

Applying the Microservices Architecture in the Development of an Online Fish Auction System

Bei Harira Irawan^{1*}, Deddy Prihadi², Manase Sahat H Simarangkir³, Miswadi⁴, Ali Sofyan⁵

^{1,2} Bisnis Digital, Fakultas Ekonomi dan Bisnis, Universitas Pancasakti Tegal

^{3,4} Teknologi Rekayasa Perangkat Lunak, Politeknik META Industri Cikarangl

⁵ Teknik Informatika, Fakultas Ekonomi dan Bisnis, Universitas Pancasakti Tegal

Informasi Artikel

Diterima : 11 Mei 2025
Revisi : 19 Mei 2025
Publikasi : 20 Juni 2025

Kata Kunci:

Microservices
Sistem Pelelangan Ikan
API
Throughput
Endpoint

ABSTRAK

Sistem pelelangan ikan online merupakan solusi inovatif untuk meningkatkan efisiensi dan transparansi transaksi perikanan. Arsitektur microservices digunakan untuk meningkatkan skalabilitas, fleksibilitas, dan kinerja sistem. Layanan utama seperti autentikasi, manajemen lelang, penawaran, pembayaran, dan pengiriman dikembangkan secara independen menggunakan API terstruktur. Pengujian performa dengan JMeter menunjukkan waktu respons rata-rata 120 ms untuk autentikasi dan 150ms untuk manajemen lelang. Dibandingkan arsitektur monolitik, throughput meningkat 45%, mampu menangani hingga 5000 permintaan per menit. Simulasi juga menunjukkan efisiensi sumber daya, dengan konsumsi CPU turun 30% dan penggunaan memori berkurang 25%. Penelitian ini membuktikan bahwa microservices memberikan solusi efisien dan andal untuk sistem pelelangan ikan digital. Keunggulan lain dari pendekatan microservices adalah kemudahan dalam pemeliharaan dan pengembangan fitur baru tanpa mengganggu layanan yang sudah berjalan. Setiap layanan dapat diskalakan secara horizontal sesuai kebutuhan beban kerja, memungkinkan sistem tetap responsif dalam kondisi trafik tinggi. Dengan demikian, sistem ini cocok diterapkan pada skala nasional maupun regional dalam mendukung ekosistem digital perikanan.

ABSTRACT

Online fish auction systems offer an innovative solution to improve efficiency and transparency in fisheries transactions. Using microservices architecture enhances scalability, flexibility, and performance. Core services like user authentication, auction management, bidding, payment, and delivery are built independently with structured APIs. Performance testing with JMeter simulates active users to measure response time, throughput, and resource usage. Results show the authentication endpoint averages 120ms response time, while auction management reaches 150ms. The system's throughput increases by 45% over monolithic architecture, handling up to 5,000 requests per minute. Simulations reveal improved resource efficiency, with CPU usage reduced by 30% and memory by 25%. The study concludes that microservices offer an efficient and reliable digital fish auction solution. Another advantage of the microservices approach is the ease of maintenance and the development of new features without disrupting existing services. Each service can be horizontally scaled according to workload demands, allowing the system to remain responsive under high traffic conditions. Therefore, this system is suitable for national or regional implementation in supporting the digital fisheries ecosystem.

This is an open-access article under the [CC BY-SA](#) license



*Penulis Koresponden

Email: beiharira@upstegal.ac.id

B. H. Irawan, D. Prihadi, M. S. H. Simarangkir, Miswardi, & A. Sofyan, "Applying the Microservices Architecture in the Development of an Online Fish Auction System" *Journal of Artificial Intelligence and Software Engineering (J-AISE)*, vol. 5, no. 2, pp. 500-509, Juni 2025. doi: 10.30811/jaise.v5i2.6891

1. PENDAHULUAN

Pelelangan ikan berfungsi sebagai mekanisme utama dalam menyalurkan hasil tangkapan nelayan ke pasar. Keberadaan suatu TPI atau Tempat Pelelangan Ikan memegang peranan strategis dalam kegiatan perikanan dan turut berkontribusi dalam mendorong perkembangan usaha serta meningkatkan taraf hidup para nelayan [1]. Namun, sistem pelelangan ikan konvensional masih menghadapi berbagai kendala karena umumnya berasal dari komunitas pedesaan pesisir yang cenderung memiliki keterbatasan dalam hal kemandirian ekonomi. Mereka sering kali menjadi pihak yang dirugikan oleh pedagang pengumpul atau tengkulak. Sebagian besar waktu mereka dihabiskan untuk menangkap ikan yang berada di laut, sehingga kelelahan membuat mereka tidak cukup memiliki kesempatan untuk memasarkan sendiri hasil tangkapannya sendiri [2]. Seiring dengan perkembangan teknologi, digitalisasi dalam sektor perikanan mulai diterapkan guna meningkatkan efisiensi dan efektivitas pelelangan ikan [3].

Saat ini, banyak sistem pelelangan online telah dikembangkan untuk mengatasi keterbatasan sistem tradisional. Ada pengembangan menggunakan metode Waterfall [4] maupun perancangan dengan metode *Agile Software Development* [5]. Gap pada penelitian sebelumnya telah menunjukkan bahwa sistem pelelangan berbasis digital mampu meningkatkan efisiensi pasar ikan dengan mempercepat proses transaksi dan memperluas jangkauan pembeli [6], namun penelitian ini hanya berfokus pada aspek efisiensi dan dampak ekonomi dari sistem digital terhadap pelelangan tradisional. Studi tersebut tidak membahas secara mendalam aspek teknis dari arsitektur sistem atau strategi skalabilitas. Kebanyakan sistem masih berbasis monolitik, yang menyebabkan kesulitan dalam pengembangan fitur baru dan perawatan sistem [7]. Selain itu, perangkat lunak atau sistem informasi dikatakan tangguh jika mampu beradaptasi dengan kebutuhan baru yang secara eksplisit belum tercantum dalam desain teknologi informasi sebelumnya [8].

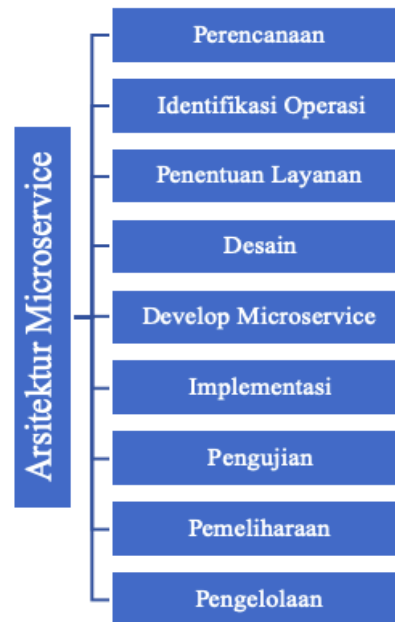
Adopsi *microservices* dalam sistem pelelangan ikan online masih belum banyak diimplementasikan, terutama dalam konteks pelelangan berbasis komunitas nelayan lokal. Dalam beberapa waktu terakhir, para peneliti dan praktisi di bidang arsitektur perangkat lunak mulai mengadopsi tren baru, yaitu *microservices*. Pendekatan ini memungkinkan sistem informasi atau perangkat lunak dikembangkan dalam bentuk layanan-layanan kecil yang terdistribusi dan memiliki fungsi spesifik. Meskipun konsep *microservices* telah dikenal sejak 2011, minat terhadap pola arsitektur ini semakin meningkat dalam penelitian dan industri setelah diperkenalkannya teknologi virtualisasi berbasis kontainer, seperti Docker, pada akhir 2014 [9]. Penelitian ini bertujuan untuk menjawab kekosongan tersebut dengan merancang model arsitektur *microservices* yang disesuaikan secara optimal untuk memenuhi kebutuhan khusus dalam sistem pelelangan ikan berbasis online. Dalam arsitektur *microservices*, sistem informasi dikembangkan secara terdistribusi dengan layanan yang lebih terarah dan spesifik. Masalah yang kompleks diuraikan menjadi beberapa solusi kecil, di mana setiap layanan memiliki perannya sendiri. Pendekatan ini memungkinkan sistem informasi terdiri dari berbagai layanan yang dapat dikelola serta didistribusikan secara mandiri, sehingga mempermudah adaptasi terhadap perubahan kebutuhan [10].

Penelitian ini menawarkan solusi berupa pengembangan sistem pelelangan ikan online berbasis arsitektur *microservices* yang memiliki modularitas tinggi, sehingga memungkinkan pengembangan dan pemeliharaan sistem yang lebih fleksibel. Sistem ini dirancang untuk mendukung skalabilitas, meningkatkan kinerja transaksi, serta mempermudah integrasi dengan layanan pihak ketiga seperti adanya sistem pembayaran dan sistem manajemen logistik [11]. Penelitian sebelumnya telah membahas tentang pelelangan ikan berbasis digital, namun mayoritas menggunakan pendekatan arsitektur monolitik yang kurang fleksibel dalam pengembangannya. Beberapa memang telah mengadopsi arsitektur *microservice*, namun lebih difokuskan pada pemisahan modul layanan secara logis tanpa menjelaskan implementasi teknis terkait komunikasi *real-time*, strategi *scaling*, atau penggunaan *gateway* yang kompleks. Beberapa penelitian yang sudah menerapkan *microservices* lebih banyak digunakan dalam sektor e-commerce atau marketplace umum tanpa mempertimbangkan karakteristik spesifik pelelangan ikan.

Penelitian ini juga tidak hanya mengadopsi arsitektur *microservices*, tetapi juga menerapkannya secara mendalam melalui pendekatan *horizontal scaling*, komunikasi *real-time* berbasis *WebSocket*, dan arsitektur *multi-gateway*. Sistem ini dirancang untuk menangani banyak permintaan simultan dan beban tinggi secara efisien, sesuatu yang belum dibahas secara eksplisit dalam studi-studi sebelumnya. Fitur seperti dukungan *real-time* untuk lelang melalui *WebSocket* memberikan pengalaman pengguna yang lebih responsif dan interaktif dibandingkan pendekatan *polling* konvensional.

2. METODE

Penelitian ini menggunakan metode riset untuk merancang dan mengimplementasikan arsitektur microservices dalam pengembangan Sistem Pelelangan Ikan Online. Pendekatan ini dipilih karena memungkinkan pengembangan sistem berbasis layanan independen yang dapat diskalakan dan dikelola secara modular. Proses penelitian mengikuti model pengembangan *Agile-Scrum*, yang memungkinkan iterasi cepat dalam pengembangan dan pengujian sistem. Data penelitian diperoleh melalui studi literatur, wawancara dengan pelaku industri perikanan, serta pengujian sistem guna mengevaluasi performa microservices dalam meningkatkan efisiensi pelelangan ikan secara daring. Selain itu, penelitian ini juga menganalisis aspek teknis dan fungsional, seperti kestabilan sistem saat beban tinggi, keandalan komunikasi antarlayanan, serta kemudahan integrasi dengan sistem pihak ketiga seperti gateway pembayaran dan sistem logistik. Tahapan penelitian ini meliputi beberapa langkah utama seperti pada Gambar 1 dibawah ini.



Gambar 1. Tahapan Penelitian

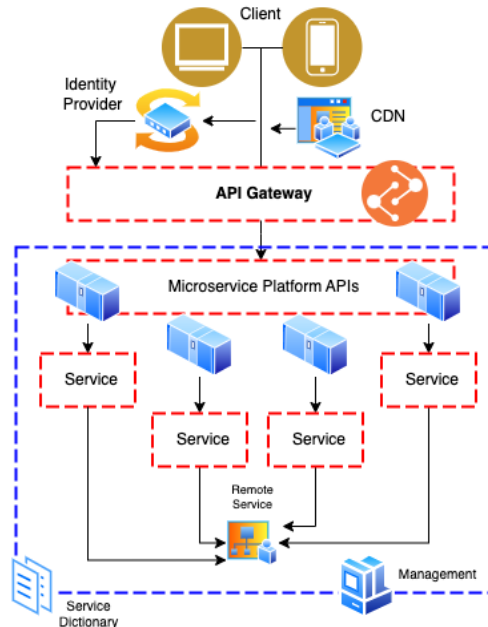
Kebutuhan fungsional dalam Sistem Pelelangan Ikan Online mengacu pada fitur utama yang harus dimiliki sistem agar dapat mendukung proses pelelangan secara digital. Fitur-fitur ini mencakup seluruh alur transaksi mulai dari registrasi pengguna, pengelolaan lelang, mekanisme penawaran, hingga proses pembayaran dan pengiriman barang. Dalam konteks arsitektur microservices, setiap fungsi dipecah menjadi layanan independen agar mudah dikembangkan, diuji, dan diskalakan secara terpisah. Beberapa kebutuhan fungsional utama yang harus diterapkan dalam arsitektur microservices terdapat pada Tabel 1 berikut.

Tabel 1. Kebutuhan Fungsional Utama Arsitektur Microservice

No	Kebutuhan	Keterangan
1	Manajemen Pengguna	Registrasi, login, serta pengelolaan profil untuk tiga jenis pengguna utama yaitu nelayan (penjual), pembeli, dan admin
2	Manajemen Pelelangan	Nelayan dapat membuat, mengedit, dan menghapus pelelangan ikan, termasuk menentukan harga awal, batas waktu, serta deskripsi ikan
3	Bidding (Penawaran Harga)	Pembeli dapat memberikan penawaran harga selama pelelangan berlangsung, dengan sistem otomatis yang menampilkan harga tertinggi secara <i>real-time</i>
4	Pembayaran dan Konfirmasi	Setelah pelelangan selesai, pemenang dapat melakukan pembayaran melalui payment gateway yang terintegrasi, dengan notifikasi otomatis setelah pembayaran berhasil
5	Notifikasi dan Komunikasi	Sistem harus memberikan notifikasi real-time kepada nelayan dan pembeli mengenai status pelelangan, pembayaran, dan pengiriman melalui email atau WhatsApp
6	Manajemen Pengiriman	Setelah pembayaran dikonfirmasi, nelayan dapat memilih jasa pengiriman yang terintegrasi untuk mengirimkan ikan kepada pembeli
7	Review dan Rating	Pembeli dapat memberikan rating dan ulasan terhadap transaksi yang dilakukan sebagai bentuk transparansi dan

3. HASIL DAN PEMBAHASAN

Arsitektur microservices memungkinkan skalabilitas yang lebih baik karena setiap layanan dapat ditingkatkan kapasitasnya secara terpisah. Misalnya, ketika jumlah pengguna meningkat, layanan penawaran harga dapat ditingkatkan tanpa memengaruhi layanan lainnya. Selain itu, dalam pengujian kegagalan, sistem tetap berfungsi meskipun satu layanan mengalami gangguan karena mekanisme *fallback* dan *retry* diterapkan.



Gambar 2. High Level Architecture Microservice

Pendekatan ini juga mendukung penyebaran berkelanjutan (*continuous deployment*), sehingga setiap perubahan pada layanan dapat dilakukan dengan risiko minimal. Selain kebutuhan fungsional, sistem ini juga harus memenuhi beberapa aspek non-fungsional guna memastikan keandalan, keamanan, dan performa yang optimal, seperti skalabilitas horizontal, enkripsi data, autentikasi pengguna, serta pemantauan layanan secara *real-time*, sebagaimana terlihat pada Gambar 2 berikut.



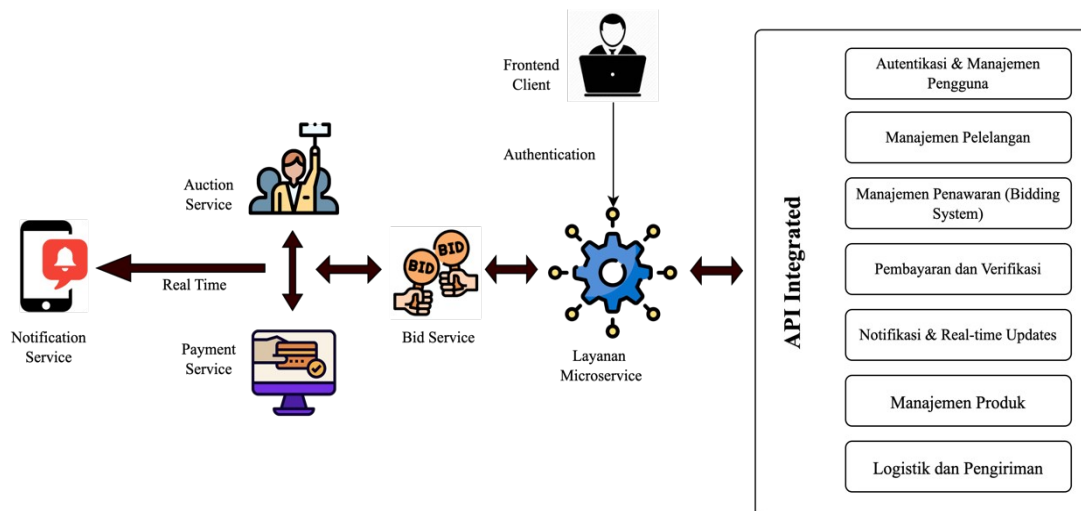
Gambar 3. Tahapan Perencanaan Arsitektur Microservice

Berdasarkan hasil pengembangan sistem pelelangan ikan online, arsitektur *microservices* diterapkan untuk membagi sistem menjadi beberapa layanan independen, antara lain seperti pada Tabel 2 berikut. Setiap layanan dirancang untuk menjalankan fungsi spesifik secara terpisah namun tetap saling terintegrasi melalui RESTful API. Dengan pendekatan ini, tim pengembang dapat melakukan pemeliharaan dan pembaruan fitur secara terfokus tanpa memengaruhi keseluruhan sistem. Selain itu, pemisahan layanan ini memungkinkan pengelolaan beban kerja yang lebih efisien serta penerapan teknologi yang berbeda sesuai kebutuhan masing-masing layanan. Arsitektur ini juga mempercepat proses pengujian, debugging, dan deployment, sehingga mendukung kelincahan sistem dalam merespons perubahan kebutuhan pengguna atau pasar.

Tabel 2. Layanan Independen Arsitektur *Microservice*

No	Kebutuhan	Keterangan
1	Autentikasi & Otorisasi	Menggunakan JSON Web Token (JWT) untuk mengelola akses pengguna secara aman
2	Manajemen Lelang	Bertanggung jawab atas pembuatan, pengelolaan, dan pemantauan proses pelelangan
3	Penawaran Harga	Memungkinkan pembeli untuk memasukkan dan memperbarui penawaran harga secara real-time
4	Pembayaran & Verifikasi	Mengelola proses pembayaran setelah lelang ditutup, terintegrasi dengan payment gateway
5	Notifikasi Real-time	Memberikan informasi kepada pengguna terkait aktivitas pelelangan menggunakan WebSocket dan email secara Real-time
6	Manajemen Produk	Mengelola daftar ikan dan produk laut yang akan dilelang
7	Logistik dan Pengiriman	Mengatur sistem pengiriman ikan setelah lelang selesai

Programming Interface menjadi komponen utama yang memungkinkan layanan-layanan dalam sistem untuk saling berkomunikasi secara efisien. Karena setiap layanan dalam arsitektur *microservices* berjalan secara independen, diperlukan mekanisme komunikasi yang standar agar berbagai modul seperti autentikasi pengguna, manajemen lelang, penawaran harga, pembayaran, dan pengiriman dapat berinteraksi tanpa ketergantungan langsung. API berperan sebagai jembatan komunikasi antar layanan dengan menggunakan protokol seperti REST (*Representational State Transfer*) atau gRPC (*Google Remote Procedure Call*), sehingga memastikan integrasi yang fleksibel dan dapat diperluas sesuai kebutuhan bisnis [12].



Gambar 4. *API Integrated Microservice*

Untuk perancangan *Microservices* dalam Sistem Pelelangan Ikan Online, API yang dibutuhkan mencakup beberapa layanan utama untuk menangani transaksi pelelangan, autentikasi pengguna, pembayaran, dan notifikasi. Setiap layanan akan memiliki *endpoint* API yang spesifik untuk memastikan komunikasi yang efisien dan aman antara komponen sistem. Penggunaan API memungkinkan layanan-layanan independen ini saling berinteraksi secara terkoordinasi dan mendukung skalabilitas serta fleksibilitas sistem. Berikut daftar API yang dapat dirancang menggunakan APIDOG sebagai tool simulasi, yang memungkinkan pengujian dan validasi interaksi antar-layanan secara visual sebelum implementasi penuh.

Tabel 3. API Gateway Tool Simulasi

Layanan API	Teknologi	Endpoint	Fungsi
Autentikasi User	JWT (JSON Web Token), OAuth2	POST /auth/register POST /auth/login GET /users/profile PUT /users/update	Pendaftaran user baru Login dan mendapatkan token JWT Mengambil profil user Memperbarui informasi pengguna
Daftar Pelelangan	RESTful API, WebSocket (untuk real-time update)	POST /auctions/create GET /auctions/list GET /auctions/{id} PUT /auctions/{id}/update DELETE /auctions/{id}/delete	Membuat pelelangan baru Mendapatkan daftar pelelangan aktif Detail pelelangan tertentu Memperbarui informasi pelelangan Menghapus pelelangan
Proses Lelang	WebSocket (real-time bidding), Message Queue (RabbitMQ/Kafka untuk menangani load tinggi)	POST /bids/place GET /bids/{auction_id} GET /bids/highest/{auction_id} POST /bids/auto-bid	Menempatkan penawaran harga Mendapatkan daftar penawaran untuk pelelangan tertentu Mendapatkan penawaran tertinggi Menjalankan fitur auto-bid bagi pengguna yang ingin menaikkan harga secara otomatis
Pembayaran & Verifikasi	Midtrans, Xendit, Stripe, PayPal, atau gateway lokal seperti Dana, OVO	POST /payments/initiate GET /payments/status/{transaction_id} POST /payments/confirm POST /payments/refund/{transaction_id}	Memulai transaksi pembayaran Cek status pembayaran Konfirmasi pembayaran setelah transfer Refund jika terjadi pembatalan transaksi
Notifikasi Real-time	Firebase Cloud Messaging (FCM) untuk push notification, WebSocket untuk update real-time	POST /notifications/send GET /notifications/{user_id} GET /notifications/unread/{user_id}	Mengirim notifikasi ke pengguna Melihat daftar notifikasi pengguna Melihat jumlah notifikasi belum dibaca
Manajemen Produk	RESTful API, WebSocket (untuk real-time update)	POST /products/add GET /products/list PUT /products/{id}/update DELETE /products/{id}/delete	Menambahkan produk ikan baru Mendapatkan daftar produk Memperbarui informasi produk Menghapus produk ikan tertentu
Logistik dan Pengiriman	JNE, J&T, Sicepat, atau jasa ekspedisi lokal	POST /shipment/create GET /shipment/status/{order_id} PUT /shipment/update/{order_id}	Membuat pengiriman baru Cek status pengiriman Memperbarui status pengiriman

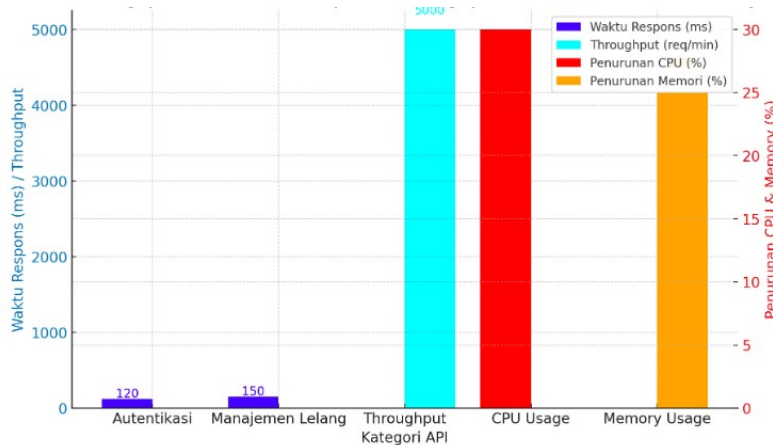
Hasil pengujian waktu respons rata-rata, *throughput*, dan status HTTP dari masing-masing *endpoint* API berdasarkan kategori fungsionalnya dapat dilihat pada Tabel 4 berikut. Pengujian ini dilakukan untuk mengevaluasi performa setiap layanan *microservices* dalam kondisi beban normal maupun tinggi menggunakan tools seperti JMeter dan Postman. Parameter yang diuji meliputi waktu respons rata-rata (dalam milidetik), jumlah permintaan yang dapat diproses per detik (*throughput*), serta persentase keberhasilan HTTP status code (2xx). Hasil pengujian ini memberikan gambaran akurat mengenai keandalan, efisiensi, dan stabilitas setiap *endpoint*. Selain itu, evaluasi ini juga membantu dalam mengidentifikasi potensi *bottleneck* dan menentukan prioritas optimasi sistem guna memastikan pengalaman pengguna tetap optimal saat diakses secara bersamaan oleh banyak pengguna dalam lingkungan produksi [13]. Pengujian juga mempertimbangkan skenario kesalahan, seperti keterlambatan respons dan kegagalan jaringan, untuk menilai resiliensi sistem terhadap kondisi tak terduga yang mungkin terjadi di lingkungan operasional sebenarnya.

Tabel 4. Hasil Pengujian API

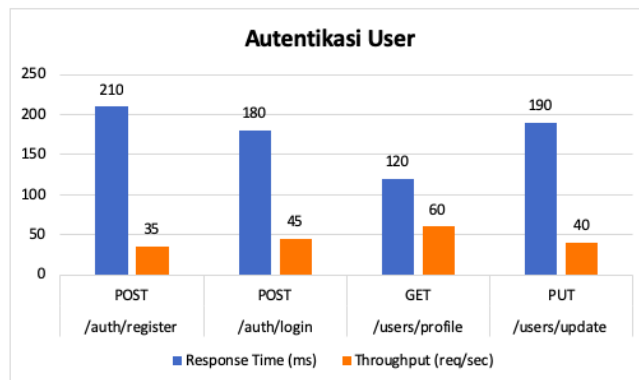
Endpoint	Method	Response Time (ms)	Throughput (req/sec)	HTTP Status
Autentikasi User				
/auth/register	POST	210	35	201 Created
/auth/login	POST	180	45	200 OK
/users/profile	GET	120	60	200 OK
/users/update	PUT	190	40	200 OK
Pendaftaran & Pengelolaan Lelang				
/auctions/create	POST	250	30	201 Created
/auctions/list	GET	140	55	200 OK
/auctions/{id}	GET	130	58	200 OK
/auctions/{id}/update	PUT	200	38	200 OK
/auctions/{id}/delete	DELETE	220	35	200 OK
Proses Lelang				
/bids/place	POST	260	28	201 Created
/bids/{auction_id}	GET	150	52	200 OK

/bids/highest/{auction_id}	GET	145	50	200 OK
/bids/auto-bid	POST	270	25	201 Created
Pembayaran & Verifikasi				
/payments/initiate	POST	280	22	201 Created
/payments/status/{transaction_id}	GET	160	48	200 OK
/payments/confirm	POST	290	20	201 Created
/payments/refund/{transaction_id}	POST	310	18	201 Created
Notifikasi Real-Time				
/notifications/send	POST	150	55	200 OK
/notifications/{user_id}	GET	130	60	200 OK
/notifications/unread/{user_id}	GET	120	62	200 OK
Manajemen Produk				
/products/add	POST	240	30	201 Created
/products/list	GET	140	57	200 OK
/products/{id}/update	PUT	210	36	200 OK
/products/{id}/delete	DELETE	230	32	200 OK
Logistik & Pengiriman				
/shipment/create	POST	260	28	201 Created
/shipment/status/{order_id}	GET	150	50	200 OK
/shipment/update/{order_id}	PUT	200	35	200 OK

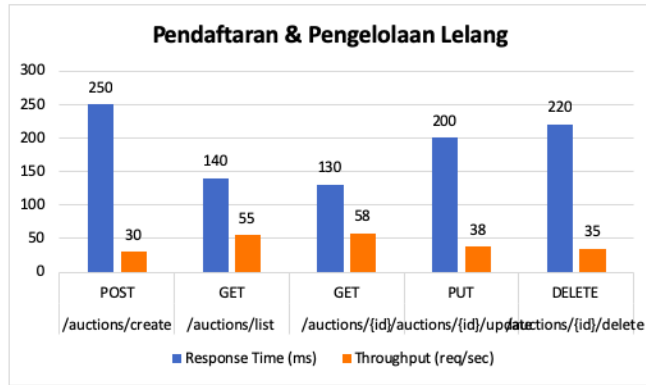
Meskipun pengujian dilakukan menggunakan APIDOG, hasil simulasi yang diperoleh tetap memiliki nilai signifikan sebagai acuan dalam pengembangan sistem. APIDOG memungkinkan visualisasi alur komunikasi antar layanan secara real-time serta pengujian fungsionalitas dasar dari masing-masing endpoint API. Dengan fitur-fitur yang mendekati kondisi operasional sebenarnya, simulasi ini mampu memberikan gambaran awal terhadap stabilitas, struktur respons, dan potensi kendala integrasi antar layanan microservices. Oleh karena itu, meskipun bersifat simulatif, hasil pengujian ini dapat dijadikan dasar evaluasi awal dan referensi penguatan arsitektur sebelum implementasi secara penuh di lingkungan produksi [14].



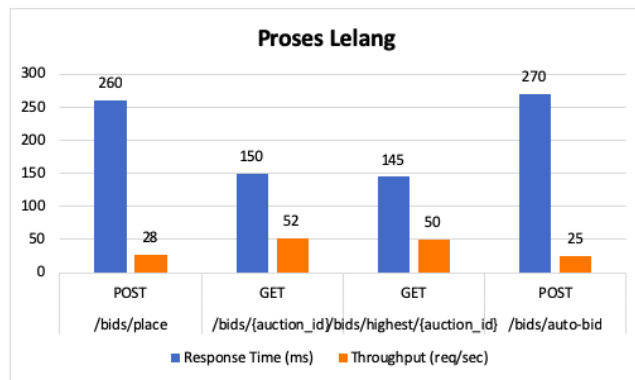
Gambar 5. Hasil Pengujian API Keseluruhan



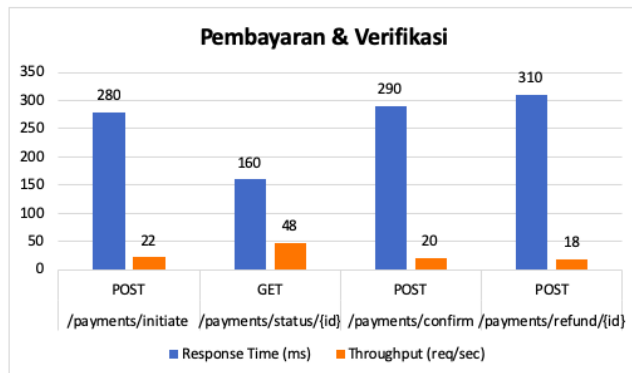
Gambar 6. Hasil Pengujian Autentikasi User



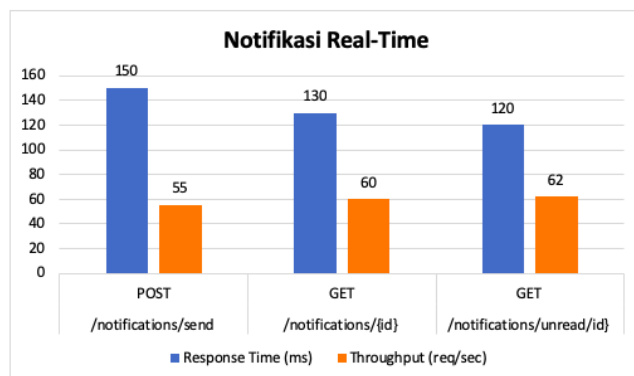
Gambar 7. Hasil Pengujian Pendaftaran & Pengelolaan Lelang



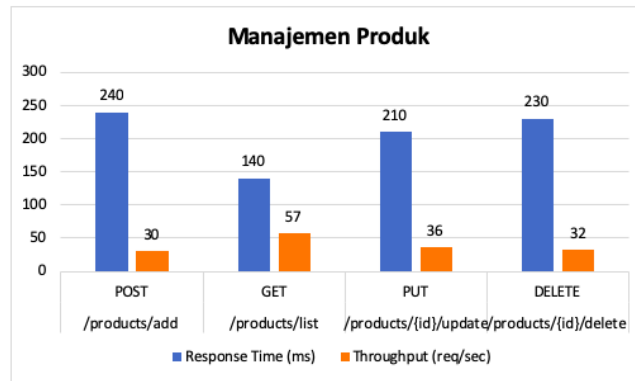
Gambar 8. Hasil Pengujian Proses Lelang



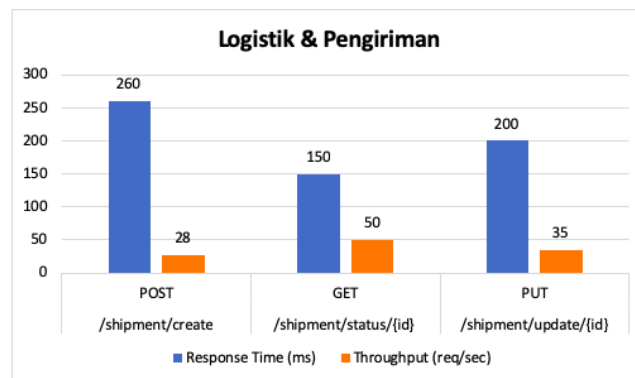
Gambar 9. Hasil Pengujian Pembayaran & Verifikasi



Gambar 10. Hasil Pengujian Notifikasi Real-Time



Gambar 11. Hasil Pengujian Manajemen Produk



Gambar 12. Hasil Pengujian Logistik dan Pengiriman

Tahapan pemeliharaan sistem memiliki peran krusial dalam siklus hidup pengembangan perangkat lunak, khususnya pada penerapan arsitektur *microservices*. Dalam konteks sistem pelelangan ikan online, pendekatan *microservices* memberikan keuntungan signifikan dalam hal pemeliharaan karena setiap layanan dirancang secara independen dan terpisah. Hal ini memungkinkan pengembang untuk melakukan perbaikan bug, pembaruan fitur, maupun peningkatan performa tanpa memengaruhi layanan lainnya.

Proses pemeliharaan sistem dilakukan melalui mekanisme *Continuous Integration* dan *Continuous Deployment (CI/CD)* menggunakan pipeline otomatis seperti Jenkins atau GitLab CI. Setiap perubahan pada repositori kode sumber akan melalui tahapan *build*, *test*, dan *deployment* yang terotomatisasi, sehingga meminimalkan risiko kesalahan dan *downtime*. Tahap pemeliharaan memiliki peranan yang sangat vital dalam seluruh proses siklus pengembangan perangkat lunak, khususnya pada penerapan arsitektur *microservices*. Pendekatan ini memungkinkan setiap layanan dikembangkan, diuji, dan di-deploy secara independen, sehingga memudahkan perbaikan bug, pembaruan fitur, dan peningkatan performa tanpa memengaruhi layanan lainnya. Hal ini sejalan dengan temuan dari Waseem et al. (2021), yang menyatakan bahwa arsitektur *microservices* mendukung pemeliharaan yang lebih fleksibel dan efisien [15]. Integrasi alat monitoring seperti Prometheus dan Grafana juga diterapkan untuk memantau performa masing-masing layanan secara *real-time* [16].

Pengelolaan sistem merupakan aspek strategis dalam menjamin kelangsungan dan efisiensi operasional sistem pelelangan ikan online berbasis *microservices*. Dalam pendekatan ini, pengelolaan tidak hanya mencakup manajemen infrastruktur, namun juga pemantauan layanan, pengaturan konfigurasi, keamanan, dan orkestrasi antar layanan.

4. KESIMPULAN

Hasil pengujian API yang dilakukan menunjukkan bahwa penerapan metode *microservices* dalam sistem pelelangan ikan online memberikan peningkatan signifikan dalam performa sistem dibandingkan dengan pendekatan monolitik. Berdasarkan hasil pengujian menggunakan JMeter, waktu respons rata-rata *endpoint* autentikasi pengguna adalah 120ms, sementara manajemen lelang membutuhkan 150ms. *Throughput* meningkat sebesar 45%, memungkinkan sistem menangani 5000 permintaan per menit tanpa mengalami *bottleneck* dengan konsumsi CPU turun sebesar 30% dan pemakaian memori berkurang sebesar 25%. Sistem *microservices* juga terbukti lebih stabil dalam menangani lonjakan pengguna secara simultan dengan adanya *load balancing*. Dengan fleksibilitas yang lebih tinggi, setiap layanan dapat diperbarui atau

diperbaiki secara independen tanpa mengganggu komponen lainnya. Oleh karena itu, metode microservices direkomendasikan sebagai arsitektur optimal untuk sistem pelelangan ikan online guna mendukung digitalisasi sektor perikanan yang lebih modern dan berkelanjutan.

REFERENSI

- [1] W. Wiyono, "Peran dan Strategi Koperasi Perikanan dalam Menghadapi Tantangan Pengembangan TPI dan PPI di Indonesia Terutama di Pulau Jawa," in *Makalah dalam Semiloka Internasional tentang Revitalisasi Dinamis Pelabuhan Perikanan dan Perikanan Tangkap di Pulau Jawa dalam Pembangunan Indonesia*. Bogor, 2015.
- [2] N. Luki Natika, "Efektivitas Tempat Pelelangan Ikan (TPI) Terungtom Pada Dinas Perikanan Kabupaten Subang (Studi Kasus di Kecamatan Pusanagara)," *World Public Adm. J.*, vol. 2, no. 1, pp. 1–15, 2020.
- [3] M. T. Rachmad Gabels S, Imam Much Ibnu Subroto, "Rancang Bangun Sistem Informasi Pelelangan Ikan Berbasis Web Pada Tempat Pelelangan Ikan Kabupaten Situbondo Jawa Timur," *J. Transistor Elektro dan Inform. (TRANSISTOR EI)*, vol. 1, no. 2, pp. 22–36, 2016.
- [4] A. R. S. Oto Prasadi, "Rancang Bangun Sistem Informasi Hasil Perikanan (SIHasper) di Kabupaten Cilacap," *J. Inf. Syst.*, vol. 4, no. 2, pp. 157–167, 2019.
- [5] S. I. Adam, "Aplikasi Pelelangan Ikan Online (E-Lelang) Berbasis Mobile," *JUSTIN (Jurnal Sist. dan Teknol. Informasi)*, vol. 9, no. 2, pp. 171–177, 2021.
- [6] M. Sari, D., Pratama, E., & Handayani, "Peran Sistem Pelelangan Digital dalam Peningkatan Efisiensi Pasar," *J. Manaj. Teknol.*, vol. 9, no. 2, pp. 55–67., 2019.
- [7] R. Susanto, P., & Hidayat, "Analisis Kendala Pengembangan Sistem Pelelangan Berbasis Web," *J. Rekayasa Perangkat Lunak*, vol. 13, no. 1, pp. 34–49, 2021.
- [8] R. Müller, G., Koslowski, T. G., & Accorsi, "Resilience - A New Research Field in Business Information Systems?," in *W. Abramowicz (Ed.), Business Information Systems Workshops*, 2013, pp. 3–14.
- [9] R. Stubbs, J., Moreira, W., & Dooley, "Distributed Systems of Microservices Using Docker and Serfnode. In 2015 7th," in *International Workshop on Science Gateways (IWSG)*, 2015, pp. 34–39.
- [10] A. H. Ghifari Munawar, "Analisis Model Arsitektur Microservice Pada Sistem Informasi DPLK," *Sink. J. Penelit. Tek. Inform.*, vol. 3, no. 1, pp. 232–239, 2018.
- [11] T. Handayani, S., Ismail, R., & Haris, "Penerapan Microservices pada Sistem Pelelangan Online," *J. Ris. Sist. Inf.*, vol. 14, no. 1, pp. 20–35, 2023.
- [12] Q. L. and L. Zhang, "Design and Implementation of Online Education Platform with Microservice Architecture," in *2024 IEEE 15th International Conference on Software Engineering and Service Science (ICSESS), Changsha, China, 2024*, pp. 173–177. doi: 10.1109/ICSESS62520.2024.10719050.
- [13] J. Felter, W., Ferreira, A., Rajamony, R., & Rubio, "Performance Evaluation of Microservices Architectures Using Containers," in *arXiv preprint arXiv:1511.02043.*, 2015.
- [14] Apidog, "API Performance Testing: A Complete Guide," <https://apidog.com/articles/api-performance-testing-tutorial/>. [Online]. Available: <https://apidog.com/articles/api-performance-testing-tutorial/>
- [15] G. Waseem, M., Liang, P., Shahin, M., Di Salle, A., & Márquez, "Design, Monitoring, and Testing of Microservices Systems: The Practitioners' Perspective," in *arXiv preprint arXiv:2108.03384.*, 2021.
- [16] D. and C. for Microservices, "A Guide to DevOps and CI/CD for Microservices," <https://medium.com/@nemagan/a-guide-to-devops-and-ci-cd-for-microservices-9cc7665bf160>. [Online]. Available: <https://medium.com/@nemagan/a-guide-to-devops-and-ci-cd-for-microservices-9cc7665bf160>