

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Komunikasi antarlayanan memiliki peran penting dalam arsitektur *microservices* karena setiap layanan saling berinteraksi untuk membentuk satu kesatuan sistem yang utuh. Seiring berkembangnya arsitektur berbasis layanan, kebutuhan akan mekanisme komunikasi yang andal dan efisien menjadi semakin penting (Gough dkk., 2022). Efisiensi ini menentukan kinerja keseluruhan sistem, mengingat satu permintaan pengguna biasanya melibatkan beberapa layanan sekaligus (Indrasiri & Kuruppu, 2020). Namun, pemilihan atau implementasi metode komunikasi antarlayanan yang kurang tepat, terutama ketika jumlah layanan dan kompleksitas sistem meningkat, dapat menjadi sumber hambatan performa (Chandra & Farisi, 2025; Satria & Prihandoko, 2024).

Di sisi lain, penelitian dari Arzo dkk. (2022) menunjukkan bahwa penerapan arsitektur *microservices* memberikan keuntungan dari sisi skalabilitas dan fleksibilitas, tetapi juga memperkenalkan tantangan baru pada pengelolaan komunikasi antar layanan. Salah satu tantangan utama adalah kebutuhan sistem untuk melakukan pertukaran data secara cepat dan stabil, yang menjadi faktor penting dalam efektivitas komunikasi antarlayanan (Yanuardi dkk., 2024). Ketidaktepatan dalam memilih arsitektur *Application Programming Interface* (API) dapat berdampak pada peningkatan penggunaan sumber daya, keterlambatan respons, hingga penurunan stabilitas sistem.

Beragamnya pilihan arsitektur API seperti *Representational State Transfer* (REST), GraphQL, dan gRPC juga menimbulkan tantangan tersendiri, karena setiap pendekatan memiliki keunggulan yang berbeda sesuai konteks aplikasi dan lingkungan infrastruktur yang digunakan (Chandra & Farisi, 2025). Perancangan komunikasi yang tidak optimal dapat mengakibatkan keterlambatan proses dan penurunan performa sistem (Satria & Prihandoko, 2024). Temuan ini diperkuat oleh Sanjaya & Murnawan (2024) yang menjelaskan bahwa keterbatasan komunikasi antar komponen menjadi salah satu faktor utama yang menyebabkan penurunan

performa, terutama ketika sistem mengalami peningkatan jumlah pengguna atau beban tinggi.

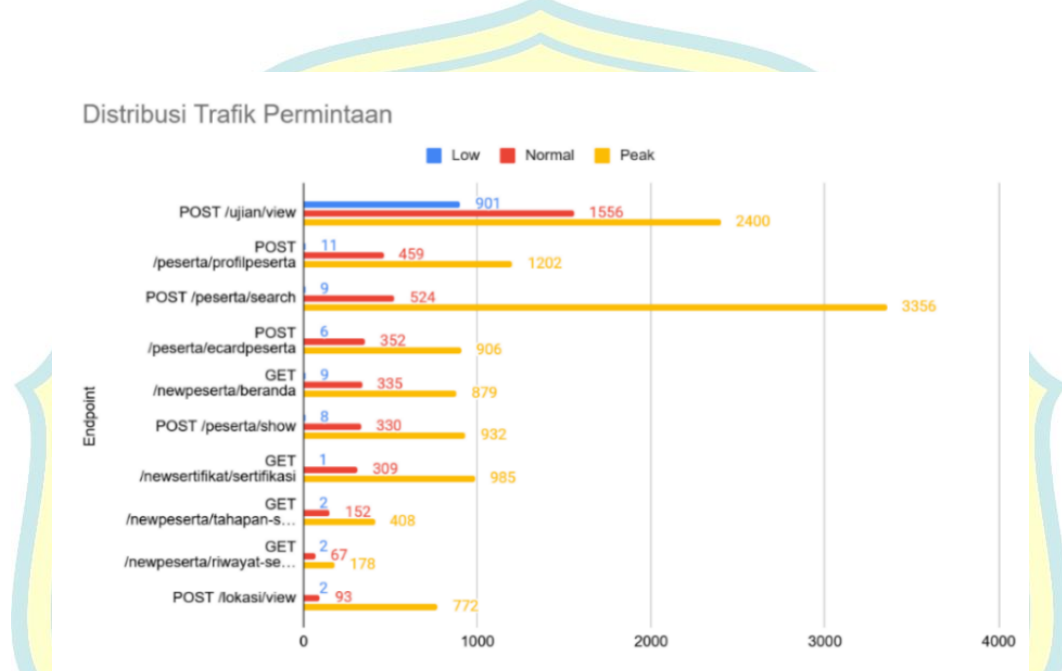
Dalam konteks industri, permasalahan komunikasi antarlayanan dihadapi oleh PT Cyberindo Mega Persada, perusahaan yang bergerak di bidang pengembangan perangkat lunak dan telah mengimplementasikan arsitektur *microservices* dengan REST API untuk komunikasi antarlayanan. Seiring meningkatnya skala sistem dan intensitas interaksi antarlayanan, diperlukan peninjauan terhadap efisiensi mekanisme komunikasi yang digunakan agar performa sistem tetap optimal. Selain itu, berdasarkan hasil wawancara dengan manajer divisi Enterprise System Support (ESS), perusahaan ini memiliki komitmen untuk memastikan bahwa teknologi yang digunakan selalu relevan dengan perkembangan terkini.

Berdasarkan analisis awal terhadap performa salah satu *endpoint* kritikal pada sistem, yaitu */newpeserta/tahapan-sertifikasi* yang digunakan untuk menampilkan status dan tahapan sertifikasi yang sedang diikuti oleh peserta, eksekusi *query database* ditemukan sebagai faktor mayoritas waktu respons, menyumbang 80–97% dari total waktu eksekusi. Komunikasi antar layanan menggunakan protokol REST menyumbang sekitar 10% dari total waktu respons, yang berpotensi berdampak pada akumulasi latensi pada panggilan antar layanan. Analisis performa ini dilakukan dengan mempertimbangkan tiga faktor utama yang umumnya memengaruhi waktu respons pada arsitektur *microservices*, yaitu komunikasi antar layanan, *query database*, dan *internal logic processing* (Indrasiri & Kuruppu, 2020; Bao dkk., 2019 dikutip dalam Zeng dkk., 2022). Hasil analisis ditampilkan pada Tabel 1.1.

Tabel 1. 1. Hasil Analisis Waktu Respons

Faktor Waktu Respons	Rentang Waktu (ms)	Persentase Terhadap Total Latensi
Komunikasi antar layanan	33 – 152	2,5% – 10,7%
<i>Query database</i>	1.263 – 1.451	80,2% – 97,3%
<i>Internal logic processing</i>	0 – 305	0% – 16,9%
Total eksekusi <i>endpoint</i>	1,323 – 1,809	100%

Analisis awal juga dilakukan terhadap data *access log* dari sistem terkait untuk memahami karakteristik beban permintaan pada *endpoint* yang melibatkan proses komunikasi antarlayanan. Data *access log* yang dikumpulkan selama periode pengamatan pada tanggal 29 November hingga 5 Desember 2025 kemudian diolah dan dikelompokkan berdasarkan jumlah permintaan per jam, sehingga diperoleh tiga kategori skenario beban sistem, yaitu *low traffic*, *normal traffic*, dan *high traffic*. Hasil pengolahan data *access log* disajikan pada Gambar 1.1.



Gambar 1. 1. Distribusi Trafik Permintaan pada Sistem *Microservices* yang Dikelola PT Cyberindo Mega Persada

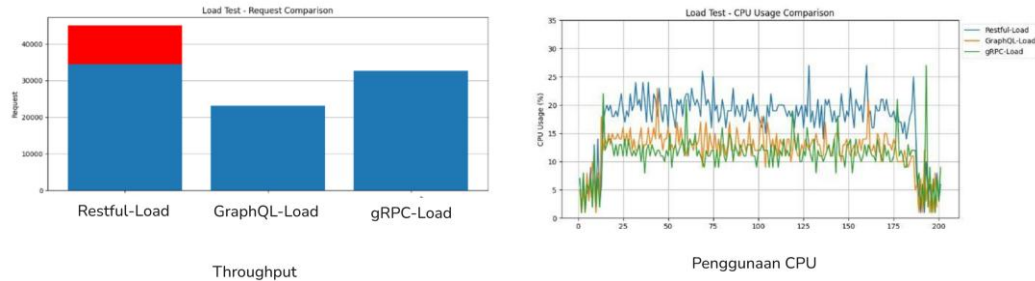
Gambar 1.1 menunjukkan bahwa beban permintaan antar *endpoint* bervariasi, dengan total permintaan per jam tertinggi mencapai 3356 permintaan pada kondisi *high traffic*, sementara *endpoint* pendukung memproses ratusan hingga ribuan permintaan per jam secara kolektif. Peningkatan beban ini berpotensi memengaruhi performa sistem melalui peningkatan waktu respons ataupun risiko penurunan *availability* (Indrasiri & Kuruppu, 2020). Dalam konteks bisnis, hal ini meningkatkan risiko pelanggaran *Service Level Agreement* (SLA) perusahaan sebesar 99,8% per bulan, yang berimplikasi langsung pada kewajiban pemberian kompensasi finansial kepada klien berdasarkan selisih *availability* aktual terhadap SLA dan biaya layanan bulanan. Dengan ketentuan tersebut, bila *availability* sistem

hanya mencapai 95% pada suatu bulan dan biaya layanan klien sebesar Rp5.000.000, maka kompensasi yang harus dibayarkan perusahaan adalah sebesar 4,8% dari biaya layanan, yaitu sebesar Rp240.000.

Selain risiko kompensasi di sisi perusahaan, penurunan *availability* juga berpotensi berdampak langsung pada aktivitas bisnis yang bergantung pada sistem. Pada paruh kedua tahun 2025, sistem memproses sekitar 450–1.200 transaksi per bulan dengan nilai transaksi bulanan berkisar antara Rp1–4 miliar, sehingga apabila *availability* hanya mencapai 95% potensi nilai transaksi yang terdampak berada pada kisaran Rp100–150 juta per bulan. Kondisi tersebut menunjukkan bahwa PT Cyberindo Mega Persada perlu memastikan setiap komponen yang berkontribusi terhadap performa sistem dapat beroperasi secara optimal, termasuk mekanisme komunikasi antarlayanan yang digunakan pada arsitektur *microservices*. Meskipun sebagian besar waktu respons berasal dari *query database*, kontribusi komunikasi antar layanan tetap dapat berdampak pada waktu respons *end-to-end*, terutama pada sistem yang melibatkan banyak panggilan antar layanan (Indrasiri & Kuruppu, 2020). Oleh karena itu, memahami karakteristik mekanisme komunikasi, termasuk perbandingan protokol seperti REST dan gRPC, menjadi penting untuk mengevaluasi performa sistem secara menyeluruh.

Fakta mengenai pentingnya waktu respons sebagai indikator performa layanan menekankan perlunya evaluasi mekanisme komunikasi antar layanan dalam arsitektur *microservices*. Sejumlah penelitian relevan telah menyoroti perbedaan karakteristik performa antara protokol komunikasi yang umum digunakan, seperti REST dan gRPC. Chandra & Farisi (2025) menunjukkan bahwa penggunaan *Central Processing Unit* (CPU) pada gRPC lebih stabil dibandingkan REST seiring peningkatan beban data. Selain itu, ditunjukkan juga bahwa pada skema *load testing*, REST memiliki *throughput* yang lebih tinggi namun memiliki tingkat kegagalan yang tinggi dibandingkan dengan gRPC, sebagaimana ditunjukkan pada Gambar 1.2. Sementara itu, Ain dkk. (2025) menemukan bahwa pada kondisi *low traffic*, REST memiliki *response time* yang lebih rendah pada beberapa metode. Namun, dari sisi efisiensi transfer data, gRPC unggul signifikan, sebagaimana ditunjukkan pada Gambar 1.3. Perbedaan karakteristik ini

menunjukkan perlunya evaluasi empiris terhadap pengaruh protokol komunikasi pada performa sistem yang relevan dengan kondisi yang diteliti.



Sumber: Chandra & Farisi, 2025

Gambar 1. 2. Perbandingan *Throughput* dan Penggunaan CPU antara REST dan gRPC



Sumber: Ain dkk., 2025

Gambar 1. 3. Perbandingan Waktu Respons dan Ukuran Respons antara REST dan gRPC

Berdasarkan temuan-temuan tersebut, penelitian ini berfokus pada pengujian empiris performa komunikasi antarlayanan menggunakan REST API dan gRPC pada sistem *microservices*. Melalui penelitian ini, diharapkan diperoleh hasil empiris yang dapat menjadi acuan bagi PT Cyberindo Mega Persada dalam menentukan strategi pengembangan sistem API di masa mendatang. Dengan demikian, perusahaan dapat mengambil keputusan berbasis data dalam mempertimbangkan adopsi teknologi baru.

1.2. Identifikasi Masalah

Berdasarkan uraian pada latar belakang, dapat diidentifikasi beberapa permasalahan sebagai berikut:

1. Komunikasi antarlayanan pada arsitektur *microservices* dari berbagai penelitian masih menunjukkan permasalahan efisiensi dan latensi ketika pemilihan protokol tidak sesuai, sehingga dapat memengaruhi kinerja sistem secara keseluruhan.
2. Kinerja komunikasi antarlayanan pada sistem *microservices* di PT Cyberindo Mega Persada berpotensi dipengaruhi oleh mekanisme komunikasi yang digunakan, terutama pada aspek efisiensi transfer data, latensi, dan penggunaan sumber daya ketika sistem berada pada kondisi beban tinggi.
3. Pengambilan keputusan terkait adopsi teknologi komunikasi baru belum memiliki dasar empiris yang terukur, sehingga diperlukan evaluasi komparatif antara REST API dan gRPC untuk memastikan peningkatan performa dan efisiensi yang dapat dicapai.

1.3. Pembatasan Masalah

Penelitian ini dibatasi pada hal-hal berikut:

1. Analisis hanya difokuskan pada perbandingan performa komunikasi antarlayanan pada arsitektur *microservices* menggunakan REST API dan gRPC pada Sistem Manajemen Sertifikasi Pembiayaan yang dikembangkan di bawah PT Cyberindo Mega Persada, sebagai dasar perbandingan efisiensi antara REST API dan gRPC.
2. Penelitian hanya menganalisis mekanisme komunikasi antarlayanan berbasis *synchronous request-response* menggunakan REST API dan gRPC. Mekanisme komunikasi *asynchronous* berbasis *message broker*, seperti RabbitMQ, tidak termasuk dalam ruang lingkup penelitian.
3. Penelitian ini difokuskan pada mekanisme komunikasi antarlayanan dengan proses internal aplikasi yang dibuat konsisten pada setiap skenario pengujian, tanpa membahas optimasi logika bisnis, struktur basis data, maupun efisiensi *query*.

4. Penelitian tidak mencakup pembahasan terkait keamanan API, desain arsitektur *microservices* secara menyeluruh, maupun aspek bisnis dan manajerial dalam pengambilan keputusan teknologi.
5. Analisis komunikasi antarlayanan hanya mencakup interaksi antar layanan internal yang dikembangkan dan dikelola oleh perusahaan tempat penelitian dilakukan, dan tidak mencakup komunikasi antara layanan internal dengan layanan eksternal atau vendor pihak ketiga. Apabila sistem membutuhkan dependensi layanan vendor, maka digunakan layanan *dummy* sebagai pengganti.
6. Penelitian ini dibatasi pada sejumlah *endpoint* yang dipilih dengan pendekatan *multistage sampling* dengan mempertimbangkan kriteria volume trafik dan tingkat dependensi antar layanan.
7. Pengukuran efisiensi *payload* dilakukan pada endpoint yang mengirimkan variasi ukuran data dalam komunikasi antarlayanan, tanpa melibatkan pengaruh performa *database*.
8. Pengujian dilakukan pada lingkungan lokal yang disusun dengan spesifikasi serupa server produksi, agar hasil mencerminkan kondisi nyata.
9. Layanan yang dianalisis merupakan layanan yang tercatat dalam data *access log* pada rentang waktu 1 Maret 2026 hingga 7 Maret 2026.

1.4. Rumusan Masalah

Berdasarkan latar belakang dan pembatasan masalah yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah:

1. Apakah terdapat perbedaan performa yang signifikan antara API berbasis REST dan gRPC dalam komunikasi antarlayanan pada arsitektur *microservices* di lingkungan PT Cyberindo Mega Persada?
2. Apakah API berbasis gRPC memiliki performa yang lebih baik dibandingkan API berbasis REST berdasarkan hasil analisis terhadap metrik performa sistem dalam efisiensi komunikasi antarlayanan pada arsitektur *microservices*?

1.5. Tujuan Penelitian

Penelitian ini bertujuan untuk:

1. Mendesain dan mengimplementasikan API berbasis gRPC agar komunikasi antar layanan memiliki performa yang lebih baik dibandingkan API berbasis REST yang telah digunakan di PT Cyberindo Mega Persada.

1.6. Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat sebagai berikut:

1. Memberikan rekomendasi berbasis data bagi PT Cyberindo Mega Persada dalam menentukan strategi pengembangan sistem API yang lebih efisien dan berperforma tinggi.
2. Menjadi acuan praktis bagi tim pengembang dalam mengevaluasi dan memilih teknologi komunikasi antarlayanan yang sesuai dengan kebutuhan sistem *microservices*.
3. Memberikan kontribusi akademik dengan menambah referensi empiris mengenai implementasi dan evaluasi performa gRPC pada arsitektur *microservices* modern.

