

PENGARUH KARAKTERISTIK RESPONS API TERHADAP PERFORMA APLIKASI SELULER REACT NATIVE DALAM ARSITEKTUR *MICROSERVICES*

REYVIDO YOGA DWIMARSHA



**PROGRAM MAGISTER ILMU KOMPUTER
SEKOLAH SAINS DATA, MATEMATIKA, DAN INFORMATIKA
INSTITUT PERTANIAN BOGOR
BOGOR
2026**

@Hak cipta milik IPB University

IPB University



IPB University
— Bogor Indonesia —

- Hak Cipta Dilindungi Undang-undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.

Perpustakaan IPB University



PERNYATAAN MENGENAI TESIS DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA

Dengan ini saya menyatakan bahwa tesis dengan judul “Pengaruh Karakteristik Respons API terhadap Performa Aplikasi Seluler React Native dalam Arsitektur *Microservices*” adalah karya saya dengan arahan dari dosen pembimbing dan belum diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir tesis ini.

Dengan ini saya melimpahkan hak cipta dari karya tulis saya kepada Institut Pertanian Bogor.

Bogor, Juli 2026

Reyvido Yoga Dwimarsha
M0503241043

RINGKASAN

REYVIDO YOGA DWIMARSHA. Pengaruh Karakteristik Respons API terhadap Performa Aplikasi Seluler React Native dalam Arsitektur *Microservices*. Dibimbing oleh SRI WAHJUNI dan HERU SUKOCO.

Performa aplikasi seluler merupakan faktor penting dalam menentukan kualitas pengalaman pengguna, terutama pada aplikasi modern yang bergantung pada komunikasi jaringan dan pemrosesan data *real-time*. Dalam arsitektur *microservices*, respons API secara alami memiliki variasi latensi dan ukuran *payload* akibat pemecahan layanan dan proses agregasi data lintas layanan. React Native, sebagai kerangka kerja lintas platform dengan pemisahan JavaScript *thread* dan UI *thread*, memiliki sensitivitas tinggi terhadap keterlambatan respons dan beban data, karena seluruh proses *parsing* JSON dan logika aplikasi dieksekusi pada JavaScript *thread* yang sama. Meskipun pedoman industri menekankan pentingnya API yang efisien, batas toleransi empiris terhadap latensi dan ukuran *payload* dalam konteks React Native belum tersedia sebagai rujukan yang terukur dan dapat direplikasi. Penelitian ini bertujuan untuk mengkaji secara empiris pengaruh karakteristik respons API, khususnya latensi dan ukuran *payload*, terhadap performa aplikasi React Native pada sisi klien. Variabel latensi dan *payload* direkayasa secara deterministik untuk mengamati dampaknya terhadap lima metrik performa, yaitu waktu pemuatan data, waktu *rendering* UI, penggunaan CPU, penggunaan memori, dan *frame rate*, tanpa melakukan analisis mendalam terhadap mekanisme internal *backend microservices*. Penelitian dibatasi pada kondisi tunak (*steady-state*) dengan profil beban satu pengguna, sehingga pengaruh kedua variabel dapat diisolasi dari faktor konkurensi maupun variabilitas jaringan nyata.

Metode penelitian menggunakan eksperimen kuantitatif dengan rancangan faktorial penuh 4×4 . Latensi diatur pada empat tingkat (50 ms, 150 ms, 300 ms, dan 600 ms), sedangkan ukuran *payload* ditetapkan pada empat tingkat (5 KB, 20 KB, 50 KB, dan 100 KB), menghasilkan 16 kondisi eksperimen yang masing-masing dijalankan sebanyak 20 replikasi sehingga diperoleh 320 observasi. Aplikasi uji React Native dikembangkan pada konfigurasi mode *release* dengan Hermes engine dan diuji pada Xiaomi Redmi Note 14 (RAM 8 GB, MediaTek Helio G99-Ultra, prosesor *octa-core*, resolusi FHD+, *refresh rate* 60 Hz) dalam lingkungan terkontrol; metrik direkam menggunakan instrumen bawaan aplikasi, antarmuka FrameMetrics *native* Android, dan *profiler* berbasis ADB. Analisis data mengikuti alur bertahap yang terdiri atas statistik deskriptif, uji asumsi (normalitas, homogenitas varians, dan independensi residual), Two-Way ANOVA, regresi linear berganda, serta uji lanjut Tukey HSD untuk kontras kondisi terbaik (L1P1) dan terburuk (L4P4).

Hasil penelitian menunjukkan bahwa latensi dan ukuran *payload* berpengaruh signifikan terhadap kelima metrik performa, tetapi bekerja pada aspek yang berbeda. Latensi merupakan faktor dominan pada waktu pemuatan data ($\eta^2 = 0,975$), dengan setiap tambahan 1 ms latensi memperpanjang waktu muat sekitar 0,60 ms. Sebaliknya, ukuran *payload* menjadi determinan terkuat pada empat metrik lainnya, di antaranya waktu *rendering* UI ($\eta^2 = 0,981$) yang meningkat sekitar 0,80 ms, penggunaan memori yang bertambah sekitar 0,91 MB, dan *frame rate* yang



@Hak Cipta Dilindungi Undang-undang

IPB University

menurun sekitar 0,079 FPS untuk setiap tambahan satu kilobit *payload*. Model regresi linear berganda menunjukkan kecocokan yang sangat kuat pada seluruh metrik, dan efek interaksi antara latensi dan *payload* tidak signifikan pada semua metrik, yang mengindikasikan bahwa kedua faktor bekerja secara aditif dan independen.

Uji lanjut Tukey HSD mengonfirmasi bahwa kondisi terburuk (L4P4) berbeda signifikan dari kondisi terbaik (L1P1) pada seluruh metrik ($p < 0,001$): waktu pemuatan data meningkat dari 128,70 ms menjadi 731,88 ms, waktu *rendering* UI dari 24,11 ms menjadi 110,00 ms, penggunaan CPU dari 21,23% menjadi 38,34%, penggunaan memori dari 167,66 MB menjadi 291,48 MB, dan *frame rate* menurun dari 59,19 menjadi 48,38 FPS. Besarnya selisih tersebut menunjukkan bahwa akumulasi latensi tinggi dan *payload* besar menghasilkan degradasi performa yang substansial, bukan sekadar signifikan secara statistik.

Temuan ini menegaskan bahwa latensi dan ukuran *payload* memengaruhi dua fase yang terpisah dalam *pipeline* eksekusi aplikasi: latensi menentukan lamanya fase penantian data, sedangkan *payload* menentukan beban fase pemrosesan dan *rendering* setelah data diterima. Implikasi praktisnya, pengendalian ukuran *payload* melalui paginasi, seleksi atribut, dan kompresi data perlu menjadi prioritas optimasi sisi klien, sementara pengendalian latensi lebih efektif dilakukan pada sisi server dan desain API. Karena tidak terdapat efek interaksi, kedua strategi optimasi tersebut dapat diterapkan secara terpisah dan paralel. Secara keseluruhan, penelitian ini memberikan bukti empiris bahwa perancangan API pada arsitektur *microservices* bukan hanya isu *backend*, melainkan faktor penentu kualitas dan responsivitas aplikasi seluler lintas *platform*.

Kata kunci: aplikasi seluler, latensi, *microservices*, *payload*, React Native

SUMMARY

REYVIDO YOGA DWIMARSHA. The Impact of API Response Characteristics on the Performance of React Native Mobile Applications in Microservices Architecture. Supervised by SRI WAHJUNI, and HERU SUKOCO.

Mobile application performance is an important factor in determining the quality of user experience, especially in modern applications that rely on network communication and real-time data processing. In microservices architecture, API responses naturally vary in latency and payload size due to service decomposition and cross-service data aggregation. React Native, as a cross-platform framework that separates the JavaScript thread from the UI thread, is highly sensitive to response delays and data load, because all JSON parsing and application logic are executed on the same JavaScript thread. Although industry guidelines emphasize the importance of efficient APIs, empirical tolerance thresholds for latency and payload size in the React Native context are not yet available as measurable, replicable references. This study aims to empirically examine the influence of API response characteristics, particularly latency and payload size, on the client-side performance of React Native applications. The latency and payload variables were engineered deterministically to observe their impact on five performance metrics,

namely data load time, UI rendering time, CPU usage, memory usage, and frame rate, without conducting an in-depth analysis of the internal mechanisms of microservices backends. The study was limited to steady-state conditions with a single-user load profile, so that the effects of both variables could be isolated from concurrency factors and real-network variability.

The research method used a quantitative experiment with a full 4×4 factorial design. Latency was set at four levels (50 ms, 150 ms, 300 ms, and 600 ms), while payload size was set at four levels (5 KB, 20 KB, 50 KB, and 100 KB), resulting in 16 experimental conditions, each run with 20 replications for a total of 320 observations. The React Native test application was built in release mode with the Hermes engine and tested on a physical mid-range Android device in a controlled environment; metrics were recorded using in-app instrumentation, the native Android FrameMetrics interface, and an ADB-based profiler. Data analysis followed a staged pipeline consisting of descriptive statistics, assumption testing (normality, homogeneity of variance, and residual independence), Two-Way ANOVA, multiple linear regression, and a Tukey HSD post-hoc contrast between the best (L1P1) and worst (L4P4) conditions.

The results show that latency and payload size significantly affect all five performance metrics, but operate on different aspects. Latency is the dominant factor for data load time ($\eta^2 = 0.975$), with each additional 1 ms of latency extending load time by approximately 0.60 ms. Conversely, payload size is the strongest determinant of the other four metrics, including UI rendering time ($\eta^2 = 0.981$), which increases by approximately 0.80 ms, memory usage, which grows by approximately 0.91 MB, and frame rate, which decreases by approximately 0.079 FPS for each additional kilobyte of payload. The multiple linear regression models show very strong fits across all metrics, and the interaction effect between latency and payload is not significant for any metric, indicating that the two factors operate additively and independently.

The Tukey HSD post-hoc test confirms that the worst condition (L4P4) differs significantly from the best condition (L1P1) on all metrics ($p < 0.001$): data load time increases from 128.70 ms to 731.88 ms, UI rendering time from 24.11 ms to 110.00 ms, CPU usage from 21.23% to 38.34%, memory usage from 167.66 MB to 291.48 MB, and frame rate decreases from 59.19 to 48.38 FPS. The magnitude of these differences shows that the accumulation of high latency and large payloads produces substantial performance degradation, not merely statistical significance.

These findings confirm that latency and payload size affect two separate phases of the application execution pipeline: latency determines the length of the data-waiting phase, while payload determines the processing and rendering load after the data is received. Practically, payload size control through pagination, selective field retrieval, and data compression should be the priority for client-side optimization, while latency control is more effectively addressed on the server side and in API design. Since no interaction effect was found, both optimization strategies can be applied separately and in parallel. Overall, this study provides empirical evidence that API design in microservices architecture is not merely a backend concern, but a decisive factor in the quality and responsiveness of cross-platform mobile applications.



Hak Cipta Dilindungi Undang-undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.

© Hak Cipta milik IPB, tahun 2026
Hak Cipta dilindungi Undang-Undang

Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan atau menyebutkan sumbernya. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik, atau tinjauan suatu masalah, dan pengutipan tersebut tidak merugikan kepentingan IPB.

Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apa pun tanpa izin IPB.

PENGARUH KARAKTERISTIK RESPONS API TERHADAP PERFORMA APLIKASI SELULER REACT NATIVE DALAM ARSITEKTUR MICROSERVICES

REYVIDO YOGA DWIMARSHA

Tesis
sebagai salah satu syarat untuk memperoleh gelar
Magister pada
Program Magister Ilmu Komputer

**PROGRAM MAGISTER ILMU KOMPUTER
SEKOLAH SAINS DATA, MATEMATIKA, DAN INFORMATIKA
INSTITUT PERTANIAN BOGOR
BOGOR
2026**



@Hak cipta milik IPB University

Hak Cipta Dilindungi Undang-undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 - b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.

Tim Penguji pada Ujian Tesis:
Dr. Hendra Rahmawan, S.Kom., M.Kom.



Judul Tesis : Pengaruh Karakteristik Respons API terhadap Performa Aplikasi
Seluler React Native dalam Arsitektur *Microservices*
Nama : Reyvido Yoga Dwimarsha
NIM : M0503241043

Disetujui oleh

Pembimbing 1:
Dr. Ir. Sri Wahjuni, M.T.

Pembimbing 2:
Dr.Eng. Heru Sukoco, S.Si., M.T.

Diketahui oleh

Ketua Program Studi:
Irman Hermadi, S.Kom, M.S., Ph.D.
NIP 197503112006041009

Dekan Sekolah Sains Data, Matematika dan Informatika:
Prof. Dr. Ir. Agus Buono, M.Si., M.Kom
NIP. 196607021993021001

Tanggal Ujian:
11 Juni 2026

Tanggal Lulus:



PRAKATA

Puji dan syukur penulis panjatkan kepada Allah subhanaahu wa ta'ala atas segala karunia-Nya sehingga karya ilmiah ini berhasil diselesaikan. Tema yang dipilih dalam penelitian yang dilaksanakan sejak bulan Januari 2025 sampai bulan Desember 2025 ini ialah aplikasi *mobile*, dengan judul “Pengaruh Karakteristik Respons API terhadap Performa Aplikasi Seluler React Native dalam Arsitektur *Microservices*”.

Keberhasilan penelitian ini tidak lepas dari bimbingan, petunjuk, dan saran dari berbagai pihak. Terima kasih penulis ucapkan kepada:

1. Dr. Ir. Sri Wahjuni, M.T. dan Dr.Eng. Heru Sukoco, S.Si., M.T. yang telah membimbing dan banyak memberi saran dalam menyelesaikan penelitian ini.
2. Seluruh dosen beserta staf Departemen Ilmu Komputer IPB atas ilmu, bantuan, dan kerjasamanya selama perkuliahan.
3. Bapak Nur Cahya Sugiarto, Ibu Dian Laksmi Herawati selaku kedua orang tua penulis, Reynaldo Yoga Pradana selaku kakak penulis, dan keluarga besar yang selalu mendoakan dan mendukung penulis.
4. Teman-teman Program Studi Ilmu Komputer IPB serta semua pihak yang telah banyak membantu yang tidak dapat penulis sebutkan satu persatu.

Semoga karya ilmiah ini bermanfaat bagi pihak yang membutuhkan dan bagi kemajuan ilmu pengetahuan.

Bogor, Juli 2026

Reyvido Yoga Dwimarsha

DAFTAR ISI

DAFTAR TABEL	x
DAFTAR GAMBAR	x
I PENDAHULUAN	12
1.1 Latar Belakang	12
1.2 Rumusan Masalah	1
1.3 Tujuan	1
1.4 Manfaat	2
1.5 Ruang Lingkup	2
1.6 Kontribusi	3
1.7 Hipotesis	4
II TINJAUAN PUSTAKA	5
2.1 React Native sebagai <i>Platform</i> Aplikasi Seluler	5
2.2 Karakteristik Respons API dan Arsitektur <i>Microservices</i>	6
2.3 Performa Aplikasi Seluler	10
2.4 Penelitian Terdahulu dan Posisi Penelitian	12
2.5 Metode Analisis Data Statistik	14
2.6 Kerangka Pemikiran	18
III METODE	19
3.1 Desain Penelitian	19
3.2 Variabel Penelitian	20
3.3 Metrik Kinerja dan Mekanisme Pengukurannya	22
3.4 Lingkungan Penelitian	25
3.5 Tahapan Penelitian	26
3.6 Analisis Data Statistik	29
IV HASIL DAN PEMBAHASAN	34
4.1 Gambaran Umum Hasil Eksperimen	34
4.2 Pengaruh Karakteristik Respons API terhadap Waktu Pemuatan Data	35
4.3 Pengaruh Karakteristik Respons API terhadap Waktu <i>Rendering</i> UI	37
4.4 Pengaruh Karakteristik Respons API terhadap Penggunaan CPU	39
4.5 Pengaruh Karakteristik Respons API terhadap Penggunaan Memori	41
4.6 Pengaruh Karakteristik Respons API terhadap <i>Frame Rate</i>	43
4.7 Sintesis Temuan Penelitian	45
V SIMPULAN DAN SARAN	48
5.1 Simpulan	48
5.2 Saran	48
DAFTAR PUSTAKA	50
RIWAYAT HIDUP	54



DAFTAR TABEL

1	Ringkasan metrik performa aplikasi seluler yang digunakan dalam penelitian.	11
2	Standarisasi lingkungan eksperimen dan faktor yang diisolasi.	20
3	Tingkat Latensi.	21
4	Tingkat <i>Payload</i> .	21
5	Definisi operasional metrik evaluasi kinerja aplikasi.	22
6	Cakupan Metrik Kinerja dan Justifikasi Penyertaan.	23
7	Spesifikasi lingkungan eksperimen.	25
8	Hasil uji asumsi analisis varians kelima metrik.	34
9	Statistik deskriptif waktu pemuatan data dalam ms, disajikan sebagai rata-rata (simpangan baku), $n = 20$ per sel.	35
10	Hasil <i>two-way</i> ANOVA waktu pemuatan data.	36
11	Hasil regresi linear berganda waktu pemuatan data.	37
12	Statistik deskriptif waktu <i>rendering</i> UI dalam ms, disajikan sebagai rata-rata (simpangan baku), $n = 20$ per sel.	38
13	Hasil <i>two-way</i> ANOVA waktu <i>rendering</i> UI.	39
14	Hasil regresi linear berganda waktu <i>rendering</i> UI.	39
15	Statistik deskriptif penggunaan CPU dalam persen, disajikan sebagai rata-rata (simpangan baku), $n = 20$ per sel.	40
16	Hasil <i>two-way</i> ANOVA penggunaan CPU.	40
17	Hasil regresi linear berganda penggunaan CPU.	41
18	Statistik deskriptif penggunaan memori dalam MB (<i>proportional set size</i>), disajikan sebagai rata-rata (simpangan baku), $n = 20$ per sel.	41
19	Hasil <i>two-way</i> ANOVA penggunaan memori.	42
20	Hasil regresi linear berganda penggunaan memori.	43
21	Statistik deskriptif frame rate dalam FPS, disajikan sebagai rata-rata (simpangan baku), $n = 20$ per sel.	43
22	Hasil <i>two-way</i> ANOVA <i>frame rate</i> .	44
23	Hasil regresi linear berganda <i>frame rate</i> .	44
24	Ringkasan hasil <i>two-way</i> ANOVA kelima metrik.	45
25	Ringkasan koefisien regresi dan nilai tukar antarfaktor kelima metrik.	46
26	Kontras kondisi terbaik (L1P1) dan kondisi terburuk (L4P4).	46

DAFTAR GAMBAR

1	Kerangka Pemikiran Penelitian	18
2	Tahapan penelitian	26
3	Arsitektur simulasi eksperimen	27
4	Rata-rata waktu pemuatan data pada setiap kombinasi tingkat latensi dan ukuran <i>payload</i> . Batang galat menunjukkan ± 1 galat baku rata-rata ($n = 20$ per sel).	36



Hak Cipta Dilindungi Undang-undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
b. Pengutipan tidak merugikan kepentingan yang wajar IPB University.
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.

- 5 Rata-rata waktu *rendering* UI pada setiap kombinasi tingkat latensi dan ukuran *payload*. Batang galat menunjukkan ± 1 galat baku rata-rata ($n = 20$ per sel). 38
- 6 Rata-rata penggunaan CPU pada setiap kombinasi tingkat latensi dan ukuran *payload*. Batang galat menunjukkan ± 1 galat baku rata-rata ($n = 20$ per sel). 40
- 7 Rata-rata penggunaan memori pada setiap kombinasi tingkat latensi dan ukuran *payload*. Batang galat menunjukkan ± 1 galat baku rata-rata ($n = 20$ per sel). 42
- 8 Rata-rata *frame rate* pada setiap kombinasi tingkat latensi dan ukuran *payload*. Batang galat menunjukkan ± 1 galat baku rata-rata ($n = 20$ per sel). 44
- 9 Proporsi keragaman (η^2) yang dijelaskan latensi dan ukuran *payload* pada kelima metrik performa, diurutkan menurut nilai tukar antarfaktor. 46