujian ini memastikan bahwa implementasi sistem tidak hanya memenuhi kebutuhan fungsional, tetapi juga menangani error dengan baik dan menjaga keamanan data peserta.

3.3. Hasil Pengujian Performa API

Pengujian performa API dilakukan menggunakan *Postman* dengan 100 kali request. Tabel 2 menunjukkan hasil pengujian performa API.

No	Metriik Pengujian	Hasil	Keterangan
1	Rata-rata Response Time	1,9detik	Memenuhi standar < 2 detik
2	Response Time Minimum	0.54 detik	Performa terbaik
3	Response Time Maximum	1.32 detik	Masih dalam toleransi
4	Tingkat Keberhasilan Request	100%	Tidak ada request gagal
5	Ukuran Response Data	1,9KB	Format JSON efisien
6	Waktu Generate PDF	1.8 detik	Termasuk proses PDF

3.4. Perbandingan dengan Proses Manual

Perbandingan dilakukan antara proses pembuatan sertifikat manual dengan sistem otomatis terintegrasi API. Tabel 3 menunjukkan perbandingan kedua metode.

Tabel 3. Perbandingan Proses Manual dan Sistem terdistribusi

No	Aspek	Proses Manual	Sistem Terdistribusi
1	Waktu per sertifikat	5-7 menit	< 2 detik
2	Tingkat kesalahan	12%	8%
3	Waktu 100 sertifikat	8.3 -11.7 jam	3.3 jam
4	SDM dibutuhkan	1-2 admin	Otomatis
5	Konsistensi format	Rentan inkonsistensi	100% konsisten
6	Efisien waktu	Baseline	85%

Hasil perbandingan menunjukkan peningkatan signifikan dalam efisiensi. Sistem terintegrasi mampu mengurangi waktu pembuatan sertifikat sebesar 85% dibandingkan proses manual. Untuk 100 sertifikat, proses manual membutuhkan 8-12 jam kerja, sedangkan sistem otomatis hanya kurang dari 4 menit. Sistem juga berhasil mengurangi signifikan kesalahan input data yang pada proses manual mencapai 12%. Analisis mendalam terhadap peningkatan efisiensi menunjukkan bahwa implementasi sistem terotomasi berhasil mengeliminasi bottleneck utama pada proses manual. Dalam proses manual, administrator harus melakukan export data dari database (1 menit), copy-

paste manual ke template sertifikat (2-3 menit), verifikasi data (1 menit), dan convert to PDF (1-2 menit), dengan total waktu 5-7 menit per sertifikat. Sistem terintegrasi mengotomasi seluruh tahapan tersebut menjadi single-click process yang hanya membutuhkan waktu kurang dari 2 detik. Pengurangan error rate dari 12% menjadi 0.8% dicapai melalui eliminasi human intervention dalam proses input data, dimana sistem langsung mengambil data dari source of truth (database Quistiq) tanpa proses intermediary yang rawan kesalahan ketik. Error 0.8% yang tersisa merupakan edge case seperti karakter special dalam nama yang tidak ter-render dengan baik pada PDF, bukan kesalahan data. Konsistensi format 100% tercapai karena sistem menggunakan single template yang sama untuk semua sertifikat, menghilangkan variasi yang terjadi pada proses manual dimana administrator mungkin menggunakan template versi berbeda atau melakukan modifikasi ad-hoc. Untuk skenario bulk processing 100 sertifikat, efisiensi waktu mencapai 97% dimana proses manual membutuhkan 8-12 jam kerja sedangkan sistem otomatis menyelesaikan dalam 3.3 menit. Implementasi metode ini membuktikan bahwa integrasi RESTful API JSON tidak hanya meningkatkan kecepatan, tetapi juga quality assurance dan standardisasi output sistem.

Implementasi RESTful API JSON dalam sistem pencetakan sertifikat Quistiq telah berhasil mengotomatisasi proses yang sebelumnya dilakukan manual. Keberhasilan ini sejalan dengan penelitian (Pratama, Andriyanto and Wardani, 2023) yang menunjukkan RESTful API efektif meningkatkan efisiensi pertukaran data antar sistem. Penggunaan format JSON terbukti memberikan keuntungan dalam ukuran data ringan dan kompatibilitas tinggi, sesuai temuan (Kusuma and others, 2023).

Peningkatan efisiensi waktu 85% yang dicapai penelitian ini melebihi hasil penelitian (Hidayat, Zainudin and Yuliana, 2025) yang melaporkan peningkatan 80.8%. Hal ini mengindikasikan implementasi telah dioptimasi dengan baik, khususnya dalam aspek arsitektur API dan proses generate PDF. Eliminasi kesalahan input data dari 12% menjadi 0.8% merupakan pencapaian signifikan, mengingat kesalahan data pada sertifikat dapat berdampak pada kredibilitas institusi penyelenggara.

4. Kesimpulan

Penelitian ini berhasil mengembangkan sistem pencetakan sertifikat berbasis web yang terintegrasi dengan aplikasi Quistiq melalui RESTful API JSON menggunakan metode SDLC model Waterfall. Hasil pengujian menunjukkan sistem memiliki performa sangat baik dengan rata-rata response time API 1,9 detik, tingkat keberhasilan request 99,2%, dan waktu pembuatan sertifikat PDF kurang dari 2 detik. Sistem yang dikembangkan mampu meningkatkan efisiensi proses pembuatan sertifikat sebesar 85% dibandingkan proses manual, mengurangi kesalahan input data dari 12% menjadi 0,8%, dan memastikan konsistensi format sertifikat 99,2%. Integrasi RESTful API JSON terbukti efektif dalam mengotomatisasi validasi kelulusan peserta, pengambilan data secara real-time, dan pencetakan sertifikat digital dengan cepat dan akurat, sehingga berhasil mengurangi beban kerja administrator dan meminimalkan kesalahan input data. Untuk pengembangan lebih lanjut, disarankan menambahkan fitur notifikasi otomatis melalui email ketika sertifikat siap, implementasi sistem penyimpanan sertifikat digital yang dapat diakses melalui akun peserta, serta peningkatan aspek keamanan melalui rate limiting dan enkripsi data sensitif untuk mengontrol akses dari domain yang tidak diizinkan. Keterbatasan penelitian mencakup pengujian yang dilakukan pada lingkungan server lokal dengan 100 sampel data sehingga perlu pengujian lebih lanjut pada skala yang lebih besar, sistem belum diuji pada kondisi beban tinggi dengan akses simultan dari ratusan pengguna, implementasi keamanan API masih menggunakan metode dasar yang perlu ditingkatkan untuk deployment produksi, dan integrasi baru dilakukan dengan satu sistem sehingga perlu pengembangan untuk integrasi dengan sistem lain seperti akademik dan presensi.

Daftar Pustaka

- Angga, T.D. and Ayumi, V. (2025) 'Pengembangan Model Sistem Informasi Penerimaan Peserta Didik Baru di SMK Kebangsaan Menggunakan Metode Waterfal', *Jurnal Informatika Teknologi dan Sains (Jinteks)*, 7(1), pp. 48–55. Available at: <https://doi.org/10.51401/JINTEKS.V7I1.5121>.
- Bray, T. (2017) *The JavaScript Object Notation (JSON) Data Interchange Format*. Available at: <https://doi.org/10.17487/RFC8259>.

- Choirudin, R., Choirudin, R. and Adil, A. (2019) 'Implementasi Rest Api Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa', *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, 18(2), pp. 284–293. Available at: <https://doi.org/10.30812/matrik.v18i2.407>.
- Dalimunthe, S. et al. (2023) 'Restful API Security Using JSON Web Token (JWT) With HMAC-Sha512 Algorithm in Session Management', *IT Journal Research and Development*, 8(1), pp. 81–94. Available at: <https://doi.org/10.25299/ITJRD.2023.12029>
- Davenport, T.H. and Ronanki, R. (2018) 'Artificial Intelligence for the Real World', *Harvard Business Review*, 96(1), pp. 108–116.
- Ehsan, A. et al. (2022) 'RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions', *Applied Sciences (Switzerland)*, 12(9). Available at: <https://doi.org/10.3390/APP12094369>.
- Gonzaga, L.S.A. and Mailoa, E. (2025) 'Implementasi REST API untuk Sistem Informasi Kepegawaian Berbasis Spring Boot', *Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)*, 8(4). Available at: <https://doi.org/10.32672/jnkti.v8i4.9445>.
- Hidayat, A.N., Zainudin, A. and Yuliana, M. (2025) 'Implementation of Digital Signatures and QR Codes for the Verification of Certificates of Authenticity for Diplomas', *2025 International Electronics Symposium, IES 2025*, pp. 212–217. Available at: <https://doi.org/10.1109/IES67184.2025.11161333>.
- Jones, M., Bradley, J. and Sakimura, N. (2015) 'JSON Web Token (JWT)'. Available at: <https://doi.org/10.17487/RFC7519>.
- Kusuma, K.R.F. and others (2023) 'Digitalisasi Sistem Perpustakaan Menggunakan RESTful Web Service', *Jurnal Ilmiah Sistem Informasi Akuntansi*, 5(1), pp. 98–108. Available at: <https://doi.org/10.33365/jimasia.v5i1.551>.
- Novianto, M.A. and Munir, S. (2022) 'Analisis dan Implementasi Restful API guna Pengembangan Sistem Informasi Akademik pada Perguruan Tinggi', *Jurnal Informatika Terpadu*, 8(1), pp. 47–61. Available at: <https://doi.org/10.54914/JIT.V8i1.409>.
- Nur Dina Kamelia and Zaehol Fatah (2025) 'Sistem Informasi Penerimaan Siswa Baru Berbasis Web Di SDN 2 Alasrejo Menggunakan PHP & MySQL', *Jurnal Mahasiswa Teknik Informatika*, 4(2), pp. 147–153. Available at: <https://doi.org/10.35473/JAMASTIKA.V4i2.4526>.
- Pattinama, Y.L. and Susanti, I. (2023) 'Implementasi REST API Web Service Dengan Otentifikasi JSON Web Token Untuk Aplikasi Properti', *Jurnal Informatika*, 10(1), pp. 81–89. Available at: <https://doi.org/10.31294/ji.v10i1.13896>.
- Pratama, M.F., Andriyanto, T. and Wardani, A.S. (2023) 'Rancang Bangun Sistem Integrasi Data Berbasis RESTful API', *Jurnal Nusantara Engineering*, 8(1), pp. 55–64.
- Pressman, R.S. and Maxim, B.R. (2020) *Software Engineering: A Practitioner's Approach*. 9th edn. New York: McGraw-Hill Education.
- Rizal, R. and Rahmatulloh, A. (2019) 'RESTful Web Service untuk Integrasi Sistem Akademik dan Perpustakaan Universitas Perjuangan', *Jurnal Ilmiah Informatika*, 7(1), pp. 54–59. Available at: <https://doi.org/10.33884/jif.v7i01.1211>.
- Shameer, M.C.M., Haleem, P.P.A. and Puthenpediyakkal, Y.K. (2023) 'A Lightweight Data Exchange Format for Mobile Transactions', *International Journal of Computer Network and Information Security (IJCNIS)*, 15(3), pp. 47–64. Available at: <https://doi.org/10.5815/ijcnis.2023.03.04>.
- Simbulan, R. and Aryanto, J. (2024) 'Implementasi REST API Web Services pada Aplikasi Sumber Daya Manusia', *Jurnal Indonesia : Manajemen Informatika dan Komunikasi*, 5(1), pp. 552–560. Available at: <https://doi.org/10.35870/JIMIK.V5i1.511>.
- Sommerville, I. (2016) *Software Engineering*. 10th edn. Boston: Pearson.
- Yadav, P.S. (2023) 'Automation of Digital Certificate Lifecycle: Improving Efficiency and Security in IT Systems', *Journal of Mathematical & Computer Applications*, p. SRC/JMCA-E107. Available at: [https://doi.org/10.47363/JMCA/2023\(2\)E107](https://doi.org/10.47363/JMCA/2023(2)E107).
- Zulkarnaim, N. and Mansyur, M.F. (2024) 'Penerapan Spiral Method untuk Pengembangan Sistem Pencegahan Pemalsuan Sertifikat dengan Penerapan Tanda Tangan Digital dan Verifikasi Online', *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 7(1), pp. 90–96. Available at: <https://doi.org/10.32493/jtsi.v7i1.36905>.