

IMPLEMENTASI PROTOKOL HL7 ORU^R01 PADA SISTEM INFORMASI LABORATORIUM UNTUK INTEGRASI ANALYZER DAN SISTEM INFORMASI RUMAH SAKIT

Moh. Harysakti Rahanyamtel, Susanto
Teknik Informatika, Universitas Semarang
Jalan Soekarno-Hatta Tlogosari Semarang, Indonesia
G231220045@student.usm.ac.id

ABSTRAK

Laboratory Information System (LIS) memiliki peranan penting dalam mempercepat pengelolaan hasil pemeriksaan medis, namun integrasi antara analyzer dan *Hospital Information System (HIS)* masih sering dilakukan secara manual sehingga berpotensi menimbulkan keterlambatan dan kesalahan pencatatan. Penelitian ini bertujuan mengimplementasikan protokol *Health Level Seven (HL7) ORU^R01* sebagai standar pertukaran data laboratorium melalui pengembangan sistem berbasis metode *Prototyping*. Sistem terdiri atas *gateway* berbasis Python yang menerima pesan HL7 dari analyzer dan mengonversinya menjadi JSON, serta server LIS berbasis CodeIgniter 4 yang menyediakan verifikasi data pasien dan publikasi hasil ke HIS melalui *Application Programming Interface (API)*. Pengujian dilakukan melalui uji fungsional, uji integrasi, dan uji kinerja, termasuk pengujian *endpoint API* menggunakan *Postman* untuk memastikan bahwa HIS dapat mengakses data yang telah diverifikasi. Hasil pengujian menunjukkan bahwa seluruh pesan HL7 dapat diproses tanpa error, konversi JSON berjalan valid, verifikasi data dapat dilakukan dengan benar, dan API merespons sukses pada setiap percobaan. *Endpoint API* hanya menampilkan data hasil verifikasi sehingga HIS menerima informasi yang telah divalidasi secara klinis. Rata-rata waktu transmisi kurang dari satu detik, dan modul *Sync Audit* mencatat tingkat keberhasilan sinkronisasi sebesar 100 persen. Implementasi ini terbukti meningkatkan efisiensi, ketepatan, dan kecepatan pengelolaan data laboratorium dalam lingkungan rumah sakit.

Kata kunci : *Laboratory Information System, Hospital Information System, HL7 ORU^R01, Prototyping, Integrasi Data*

1. PENDAHULUAN

Perkembangan teknologi informasi dalam bidang kesehatan telah mendorong kebutuhan akan sistem informasi yang mampu mengintegrasikan data antar unit pelayanan medis secara cepat, akurat, dan terstandar. Salah satu komponen penting dalam sistem tersebut adalah *Laboratory Information System (LIS)*, yang berfungsi mengelola data hasil pemeriksaan laboratorium untuk mendukung proses diagnosis dan pengambilan keputusan klinis [1]. Namun, dalam banyak kasus, LIS belum sepenuhnya terhubung dengan *Hospital Information System (HIS)*, sehingga masih terjadi redundansi data dan kesalahan input akibat proses entri manual. Permasalahan ini dapat mengurangi efisiensi pelayanan dan menurunkan akurasi data medis.

Untuk mengatasi tantangan interoperabilitas tersebut, diterapkan standar komunikasi kesehatan seperti *Health Level Seven (HL7)*, yang menyediakan format pesan terstruktur untuk pertukaran data antar sistem kesehatan. Salah satu tipe pesan yang digunakan secara luas untuk pertukaran hasil laboratorium adalah HL7 ORU^R01 (*Observation Result Unsolicited*), yang memungkinkan pengiriman data hasil pemeriksaan dari LIS ke HIS secara otomatis [2]. Implementasi protokol ini telah terbukti meningkatkan efisiensi pertukaran data, mengurangi beban kerja administratif, serta mempercepat proses diagnosis klinis di berbagai rumah sakit [3].

Meskipun demikian, penerapan HL7 tidak terlepas dari kendala teknis, seperti kompleksitas struktur pesan, perbedaan versi antar sistem, serta kebutuhan akan mekanisme *gateway* yang mampu menerjemahkan pesan HL7 ke dalam format yang dapat diterima sistem rumah sakit modern [4]. Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem integrasi berbasis HL7 ORU^R01 antara LIS dan HIS menggunakan pendekatan *middleware gateway*. Rencana penyelesaian masalah dilakukan dengan mengembangkan sistem yang mampu menerima data dari alat analyzer, mengonversi pesan HL7 ke dalam format JSON, serta mempublikasikan data hasil verifikasi ke HIS melalui *Application Programming Interface (API)*.

Penelitian ini diharapkan dapat memberikan solusi efektif terhadap permasalahan interoperabilitas data laboratorium di rumah sakit, sekaligus mendukung implementasi *smart hospital* melalui integrasi data yang lebih efisien, aman, dan sesuai dengan standar internasional. Selain itu, hasil penelitian ini juga berpotensi menjadi acuan dalam pengembangan sistem informasi kesehatan yang menerapkan protokol HL7 di Indonesia secara lebih luas [5].

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian tentang penggabungan *Laboratory Information System* (LIS) dengan *Sistem Hospital Information System* (HIS) bertujuan untuk memudahkan pembagian data medis. Misalnya, *Schmidlehner* [4] mengkaji cara kerja integrasi berbasis standar HL7 ORU^R01. Integrasi ini memungkinkan sistem informasi laboratorium dan rumah sakit untuk berbagi data secara *real-time*. Temuan penelitian menunjukkan bahwa penerapan HL7 dapat mengurangi tingkat kesalahan dalam transmisi data dan mempercepat penyebaran hasil tes, sehingga meningkatkan akurasi dokumentasi medis di rumah sakit.

Selain itu, penelitian *Mtonga* [6] menghasilkan aplikasi untuk mengelola informasi laboratorium di rumah sakit dengan sumber daya terbatas. Aplikasi ini dapat dihubungkan ke sistem informasi laboratorium dan rumah sakit menggunakan standar HL7 untuk mempercepat pemrosesan data hasil pemeriksaan.

Penelitian Lain Oleh *AlQudah* [2] melakukan studi lain yang mengkaji pemanfaatan standar HL7 dalam integrasi data medis untuk memfasilitasi pengelolaan data pasien di rumah sakit. Metode ini memungkinkan pertukaran informasi secara otomatis, sehingga memudahkan perolehan hasil lab dan mengurangi risiko kesalahan data.

Berbagai penelitian menunjukkan bahwa meskipun sistem informasi kesehatan terus maju, pendekatan yang lebih efisien dan terstandarisasi masih diperlukan untuk mencapai interoperabilitas penuh antara *Laboratory Information System* (LIS) dan *Hospital Information System* (HIS) di berbagai rumah sakit.

2.2. Laboratory Information System (LIS)

Laboratory Information System (LIS) adalah sistem komputerisasi yang membantu menjalankan laboratorium dengan mencatat hal-hal seperti registrasi pasien, manajemen sampel, pengujian, dan pelaporan hasil. Sistem ini mempercepat dan meningkatkan akurasi pemrosesan data, serta mengurangi kesalahan manual. Standar komunikasi HL7 memungkinkan LIS modern bekerja dengan sistem rumah sakit [7].

2.3. Hospital Information System (HIS)

Hospital Information System (HIS) adalah perangkat lunak dan basis data terintegrasi yang mencakup pengelolaan data pasien, rekam medis elektronik, penjadwalan dokter, pengadaan obat dan alat kesehatan, administrasi, keuangan, serta sumber daya manusia [8].

2.4. Standar Komunikasi HL7 dan Pesan ORU^R01

Health Level Seven (HL7) adalah standar internasional yang digunakan untuk pertukaran data klinis dan administratif antar sistem informasi

kesehatan. Salah satu pesan penting dalam HL7 adalah ORU^R01, yang digunakan untuk mengirimkan hasil observasi klinis, seperti hasil laboratorium, dari satu sistem ke sistem lain secara otomatis dan terstruktur [9].

2.5. Python

Python adalah bahasa pemrograman serbaguna yang mudah dipelajari, mendukung banyak paradigma, dan sangat populer untuk berbagai aplikasi modern, mulai dari web hingga kecerdasan buatan [10].

2.6. Framework CodeIgniter 4

CodeIgniter 4 adalah sebuah framework open source berbasis PHP yang digunakan untuk membangun aplikasi web secara cepat, efisien, dan terstruktur. *CodeIgniter 4* mengadopsi pola arsitektur *Model-View-Controller* (MVC), yang memisahkan logika aplikasi, tampilan, dan pengelolaan data, sehingga memudahkan pengembangan, pemeliharaan, dan pengujian aplikasi web [11]. Framework ini mendukung pembuatan *RESTful API* yang digunakan untuk integrasi data antara LIS dan HIS secara *real-time*. *CodeIgniter 4* dipilih karena memiliki kinerja cepat, keamanan tinggi, serta kemudahan pengembangan aplikasi berbasis web.

2.7. Postman

Postman merupakan sebuah *API client* yang digunakan untuk membangun, menjalankan, dan mengelola pengujian terhadap *RESTful API*. Alat ini menyediakan antarmuka interaktif yang memungkinkan pengembang mengirim *HTTP request*, meninjau *response*, menyusun koleksi pengujian, serta memvalidasi perilaku layanan secara sistematis. Dalam berbagai studi terkait pengujian *API*, Postman dikategorikan sebagai perangkat uji manual yang berfungsi sebagai sarana verifikasi awal maupun sebagai dasar bagi penyusunan pengujian otomatis. Perannya sebagai *API client* memberikan kemudahan dalam menilai fungsi, struktur data, serta konsistensi respons dari suatu *RESTful API*, sehingga menjadikannya salah satu alat yang paling banyak digunakan dalam proses evaluasi kualitas layanan berbasis *web service* [12].

2.8. Metode Pengembangan Sistem Prototyping

Metode *Prototyping* digunakan untuk menghasilkan sistem yang sesuai dengan kebutuhan pengguna melalui tahapan iteratif, yaitu: identifikasi kebutuhan, pembuatan *prototype*, evaluasi pengguna, dan perbaikan sistem [13].

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan *prototyping*, yang merupakan bagian dari metode rekayasa perangkat lunak iteratif. Pendekatan ini dipilih karena memungkinkan pengembangan sistem secara bertahap dan memungkinkan umpan balik

langsung dari pengguna untuk penyempurnaan sistem integrasi antar aplikasi medis.

Model prototyping dipilih karena pendekatan ini memungkinkan proses pengembangan sistem dilakukan secara iteratif dan responsif terhadap kebutuhan pengguna yang dinamis. Dalam konteks integrasi sistem informasi kesehatan, kebutuhan sering kali bersifat kompleks dan belum sepenuhnya terdefinisi sejak awal. Dengan menggunakan model prototyping, pengembang dapat membuat versi awal sistem (*working prototype*) yang kemudian diuji, dievaluasi, dan disesuaikan secara bertahap berdasarkan masukan dari pengguna akhir, seperti analis laboratorium, tim IT, dan staf medis. Hal ini menjadikan prototyping efektif untuk memastikan bahwa sistem integrasi REST API dan konversi pesan HL7 v2.x ke format JSON benar-benar sesuai dengan kebutuhan fungsional dan teknis di lingkungan rumah sakit.

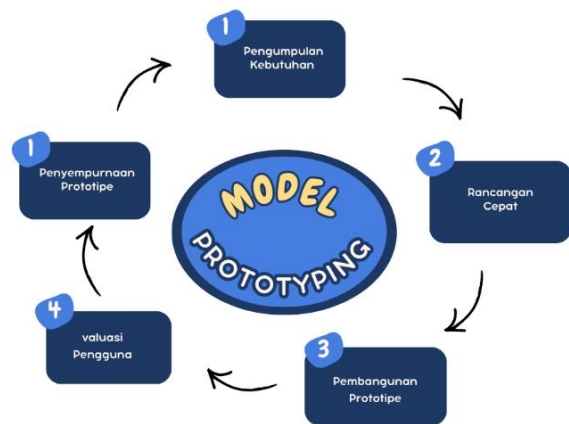
3.1. Jenis Penelitian

Penelitian ini merupakan penelitian terapan (*applied research*) yang berfokus pada pengembangan solusi teknologi untuk meningkatkan interoperabilitas sistem informasi kesehatan. Pendekatan terapan dipilih karena tujuan utama penelitian adalah menghasilkan prototipe sistem integrasi *Laboratory Information System* (LIS) dan *Hospital Information System* (HIS) yang dapat diimplementasikan secara langsung dalam lingkungan operasional. Selain itu, penelitian ini menggunakan pendekatan deskriptif untuk menjelaskan alur integrasi data, struktur pesan HL7 v2.x, serta proses konversi data menjadi format JSON yang dapat ditransmisikan melalui REST API.

3.2. Model Pengembangan Sistem

Model *prototyping* digunakan sebagai kerangka pengembangan karena memungkinkan perbaikan berkelanjutan melalui siklus iteratif *build evaluate refine*. Dalam model ini, prototipe awal dibangun untuk menggambarkan fungsi utama sistem, kemudian diuji oleh pengguna terkait untuk memperoleh umpan balik mengenai kebutuhan, kesesuaian proses bisnis, serta keandalan alur integrasi data. Hasil evaluasi tersebut digunakan untuk memperbaiki prototipe hingga mencapai bentuk akhir yang memenuhi persyaratan teknis dan fungsional. Pendekatan ini sangat relevan bagi pengembangan sistem integrasi rekam medis dan laboratorium, terutama ketika kebutuhan pengguna bersifat dinamis dan kompleks.

Gambar 1 menunjukkan tahapan model *prototyping* yang bersifat iteratif, dimulai dari pengumpulan kebutuhan, perancangan cepat, pembangunan prototipe, evaluasi pengguna, hingga penyempurnaan prototipe. Siklus ini digunakan untuk memastikan bahwa sistem integrasi HL7 REST API dapat memenuhi kebutuhan pengguna secara bertahap



Gambar 1. Model pengembangan sistem dengan metode prototyping

3.3. Tahapan Penelitian

Tahapan penelitian disusun mengikuti alur prototyping, meliputi:

- Pengumpulan kebutuhan dilakukan melalui studi literatur, analisis standar HL7 v2.x, serta identifikasi kebutuhan fungsional pada proses pertukaran data LIS-HIS.
- Perancangan prototipe awal membuat rancangan arsitektur integrasi, struktur *endpoint REST API*, desain konversi pesan HL7, serta alur komunikasi data.
- Pengembangan prototipe mengimplementasikan komponen inti seperti parser HL7, modul konversi JSON, serta layanan API untuk pengiriman dan penerimaan data.
- Evaluasi prototipe oleh pengguna melibatkan analis laboratorium, tenaga IT, dan staf medis untuk menilai fungsionalitas, keakuratan data, serta kemudahan integrasi.
- Penyempurnaan prototipe memperbaiki sistem berdasarkan umpan balik hingga diperoleh prototipe final yang stabil dan layak diuji pada lingkungan operasional.

3.4. Teknik Pengumpulan Data

Data diperoleh melalui tiga teknik utama:

- Studi literatur, meliputi jurnal ilmiah terkait interoperabilitas kesehatan, standar HL7, serta arsitektur integrasi berbasis REST API.
- Observasi proses kerja, yaitu pengamatan alur pengelolaan hasil laboratorium dan distribusi data pada sistem HIS.
- Wawancara informal, dilakukan dengan analis laboratorium dan tim IT untuk memahami kebutuhan operasional serta kendala pada sistem yang berjalan.

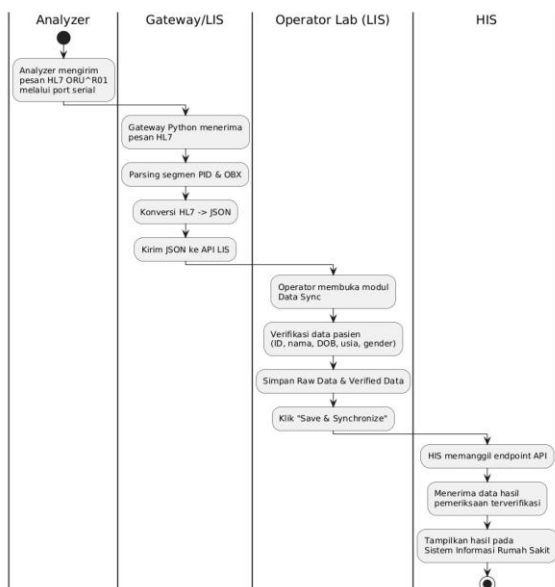
3.5. Teknik Analisis Data

Analisis data dilakukan secara kualitatif dengan menelaah kesesuaian antara kebutuhan pengguna, standar komunikasi medis, dan hasil uji prototipe. Data dari proses evaluasi prototipe dianalisis untuk menentukan sejauh mana sistem mampu menangani

parsing pesan HL7, konversi *JSON*, serta pengiriman data melalui *REST API*. Selain itu, analisis fungsional digunakan untuk memastikan bahwa prototipe memenuhi *use case* yang diharapkan dan mampu meningkatkan interoperabilitas berdasarkan indikator seperti kelengkapan data, waktu transmisi, serta keberhasilan integrasi antarkomponen sistem.

Analisis juga dilakukan terhadap mekanisme publikasi data melalui API, khususnya untuk memastikan bahwa HIS hanya menerima data hasil verifikasi. Sistem menggunakan filter *status_api = 1* sehingga hanya data yang telah dikonfirmasi operator yang dapat dipublikasikan, sedangkan data mentah tetap disimpan sebagai arsip internal. Pengujian dilakukan menggunakan *Postman* untuk memverifikasi bahwa endpoint API menampilkan data terverifikasi dengan benar dan tidak mengekspos data mentah. Seluruh perubahan data dan riwayat verifikasi dicatat dalam tabel *lab_sync_audit*, yang menyimpan nilai sebelum dan sesudah koreksi dalam format *JSON*, beserta identitas operator dan waktu tindakan. Dengan analisis ini, validitas alur integrasi dan akurasi mekanisme kontrol data dapat dipastikan sebelum sistem diterapkan pada lingkungan operasional.

Untuk mendukung proses analisis fungsional, dibuat *activity diagram* yang menggambarkan alur kerja sistem mulai dari penerimaan pesan HL7 hingga publikasi data ke HIS.



Gambar 2. Activity diagram

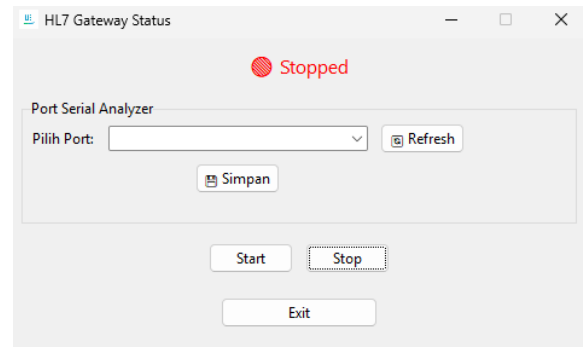
Diagram ini menggambarkan alur aktivitas mulai dari pengiriman pesan HL7 oleh *analyzer*, proses parsing dan konversi oleh *gateway Python*, verifikasi oleh operator, hingga publikasi data ke HIS.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pengembangan Sistem

Hasil pengembangan sistem menunjukkan bahwa implementasi protokol HL7 ORU^R01 berhasil direalisasikan melalui arsitektur dua komponen utama,

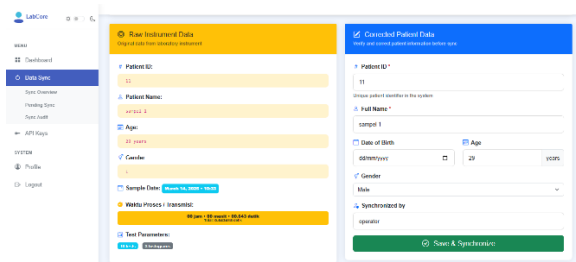
yaitu gateway berbasis *Python* dan server LIS berbasis *CodeIgniter 4*. *Gateway Python* berfungsi sebagai penerima data dari port serial *analyzer* dan sebagai parser untuk mengekstraksi informasi pasien dari segmen PID serta hasil pemeriksaan dari segmen OBX. Pesan HL7 yang diterima kemudian dikonversi menjadi format *JSON* agar dapat diproses lebih lanjut oleh server LIS.



Gambar 3. Tampilan aplikasi gateway HL7 berbasis Python

Aplikasi *EXE* ini berfungsi untuk memilih port komunikasi *analyzer*, menginisiasi koneksi, dan menjalankan proses penerimaan pesan HL7 ORU^R01 secara otomatis.

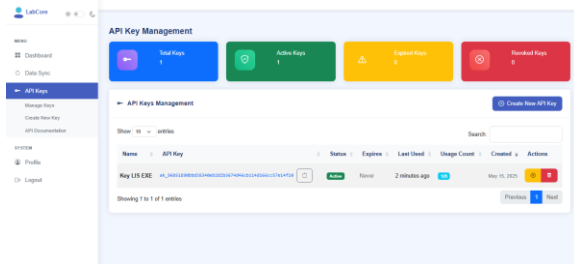
Pada sisi server, *CodeIgniter 4* berperan sebagai pusat pengelolaan data laboratorium dan integrator yang meneruskan hasil pemeriksaan ke HIS. Sistem menyediakan antarmuka verifikasi yang memungkinkan operator melakukan koreksi terhadap identitas pasien, meliputi patient ID, nama lengkap, tanggal lahir, usia, dan jenis kelamin. Setelah verifikasi dilakukan, sistem menyimpan dua versi data, yaitu data asli HL7 sebagai rekaman mentah dan data hasil verifikasi sebagai data akhir yang akan dipublikasikan melalui API. Pendekatan ini memastikan akurasi data sekaligus menjaga *traceability* dalam lingkungan laboratorium.



Gambar 4. Antarmuka verifikasi data pasien pada sistem LIS

Tampilan ini menyajikan perbandingan antara data mentah yang diterima dari instrumen (*Raw Instrument Data*), data yang telah dikoreksi oleh operator (*Corrected Patient Data*), serta informasi waktu proses atau transmisi sebelum hasil tersebut dipublikasikan ke dalam sistem HIS.

Untuk menjamin keamanan pertukaran data, setiap komunikasi antara *gateway* dan LIS, serta antara LIS dan HIS, dilindungi oleh mekanisme *API key authentication*. Pengaturan *API key* dikelola melalui modul manajemen *API* pada *CodeIgniter* sehingga hanya aplikasi yang terotorisasi yang dapat mengakses sumber data. Dengan arsitektur ini, sistem mampu mengintegrasikan data dari *analyzer* ke HIS secara otomatis, terstruktur, dan aman.



Gambar 5. Modul manajemen API key pada server LIS

Modul ini digunakan untuk menghasilkan, mengaktifkan, atau mencabut kunci *API* untuk memastikan bahwa hanya aplikasi yang terotorisasi yang dapat mengakses *endpoint* sistem.

4.2. Hasil Pengujian Integrasi Sistem

Pengujian integrasi dilakukan untuk memastikan bahwa seluruh alur pertukaran data antara *analyzer*, *gateway* HL7 berbasis *Python*, server LIS, dan HIS dapat berjalan secara otomatis, konsisten, serta tanpa kesalahan. Pengujian mencakup pengujian fungsional, pengujian kinerja, dan pengujian integrasi antarkomponen.

4.3. Pengujian Fungsional

Pengujian fungsional dilakukan untuk memastikan bahwa setiap proses utama pada sistem bekerja sesuai dengan rancangan. Ringkasan hasil pengujian fungsional sistem integrasi LIS disajikan pada Tabel 1.

Tabel 1. Hasil pengujian fungsional sistem integrasi LIS

No	Proses yang Diuji	Hasil yang Diharapkan	Hasil Aktual	Status
1	Penerimaan pesan HL7 ORU^R01	Pesan HL7 dari analyzer diterima tanpa error komunikasi	Seluruh pesan uji (sampel 1–3) diterima tanpa error dan tersimpan dalam log <i>gateway</i>	Lulus
2	Parsing segmen PID dan OBX	Seluruh elemen HL7 terpetakan ke struktur data internal	<i>Patient ID</i> , nama, usia, <i>gender</i> , dan parameter hasil berhasil diparsing ke objek data	Lulus
3	Konversi HL7 ke JSON	Payload JSON valid dan siap dikirim ke API LIS	Seluruh pesan HL7 berhasil dikonversi ke JSON yang valid menggunakan parser internal	Lulus
4	Verifikasi data pasien	Operator dapat mengoreksi identitas pasien dan menyimpan hasil verifikasi	Koreksi pada patient ID, nama, tanggal lahir, usia, dan gender tersimpan dalam <i>database</i> LIS	Lulus
5	Penyimpanan dua versi data	Sistem menyimpan <i>raw data</i> HL7 dan data hasil verifikasi	Kedua versi data tersimpan dalam tabel terpisah dan dapat ditelusuri kembali	Lulus
6	Publikasi data ke HIS melalui API	Data terverifikasi dapat dipublikasikan ke endpoint HIS	Endpoint API merespons 200 OK untuk seluruh permintaan publikasi data pemeriksaan	Lulus
7	Pencatatan aktivitas sinkronisasi (audit)	Setiap sinkronisasi tercatat dalam log audit	Modul <i>Sync Audit</i> mencatat seluruh aktivitas dengan status <i>Sync Patient</i> disertai waktu dan identitas pengguna	Lulus
8	Eksekusi aplikasi EXE	File HL7Gateway.exe dapat dijalankan tanpa error	Aplikasi GUI berjalan stabil, ikon tray aktif, dan status “Running/Stopped” tampil sesuai kondisi sistem	Lulus
9	Pemilihan Port Serial	Port komunikasi <i>analyzer</i> dapat dipilih dan disimpan pada config.ini	Port terpilih tersimpan permanen dan digunakan otomatis saat aplikasi dijalankan ulang	Lulus
10	Koneksi dan Logging	Sistem menulis log aktivitas ke folder logs/	File log harian terbentuk otomatis, mencatat semua komunikasi HL7 dan respons API	Lulus

Hasil pada Tabel 1 menunjukkan bahwa seluruh fungsi utama pada sistem berjalan sesuai dengan rancangan, tanpa ditemukan kegagalan pada skenario uji yang dilakukan.

4.4. Pengujian Kinerja

Pengujian kinerja difokuskan pada waktu transmisi, yaitu durasi sejak operator melakukan aksi *Save & Synchronize* pada LIS sampai data dinyatakan tersinkronisasi di log audit. Pengujian dilakukan

terhadap tiga sampel data uji, dan hasilnya disajikan pada Tabel 2.

Tabel 2. Hasil pengujian kinerja sistem

No	Sampel	Waktu transmisi (detik)
1	Sampel 1	0,093790
2	Sampel 2	0,255634
3	Sampel 3	0,377116

Rata-rata waktu transmisi dari ketiga sampel tersebut sekitar 0,24 detik, sehingga seluruh proses

pengiriman data terverifikasi ke HIS berlangsung jauh di bawah satu detik. Hal ini menunjukkan bahwa arsitektur integrasi yang digunakan mampu memberikan kinerja yang cepat dan mendukung kebutuhan layanan yang menuntut respon waktu singkat.

4.5. Pengujian Integrasi Antar Komponen

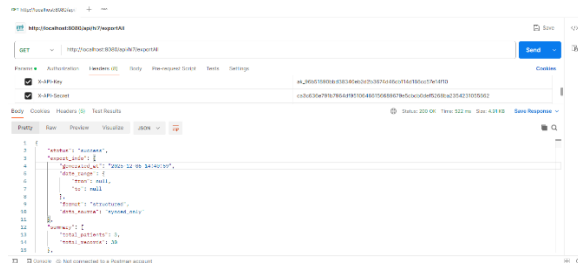
Pengujian integrasi dilakukan untuk memastikan bahwa setiap komponen *analyzer*, *gateway Python*, LIS, dan HIS dapat saling berkomunikasi dengan baik dalam satu alur kerja. Hasil pengujian integrasi untuk tiga sampel uji ditunjukkan pada Tabel 3.

Tabel 3. Hasil pengujian integrasi sistem

No	Sampel	Jalur Integrasi yang diuji	Hasil
1	Sampel 1	<i>Analyzer</i> → <i>Gateway</i> → LIS → HIS	Pesan HL7 diterima, diparsing, diverifikasi, dan dipublikasikan sukses
2	Sampel 2	<i>Analyzer</i> → <i>Gateway</i> → LIS → HIS	Seluruh tahapan integrasi berjalan tanpa error
3	Sampel 3	<i>Analyzer</i> → <i>Gateway</i> → LIS → HIS	HIS menerima data dan API merespons <i>200 OK</i>

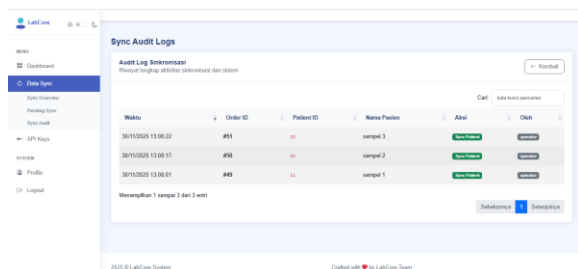
Tingkat keberhasilan integrasi diukur menggunakan modul *Sync Audit* pada LIS. Untuk seluruh sampel uji, entri yang tercatat menunjukkan status *Sync Patient* tanpa adanya status gagal. Hal ini mengindikasikan bahwa tingkat keberhasilan transmisi selama pengujian adalah 100%.

Hasil pengujian menggunakan Postman menunjukkan bahwa *endpoint API* hanya menampilkan data hasil verifikasi dan tidak menyertakan data mentah, sehingga HIS hanya menerima hasil yang telah divalidasi oleh operator.



Gambar 6. Hasil pengujian *endpoint API* menggunakan Postman

Gambar ini menunjukkan hasil pengujian *endpoint API* yang digunakan HIS untuk mengambil data hasil pemeriksaan. Respons *200 OK* dan *payload JSON* yang hanya memuat data dengan atribut *status_api = 1* menegaskan bahwa sistem LIS hanya mempublikasikan data yang telah diverifikasi operator, sehingga data mentah HL7 tidak diekspos. Temuan ini mengonfirmasi bahwa mekanisme publikasi API berjalan konsisten dan sesuai dengan prinsip validasi klinis.



Gambar 7. Sync audit

Gambar ini menunjukkan catatan aktivitas sinkronisasi pada sistem LIS. Seluruh entri berstatus *Sync Patient*, yang menandakan bahwa data hasil verifikasi berhasil dipublikasikan tanpa kesalahan. Modul ini menjadi dasar penentuan tingkat keberhasilan transmisi sebesar 100%.

5. KESIMPULAN DAN SARAN

Penelitian ini membuktikan bahwa implementasi protokol *HL7 ORU^R01* pada sistem informasi laboratorium mampu meningkatkan akurasi, interoperabilitas, dan efisiensi pertukaran data antara *analyzer*, LIS, dan HIS melalui arsitektur integrasi yang memadukan *gateway Python* dan server LIS berbasis CodeIgniter 4. *Gateway Python* terbukti stabil dalam menerima pesan HL7, melakukan parsing segmen PID dan OBX secara konsisten, serta mengonversinya menjadi JSON yang valid, sementara LIS menyediakan mekanisme verifikasi untuk memastikan bahwa identitas pasien telah benar sebelum data dipublikasikan. Sistem juga menyimpan dua versi data, data mentah HL7 dan data hasil verifikasi sehingga *traceability* terjaga dan audit perubahan dapat dilakukan dengan baik. Proses publikasi data melalui API menunjukkan performa sangat baik dengan respons *200 OK*, waktu transmisi kurang dari satu detik, serta tingkat keberhasilan sinkronisasi sebesar 100 persen, dan pengujian menggunakan Postman menegaskan bahwa *endpoint API* hanya menampilkan data yang telah diverifikasi (*status_api = 1*), sehingga HIS menerima data yang telah divalidasi secara klinis. Untuk pengembangan selanjutnya, sistem disarankan diuji langsung dengan HIS operasional untuk memastikan kompatibilitas penuh, diperluas dukungannya terhadap standar interoperabilitas modern seperti HL7 ADT, OML, dan FHIR, diperkuat keamanannya melalui mekanisme otentikasi berbasis token seperti JWT atau OAuth2, serta dievaluasi performanya dalam skenario beban kerja tinggi guna memastikan kestabilan sistem pada penggunaan intensif.

DAFTAR PUSTAKA

- [1] C. R. McCudden, M. P. A. Henderson, and B. R. Jackson, "Laboratory information management," *Contemporary Practice in Clinical Chemistry*, pp. 301–321, Jan. 2020, doi: 10.1016/B978-0-12-815499-1.00018-1.
- [2] A. A. AlQudah, M. Al-Emran, and K. Shaalan, "Medical data integration using HL7 standards for patient's early identification," *PLoS One*, vol. 16, no. 12, p. e0262067, Dec. 2021, doi: 10.1371/JOURNAL.PONE.0262067.
- [3] M. Aouachria, M. Aouache, and I. Oudjoudi, "Base Frame of Algerian E-Health System: How to Ensure Consistency of Data Exchanged with Laboratory Management System Using BPM and IHE Profiles?," *Proceedings - 2021 International Conference on Information Systems and Advanced Technologies, ICISAT 2021*, 2021, doi: 10.1109/ICISAT54145.2021.9678457.
- [4] S. Schmidlehner, "Standards-based Clinical Data Repository," 2021, *Technische Universität Wien*. doi: 10.34726/HSS.2021.27681.
- [5] S. Kumar, P. Dinakaran, and M. A. Augie, "HL7 Standards in Public Health Disease Surveillance," *SARC Publishers*, vol. 4, no. 11, pp. 11–17, Nov. 2025, doi: 10.5281/zenodo.17508110.
- [6] T. M. Mtonga, "Design, Development, and Evaluation of a Laboratory Order Entry and Results Review Application for a Low-Resource Inpatient Setting," 2020.
- [7] M. Gimah and L. Eli, "Towards Seamless Data Integration: A Comparative Study of HL7, FHIR, and LOINC in SaaS Laboratory Systems," Jun. 20, 2025, doi: 10.20944/preprints202506.1676.v1.
- [8] E. M. Hashem, M. F. Y. Shalaby, M. S. Mabrouk, and Samir. Y. Marzouk, "Integrative software design for Hospital Information Systems," *Mansoura Engineering Journal*, vol. 48, no. 4, Jul. 2023, doi: 10.58491/2735-4202.3057.
- [9] F. A. Reegu *et al.*, "Blockchain-Based Framework for Interoperable Electronic Health Records for an Improved Healthcare System," *Sustainability (Switzerland)*, vol. 15, no. 8, Apr. 2023, doi: 10.3390/su15086337.
- [10] N. Thaker and A. Shukla, "Python as Multi Paradigm Programming Language," *Int J Comput Appl*, vol. 177, no. 31, pp. 38–42, Jan. 2020, doi: 10.5120/ijca2020919775.
- [11] Suheri, "Penggunaan Framework Codeigniter Dalam Pembuatan Web Profil Program Studi Teknik Elektro Universitas Pembangunan Panca Budi Medan," *Jurnal Nasional Teknologi Komputer*, vol. 3, no. 3, pp. 227–234, Jul. 2023, doi: 10.61306/jnastek.v3i3.98.
- [12] A. Golmohammadi, M. Zhang, and A. Arcuri, "Testing RESTful APIs: A Survey," *ACM Transactions on Software Engineering and Methodology*, vol. 33, no. 1, 2023, doi: 10.1145/3617175.
- [13] E. Bjarnason, F. Lang, and A. Mjöberg, "An empirically based model of software prototyping: a mapping study and a multi-case study," *Empir Softw Eng*, vol. 28, no. 5, Sep. 2023, doi: 10.1007/s10664-023-10331-w