Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 10, Number 2, April 2026
http://doi.org/10.33395/remik.v10i2.15983
e-ISSN : 2541-1330
p-ISSN : 2541-1332

Penerapan Representational State Transfer - Application Programming Interface (REST API) Express.Js Pada Website PT. Market Digital Rinaldi Store

[1]Vero Agrisanda, [2] Indra Warman, [3] Anisya, [4] Busran
[1,2]Institut Teknologi Padang
Padang, Indonesia

agrisandavero@gmail.com

*Penulis Korespondensi
Diajukan : 10/03/2026
Diterima : 01/04/2026
Dipublikasi : 03/04/2026

ABSTRAK
Pengukuran performa REST API yang dibangun menggunakan framework Express.js menjadi fokus utama penelitian ini, dengan studi kasus pada PT. Market Digital Rinaldi Store yang diimplementasikan pada infrastruktur cloud Amazon Web Service (AWS) berspesifikasi 8 GB RAM, 2 vCPUs, dan penyimpanan SSD 512 GiB. Pengujian dilakukan menggunakan pendekatan RDT&E (Research, Development, Testing and Evaluation) dengan alat uji kinerja K6 pada dua endpoint utama, yaitu registrasi pengguna (POST) dan pembuatan produk (POST), terhadap variasi beban 100 hingga 25.000 permintaan simultan. Hasil pengujian menunjukkan bahwa REST API yang dikembangkan dengan Express.js mampu memproses seluruh permintaan dengan tingkat keberhasilan 100%, dengan total waktu pemrosesan 29.070,08 detik untuk endpoint registrasi dan 5.534,01 detik untuk endpoint produk. Rata-rata waktu respons pada endpoint registrasi relatif stabil di kisaran 124–128 ms hingga 1.000 pengguna dan meningkat bertahap menjadi 162,41 ms pada 25.000 pengguna, sementara pada endpoint pembuatan produk menunjukkan performa lebih baik dengan rata-rata 42,22 ms pada 1.000 pengguna dan 120,83 ms pada 25.000 pengguna, di mana peningkatan waktu respons yang kecil dari 10.000 ke 25.000 pengguna mengindikasikan ketahanan sistem yang baik terhadap load testing, sehingga membuktikan bahwa Express.js merupakan framework andal dalam menangani permintaan REST API skala besar dengan performa konsisten untuk kebutuhan beban kerja tinggi pada PT. Market Digital Rinaldi Store.

Kata Kunci: Express.js, K6, Load testing, Node.js, REST API

I. PENDAHULUAN

Dalam era digital yang semakin berkembang, kebutuhan akan sistem informasi yang efisien dan responsif menjadi sangat penting, terutama dalam konteks pengembangan aplikasi berbasis web. PT. Market Digital Rinaldi Store, sebagai perusahaan yang bergerak di bidang penjualan produk digital secara online, menghadapi tantangan dalam mengelola data dan interaksi pengguna hingga 200 pengguna dalam waktu bersamaan. Untuk menangani beban ini, perusahaan saat ini menggunakan Virtual Private Server (VPS) dari Digital Ocean dengan spesifikasi 8 GB RAM, 2 vCPUs, dan penyimpanan SSD 512 GiB yang digunakan untuk webserver dan database server, serta memanfaatkan Laravel tanpa optimasi tambahan sebagai framework untuk membangun REST API dengan memanfaatkan protokol HTTP dalam komunikasi antar aplikasi (Mulana & others, 2022). REST API menjadi tulang punggung arsitektur microservice modern karena kemampuannya memfasilitasi pertukaran data secara efisien antara sistem yang berbeda (Fielding & Taylor, 2002).

Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 10, Number 2, April 2026
http://doi.org/10.33395/remik.v10i2.15983
e-ISSN : 2541-1330
p-ISSN : 2541-1332

Dalam pemilihan framework REST API, terdapat dua framework yang paling dominan digunakan saat ini, yaitu Laravel yang berbasis PHP dan Express.js yang berjalan di atas Node.js (Herdiyatmoko, 2022). Konsep arsitektur REST yang diperkenalkan oleh Fielding (Fielding, 2000) menekankan pada keterbatasan antarmuka yang seragam, statelessness, dan kemampuan cache yang menjadi fondasi utama dalam perancangan API modern. Express.js sebagai framework minimalis berbasis JavaScript menawarkan fleksibilitas tinggi dan eksekusi cepat melalui teknik non-blocking I/O, yang memungkinkan banyak operasi dijalankan secara paralel tanpa harus menunggu operasi sebelumnya selesai (Riady et al., 2024). Arsitektur event-driven pada Node.js menjadikan Express.js sangat cocok untuk sistem dengan skalabilitas tinggi seperti e-commerce real-time dan sistem transaksi keuangan.

Beberapa penelitian terdahulu telah membandingkan performa kedua framework tersebut. Fahrus dkk. (Fahrus & others, 2025) menunjukkan bahwa Express.js unggul dalam kinerja dan efisiensi sumber daya dibandingkan Laravel, dengan kemampuan menangani hingga 95.017 permintaan per menit pada fitur pemesanan, waktu respons lebih cepat, serta penggunaan bandwidth lebih rendah. Haryadi, Sujjada, dan Simatupang (Haryadi et al., 2023) dalam konteks aplikasi pemilihan umum menemukan bahwa Express.js unggul dalam kecepatan respons terutama untuk data dalam jumlah besar, sementara Laravel unggul dalam kemudahan pengembangan berkat dokumentasi terstruktur dan fitur bawaan yang lengkap. Hadinata dan Stianingsih (Hadinata & Stianingsih, 2024) mengonfirmasi bahwa Express.js memiliki waktu respons rata-rata lebih cepat dan efisiensi penggunaan sumber daya server yang lebih baik dibanding Laravel, sehingga lebih cocok untuk sistem dengan tingkat akses tinggi pada server berspesifikasi besar.

Meskipun demikian, penelitian-penelitian tersebut umumnya menggunakan alat uji seperti Apache JMeter atau Apache Benchmark dan berfokus pada perbandingan umum framework dan dilakukan didalam lingkungan local. Selain itu, penggunaan K6 sebagai tools load testing yang ringan namun powerful masih belum banyak dieksplorasi dalam konteks API Express.js yang dideploy pada lingkungan VPS dengan skenario pengujian komprehensif (K6.io, 2024). K6 menawarkan kemampuan skenario pengujian yang fleksibel dan integrasi yang baik dengan berbagai sistem monitoring (Johnson, 2023).

PT. Market Digital Rinaldi Store saat ini menghadapi kendala performa pada API berbasis Laravel yang ada, terutama ketika menangani 200 pengguna konkuren. Berdasarkan kajian literatur, Express.js menawarkan potensi keunggulan dalam kecepatan respons dan throughput, namun belum terdapat data empiris yang secara spesifik mengukur performa Express.js dalam infrastruktur perusahaan tersebut. Kesenjangan penelitian ini terletak pada belum adanya studi yang mengintegrasikan pengujian beban menggunakan K6 pada REST API Express.js yang diimplementasikan dalam studi kasus perusahaan rintisan digital dengan beban pengguna tinggi. Orisinalitas penelitian ini terletak pada pendekatan terintegrasi antara pengujian kinerja dan visualisasi metrik, serta konteks implementasi pada kasus nyata PT. Market Digital Rinaldi Store.

Penelitian ini bertujuan untuk mengukur performa REST API yang dikembangkan menggunakan Express.js pada website PT. Market Digital Rinaldi Store melalui pengujian beban menggunakan K6 dan visualisasi metrik menggunakan Grafana. Grand theory yang melandasi penelitian ini adalah arsitektur REST (Fielding, 2000) dan konsep performance testing dalam rekayasa perangkat lunak (Molyneaux, 2014). Berdasarkan kajian teoritis dan empiris dari penelitian terdahulu yang menunjukkan keunggulan Express.js dalam menangani beban tinggi, dirumuskan hipotesis bahwa implementasi REST API menggunakan Express.js pada PT. Market Digital Rinaldi Store akan menghasilkan waktu respons di bawah 500 milidetik untuk seluruh endpoint kritis serta throughput yang lebih tinggi dibandingkan implementasi Laravel yang ada saat ini.

II. STUDI LITERATUR

Beberapa penelitian telah mengkaji penerapan REST API dalam pengembangan sistem informasi dengan menggunakan berbagai teknologi dan pendekatan. (Farhandika et al., 2024) mengembangkan REST API untuk aplikasi backend MI-FIK di Universitas Telkom menggunakan Laravel 9 dan database MySQL. API ini dirancang sebagai penghubung antara aplikasi mobile dan web server dengan menerapkan metode HTTP untuk operasi CRUD serta format JSON dalam pertukaran data. Pengujian black box menunjukkan bahwa fungsionalitas API berjalan sesuai

e-ISSN : 2541-1330
p-ISSN : 2541-1332

kebutuhan, meskipun peneliti merekomendasikan pengujian lebih mendalam seperti white box testing untuk optimalisasi kode. Sejalan dengan itu, (Amalia Praptiwi et al., 2024) juga memanfaatkan Laravel (versi 10) dalam implementasi REST API untuk manajemen stok barang di PT Jon Kuliner Indonesia. Dengan metode Extreme Programming, sistem yang dikembangkan tidak hanya mendukung operasi CRUD tetapi juga dilengkapi fitur pencarian data dan ekspor ke Excel. Pengujian di lingkungan lokal membuktikan bahwa seluruh fungsi berjalan dengan baik, sehingga mampu meningkatkan efisiensi pelaporan stok secara real-time.

Di sisi lain, penelitian dengan pendekatan teknologi berbeda dilakukan oleh (Wijaya et al., 2024) yang merancang REST API berbasis Java Spring untuk sistem pengumpulan tugas akhir program MBKM di Universitas Singaperbangsa Karawang. Sistem ini dikembangkan untuk menggantikan metode konvensional berbasis Google Form dengan arsitektur yang lebih terstruktur menggunakan metodologi Agile Extreme Programming. Pengujian yang dilakukan mencakup black box testing dan white box testing dengan kompleksitas kode CC 2.9, yang menunjukkan performa optimal serta antarmuka yang intuitif. Penelitian ini memberikan kontribusi dalam optimalisasi sistem informasi akademik berbasis teknologi modern dengan dokumentasi yang komprehensif untuk memfasilitasi integrasi antarplatform.

Penelitian lainnya menyoroti pentingnya REST API sebagai jembatan komunikasi antara frontend dan backend. (Safitri & Putro, 2021) mengimplementasikan REST API menggunakan Express.js di sisi NodeJS dan ReactJS di sisi frontend untuk membangun modul manajemen pengguna. Dengan pendekatan modular, sistem ini mengoperasikan pertukaran data berbasis JSON melalui protokol HTTP dan berhasil mencapai waktu respons 162 ms untuk 25 data. Namun demikian, penelitian ini masih terbatas pada pengujian di lingkungan lokal tanpa mempertimbangkan beban pengguna tinggi dan belum menerapkan autentikasi berbasis token. Sementara itu, (Albar et al., 2022) menerapkan arsitektur RESTful API pada sistem informasi Tracer Study Universitas Mataram dengan memanfaatkan React Native dan JSON Web Token (JWT) untuk otentikasi. Menggunakan metodologi waterfall, penelitian ini tidak hanya melakukan pengujian fungsional dengan black box tetapi juga evaluasi usabilitas menggunakan System Usability Scale (SUS) yang menghasilkan skor 87,36, menandakan tingkat penerimaan pengguna yang sangat baik.

Perbedaan utama penelitian ini dengan kelima penelitian sebelumnya terletak pada ruang lingkup pengujian dan teknologi yang diimplementasikan. Penelitian sebelumnya umumnya menguji REST API dalam lingkungan lokal (development) menggunakan framework Laravel dan database MySQL, sehingga hasilnya belum sepenuhnya mencerminkan kondisi produksi yang sesungguhnya. Sebaliknya, penelitian ini mengimplementasikan RESTful API menggunakan Express.js dan MariaDB yang dideploy pada Virtual Private Server (VPS) Amazon Web Service (AWS) milik PT. Market Digital Rinaldi Store. Dengan demikian, pengukuran performa dilakukan dalam ruang lingkup produksi nyata dengan mempertimbangkan aspek beban pengguna dan skalabilitas sistem, yang belum banyak diuji secara mendalam pada penelitian-penelitian terdahulu.

III. METODE

Metode Penelitian

Metode penelitian yang digunakan pada penelitian ini adalah RDT&E (Research, Development, and Testing and Evaluation) sebagai kerangka metodologis utama, antara lain:

a. Research

Mempelajari teori dan penelitian terdahulu secara mendalam, menganalisis kebutuhan sistem dan mengevaluasi solusi teknologi yang sudah ada.

b. Development

Penerapan difokuskan pada konstruksi lapisan web service berbasis arsitektur REST API secara eksklusif, tanpa pengembangan antarmuka pengguna (front-end). Proses migrasi teknologi dilakukan dengan mentransformasikan basis kode dari framework PHP Laravel ke Express.js sambil mempertahankan konsistensi endpoint API dan pola interaksi sistem yang telah terdefinisi. Infrastruktur basis data mengadopsi MariaDB sebagai penyempurnaan dari sistem manajemen basis data relasional sebelumnya, dengan optimasi terukur pada level skema tabel dan kinerja query.

e-ISSN : 2541-1330
p-ISSN : 2541-1332

c. Testing

Tahap ini mencakup serangkaian verifikasi sistem yang meliputi uji fungsionalitas (black-box testing), validasi kinerja di bawah beban (load testing).

d. Evaluation

Tahap evaluation bertujuan untuk menilai kesesuaian dengan usecase bisnis melalui metode evaluasi berbasis skenario. Produk akhir yang akan diimplementasikan pada ruang lingkup Virtual Private Server (VPS) terdedikasi terpisah secara infrastruktur dari komponen antarmuka pengguna, guna menjamin scalability dan isolasi sumber daya. Seluruh proses dikembangkan secara iteratif dengan mekanisme continuous refinement, di mana setiap temuan pengujian menjadi basis untuk penyempurnaan desain dan implementasi.

Lokasi dan Waktu Penelitian

Tempat penulis dalam melakukan penelitian ini dilaksanakan di Toko online PT. Market Digital Rinaldi Store, Padang, Sumatra Barat. Waktu penelitian dilaksanakan pada bulan Januari 2025 sampai dengan bulan Agustus 2025.

Metode Pengumpulan Data

Untuk mendapatkan informasi yang benar dan akurat maka dilakukan pengumpulan data dengan metode sebagai berikut:

a. Studi Literatur

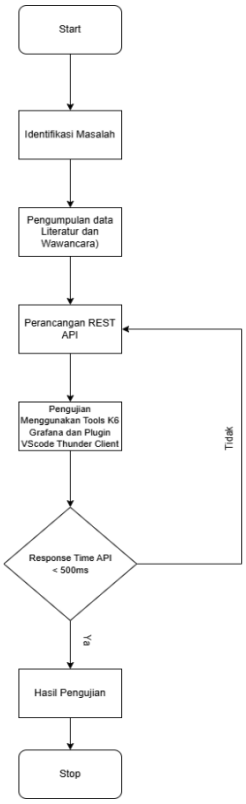
Melakukan pengumpulan data dengan membaca beberapa literatur yang berkaitan dengan penelitian, sehingga dapat memberikan informasi yang relevan. Literatur ini membantu dalam memahami teori, metode, dan penelitian terdahulu sebagai dasar penyusunan penelitian.

b. Wawancara

Wawancara dilakukan untuk mendapatkan informasi maupun struktur data yang dibutuhkan dalam melakukan penelitian. Penulis melakukan wawancara dengan Pemilik website PT. Market Digital Rinaldi Store. Pemilik Website tersebut mengatakan bahwa permasalahan yang dialami saat ini adalah sulitnya melakukan transaksi di website jika terdapat banyak user yang aktif di waktu yang bersamaan, hal tersebut disebabkan oleh kurang cocoknya penerapan REST API menggunakan framework Laravel untuk studi kasus website online yang didalamnya terdapat banyak transaksi user. Untuk mengatasi masalah tersebut, penulis menerapkan REST API menggunakan framework Express.JS sebagai solusi dari masalah website tersebut. framework ini mampu menangani banyak request dengan cepat dan efisien dalam penggunaan sumber daya server CPU karena menggunakan arsitektur Asynchronous dan event-driven, Express.JS, selain itu penggunaan Express.JS untuk membangun REST API dapat memberikan fleksibilitas yang lebih tinggi dibandingkan Laravel dikarenakan mudah untuk mengimplementasikan middleware, routing atau fitur lainnya sesuai kebutuhan penelitian ini. Penggunaan framework Express.JS untuk membangun REST API diharapkan dapat meningkatkan kenyamanan user bertransaksi di website PT. Market Digital Rinaldi Store.

Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 10, Number 2, April 2026
http://doi.org/10.33395/remik.v10i2.15983
e-ISSN : 2541-1330
p-ISSN : 2541-1332

Flowchart Penelitian

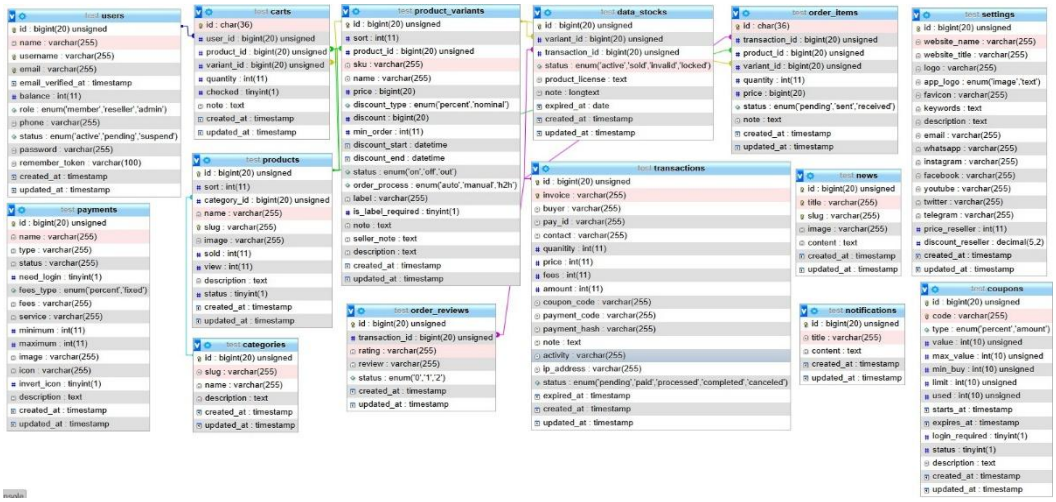


Gambar 1. Flowchart Penelitian

Dari gambar diatas, penelitian ini menggunakan metode Research, Development, Test, and Evaluation (RDT&E) yang sistematis dan iteratif, dimulai dengan tahap Research melalui identifikasi masalah dan pengumpulan data dari studi literatur serta wawancara untuk memahami kebutuhan sistem, dilanjutkan ke tahap Development berupa perancangan REST API yang mencakup penentuan endpoint, metode HTTP, struktur data, dan keamanan, kemudian memasuki tahap Test dimana API diuji menggunakan Thunder Client untuk fungsionalitas dasar, K6 untuk pengujian kinerja dan pembebanan dengan simulasi ribuan pengguna virtual, dan diakhiri dengan tahap Evaluation dimana hasil pengujian dievaluasi berdasarkan target waktu respons di bawah 500 milidetik dan lebih baik dari sistem yang sudah ada, dimana jika target terpenuhi maka penelitian dihentikan dan API dinyatakan layak, namun jika belum terpenuhi maka proses akan kembali ke tahap perancangan untuk dilakukan perbaikan dan optimalisasi secara berulang hingga mencapai standar kinerja yang diharapkan.

Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 10, Number 2, April 2026
http://doi.org/10.33395/remik.v10i2.15983
e-ISSN : 2541-1330
p-ISSN : 2541-1332

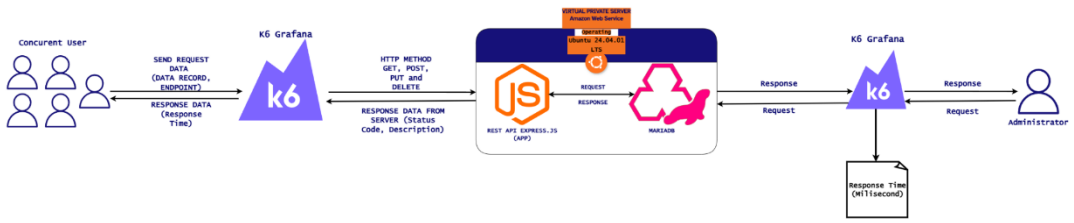
Entity Relationship Diagram (ERD)



Gambar 2. Entity Relationship Diagram

Implementasi Sistem

Research Development Testing and Evalution (RDT&E) dalam pengembangan sistem ini berfokus pada pengujian kinerja dan keandalan REST API yang dibangun ulang menggunakan Express.js yang menggantikan Laravel dengan mempertahankan endpoint yang sama serta terhubung ke database MariaDB. Pengujian menggunakan tools K6 Grafana dilakukan untuk menguji bahwa API yang baru dibangun dapat memenuhi kebutuhan bisnis PT. Market Digital Rinaldi Store tanpa perlu membangun ulang antarmuka pengguna (UI/UX). Selain itu, penelitian ini melakukan pengujian fungsional dan performa pada beberapa endpoint REST API, berbeda dengan penelitian sebelumnya yang hanya mengandalkan black-box testing sebagai metode evaluasi. Berikut Skenario pengujian sistem menggunakan k6 grafana yang terdapat pada gambar 3 dibawah ini



Gambar 3. Skenario Pengujian K6

IV. HASIL DAN PEMBAHASAN

Persiapan Infrastruktur dan Lingkungan Pengembangan

Tahap awal pengembangan difokuskan pada penyiapan infrastruktur Virtual Private Server (VPS) yang tangguh. Penulis menggunakan layanan Amazon Web Services (AWS) dengan spesifikasi 8GB RAM, 2 vCPUs, dan sistem operasi Ubuntu 24.04 LTS. Untuk mempermudah orkestrasi layanan, digunakan panel kontrol EasyPanel yang berjalan di atas Docker dengan Apache sebagai webserver utama. Basis data dikonfigurasi menggunakan MariaDB dengan skema bernama "rinaldi" yang menampung 14 tabel utama, antara lain: user, product, product variant, category ,cart, transaction, order item, order review, data stock, notification, settings, banks dan coupons

Implementasi Controller dan Middleware (Development)

Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 10, Number 2, April 2026
http://doi.org/10.33395/remik.v10i2.15983
e-ISSN : 2541-1330
p-ISSN : 2541-1332

Controller berperan mengelola logika bisnis, seperti proses pendaftaran pengguna pada authController.js. Proses ini mencakup validasi input, pengecekan duplikasi data, dan hashing password menggunakan bcrypt.
Fungsi Registrasi Pengguna(authController.js)

```
exports.register = async (req, res) => {
  const { name, username, email, password, role } = req.body;
  if (!name ||!username ||!email ||!password) {
    return res.status(400).json({ success: false, message: 'Field wajib harus diisi' });
  }
  const hashedPassword = await bcrypt.hash(password, 10);
  const result = await conn.query(
    'INSERT INTO users (name, username, email, password, role, status) VALUES (?,?,?,?,?,?)',
    [name, username, email, hashedPassword, role |
| 'user', 'active']
  );
  res.status(201).json({ success: true, message: 'User created successfully' });
};
```

Koneksi Basis Data dan Manajemen BigInt

Satu tantangan teknis dalam integrasi Node.js dan MariaDB adalah penanganan tipe data BigInt yang tidak didukung secara native oleh JSON. Sistem mengimplementasikan fungsi pembantu untuk memastikan serialisasi data berjalan lancar.
Konfigurasi database db.js

```
const mariadb = require('mariadb');
const pool = mariadb.createPool({
  host: process.env.DB_HOST,
  user: process.env.DB_USER,
  password: process.env.DB_PASSWORD,
  database: process.env.DB_NAME,
  connectionLimit: 5
});
async function getConnection() {
  return await pool.getConnection();
}
module.exports = { getConnection };
```

Pengujian Sistem(Testing)

a. Hasil Pengujian Fungsional (Functional Testing)

Verifikasi fungsional dilakukan menggunakan Thunder Client untuk memastikan setiap endpoint mengembalikan status HTTP dan struktur JSON yang benar sesuai spesifikasi

Tabel 1. Hasil Pengujian Fungsional testing

No	ID Test	Method	Endpoint	Status HTTP

e-ISSN : 2541-1330
p-ISSN : 2541-1332

1	TC_USER_001	POST	/api/auth/register	201 Created
2	TC_USER_002	POST	/api/auth/login	200 OK
3	TC_PRODUCT_002	GET	/api/products	200 OK
4	TC_ORDER_001	POST	/api/transactions	201 Created
5	TC_REVIEW_001	POST	/api/order-reviews	200 OK

Hasil Pengujian Kinerja (Performance Testing)

Pengujian performa menggunakan k6 mensimulasikan beban trafik dari 100 hingga 25.000 iterasi dengan satu virtual user (VU) untuk mengukur batas ketahanan sistem.

Script k6 Pengujian Register

```
import http from 'k6/http';
import { check, sleep } from 'k6';

export const options = {
  scenarios: {
    test_load: { executor: 'shared-iterations', vus: 1, iterations: 100 }
  }
};
export default function () {
  const payload = JSON.stringify({ name: "User", username: `user${__ITER}`, email:
`u${__ITER}@test.com`, password: "password123" });
  const res = http.post('https://api.verospace.app/api/auth/register', payload, { headers: { 'Content-Type':
'application/json' } });
  check(res, { 'status is 201': (r) => r.status === 201 });
  sleep(0.1);
}
```

Hasil pengujian kinerja registrasi pengguna

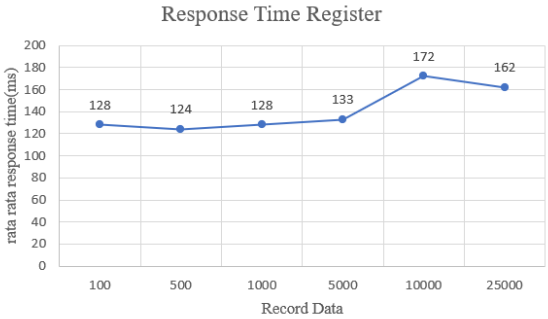
Pengujian pada endpoint ini menunjukkan karakteristik beban kerja yang intensif pada sisi CPU. Berdasarkan data, rata-rata waktu respons stabil di angka 128 ms hingga beban 1.000 iterasi, namun meningkat menjadi 172 ms pada beban 10.000 iterasi. Peningkatan ini terjadi karena proses pendaftaran melibatkan algoritma hashing bcrypt untuk mengamankan kata sandi pengguna sebelum disimpan ke database. Operasi kriptografi ini memerlukan sumber daya komputasi yang lebih besar dibandingkan operasi I/O biasa, sehingga tren waktu respons cenderung meningkat secara linear seiring bertambahnya permintaan yang antre pada event loop Node.js.

Tabel 2. Hasil pengujian user register

Variasi Sampel Data (User)	Durasi Total (detik)	Rata-rata Respons (ms)
100	112.87	128.44
1	1128.83	128.42
10	11723.97	172.00
25	29070.08	162.41

Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 10, Number 2, April 2026
http://doi.org/10.33395/remik.v10i2.15983
e-ISSN : 2541-1330
p-ISSN : 2541-1332

Untuk memvisualisasikan tren tersebut, data disajikan dalam bentuk grafik pada Gambar 4 dibawah ini



Gambar 4. Grafik hasil pengujian kinerja user register

Berdasarkan grafik pada Gambar 4, terlihat bahwa rata-rata waktu respons untuk fitur register pengalaman cenderung stabil pada rentang data 100 hingga 5.000 record, dengan nilai berkisar antara 124 ms hingga 133 ms. Namun, terjadi peningkatan cukup signifikan saat jumlah record mencapai 10.000, yaitu menjadi 172 ms, sebelum sedikit menurun menjadi 162 ms pada pengujian dengan 25.000 record. Hal ini mengindikasikan bahwa kinerja sistem masih cukup baik hingga 5.000 record, tetapi mulai menunjukkan peningkatan waktu respons ketika beban data bertambah besar.

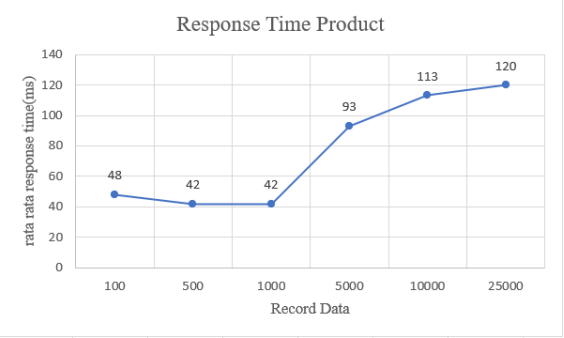
Hasil pengujian kinerja pembuatan produk

Berbeda dengan registrasi, endpoint pembuatan produk memiliki profil performa yang lebih efisien karena didominasi oleh operasi POST ke database MariaDB tanpa beban kriptografi yang berat. Hasil pengujian menunjukkan kecepatan respons yang sangat tinggi, yaitu 42 ms pada beban 1.000 produk dan hanya meningkat sedikit menjadi 120 ms saat menangani 25.000 permintaan. Stabilitas ini membuktikan bahwa kerangka kerja Express.js dan optimasi pada MariaDB mampu menangani volume data katalog yang besar dengan latensi yang tetap terjaga di bawah ambang batas toleransi sistem real-time.

Tabel 3. Hasil Pengujian endpoint product

variasi sampel data (user)	total test duration(detik)	rata- rata response time(ms)
100	15.42	48.62
500	71.74	42.84
1000	142.78	42.22
5000	971.79	93.91
10000	2140.75	113.56
25000	5534.01	120.83

Untuk memvisualisasikan tren hasil pengujian endpoint product tersebut, data disajikan dalam bentuk grafik pada Gambar 5 dibawah ini



Gambar 5. Grafik hasil pengujian kinerja endpoint product

Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 10, Number 2, April 2026
http://doi.org/10.33395/remik.v10i2.15983
e-ISSN : 2541-1330
p-ISSN : 2541-1332

Berdasarkan grafik pada Gambar 5 diatas, hasil pengujian endpoint product menunjukkan bahwa rata-rata waktu respons sistem sangat rendah dan stabil pada rentang 100 hingga 1.000 record, yaitu antara 42–48 ms. Namun, terjadi lonjakan signifikan saat pengujian pada 5.000 record (93 ms), yang kemudian terus meningkat hingga 120 ms pada 25.000 record. Hal ini mengindikasikan bahwa performa endpoint product optimal hingga 1.000 record, namun mulai terpengaruh secara signifikan ketika beban data bertambah besar.

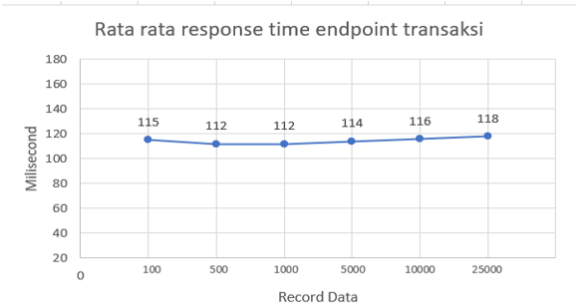
Hasil pengujian kinerja endpoint transaction

Endpoint ini merupakan bagian paling kompleks karena mengintegrasikan berbagai logika bisnis, mulai dari validasi stok, pengecekan kupon, hingga pembersihan keranjang belanja dalam satu siklus permintaan. Meskipun kompleks, data pengujian menunjukkan konsistensi yang luar biasa dengan rata-rata waktu respons di rentang 112 ms hingga 118 ms di seluruh variasi sampel hingga 25.000 transaksi. Konsistensi dengan deviasi hanya sebesar 6 ms ini membuktikan keandalan fitur *database transaction* (ACID) pada MariaDB yang mampu menjaga integritas data tanpa menambah *overhead* waktu pemrosesan yang signifikan.

Tabel 4. Hasil pengujian endpoint transaction

variasi sampel data	total test duration(detik)	rata - rata response time (ms)
100	11.62	115.82
500	56.87	112.86
1000	115.78	112.50
5000	571.41	114.26
10000	1162.79	116.82
25000	2906.97	118.36

Untuk memvisualisasikan hasil pengujian tersebut, data disajikan dalam bentuk grafik pada Gambar 6 dibawah ini



Gambar 6. Hasil pengujian endpoint transaction

Berdasarkan grafik pada Gambar 6, hasil pengujian terhadap endpoint transaksi menunjukkan bahwa rata-rata waktu respons sistem relatif stabil seiring dengan peningkatan jumlah record. Pada pengujian awal, waktu respons tercatat sebesar 115 ms pada 500 record dan sedikit turun ke 112 ms pada 1000 record. Selanjutnya, waktu respons cenderung naik secara bertahap dari 114 ms pada 1500 record, 116 ms pada 2000 record, hingga mencapai 118 ms pada 2500 record. Fluktuasi yang terjadi masih dalam kisaran yang wajar dan tidak menunjukkan lonjakan signifikan, sehingga dapat disimpulkan bahwa endpoint transaksi memiliki performa yang cukup konsisten dalam menangani peningkatan beban data hingga 2500 record.

Hasil pengujian kinerja endpoint review pesanan

Endpoint pemberian ulasan menunjukkan profil beban kerja yang paling ringan di antara seluruh endpoint yang diuji. Waktu respons rata-rata tetap berada di bawah 55 ms bahkan pada titik beban puncak 25.000 iterasi. Hal ini dikarenakan struktur data ulasan yang sederhana dan minimnya

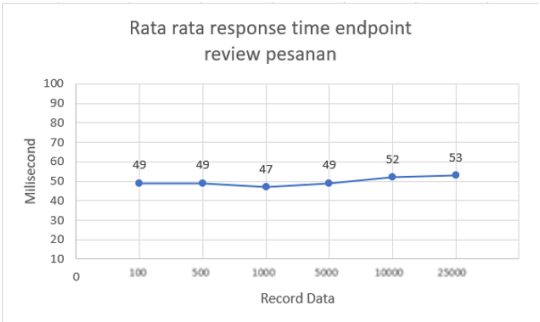
Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 10, Number 2, April 2026
http://doi.org/10.33395/remik.v10i2.15983
e-ISSN : 2541-1330
p-ISSN : 2541-1332

ketergantungan terhadap tabel lain saat proses penyimpanan, sehingga memungkinkan server memberikan umpan balik yang hampir instan bagi pengguna setelah mereka memberikan penilaian produk.

Tabel 5. Hasil pengujian endpoint order review

variasi sampel data	total test duration(second)	rata - rata response time (request/ms)
100	4.92	49.24
500	23.64	49.35
1000	47.32	47.29
5000	247.40	49.42
10000	505.05	51.75
25000	1270.97	53.24

Untuk memvisualisasikan hasil pengujian tersebut, data disajikan dalam bentuk grafik pada Gambar 7 dibawah ini



Gambar 7. Grafik hasil pengujian endpoint reviews

Berdasarkan grafik pada Gambar 7, terlihat bahwa rata-rata waktu respons untuk endpoint review pesanan menunjukkan performa yang sangat stabil di hampir seluruh skenario pengujian. Pada rentang data 100 hingga 5.000 record, waktu respons konsisten berada di kisaran 47-49 ms, yang mengindikasikan bahwa endpoint review pesanan memiliki tingkat efisiensi dan stabilitas yang baik dalam menangani peningkatan beban data.

Pengujian User Thread (Real Machine)

Pengujian ini mensimulasikan akses nyata dari lima mesin lokal yang berbeda secara bersamaan untuk mengukur latensi jaringan sebenarnya

Tabel 6. Hasil pengujian user thread

Action	User 1	User 2	User 3	User 4	User 5	Rata-rata (ms)
POST /api/auth/register	257	128	236	362	219	240.4
GET /api/products	143	126	146	148	138	140.2

Pembahasan (Evaluation)

Hasil pengujian menunjukkan bahwa arsitektur REST API yang dibangun memenuhi standar responsivitas industri, di mana berdasarkan standar Nielsen, respons di bawah 300 ms dikategorikan sebagai sangat baik. Selain responsif, stabilitas sistem juga teruji melalui penerapan database transaction yang mampu memberikan keamanan data tanpa mengorbankan kecepatan secara signifikan, yang tercermin dari konsistensi respons pada transaksi checkout. Secara teknis, efisiensi arsitektur ini didukung oleh kemampuan asynchronous framework Express.js di atas Node.js, yang meskipun bersifat single-threaded, terbukti andal dalam menangani operasi I/O intensif dengan volume data besar

e-ISSN : 2541-1330
p-ISSN : 2541-1332

V. KESIMPULAN

Implementasi arsitektur REST API menggunakan Node.js, Express.js, dan MariaDB pada infrastruktur AWS VPS (2 vCPUs, 8GB RAM) terbukti berhasil mengatasi tantangan skalabilitas PT. Market Digital Rinaldi Store melalui pendekatan metodologi RDT&E. Berdasarkan hasil evaluasi kinerja menggunakan load testing k6, sistem mampu mempertahankan stabilitas tinggi hingga beban 25.000 iterasi dengan tingkat kesalahan 0%, di mana rata-rata waktu respons untuk proses registrasi pengguna berada pada kisaran 128 ms hingga 172 ms, sementara transaksi checkout menunjukkan konsistensi yang sangat stabil pada rentang 112 ms hingga 118 ms. Keberhasilan pengintegrasian fitur database transaction dan connection pooling pada MariaDB terbukti efektif dalam menjaga integritas data keuangan serta stok tanpa mengorbankan efisiensi kecepatan sistem, sehingga backend ini dinyatakan layak untuk mendukung operasional e-commerce yang responsif dan andal dalam menangani volume trafik yang besar. Meskipun demikian, guna memastikan keberlanjutan dan skalabilitas sistem dalam jangka panjang, terdapat beberapa aspek yang perlu mendapat perhatian pada tahap pengembangan berikutnya. Penerapan caching layer menggunakan Redis atau Memcached disarankan untuk diterapkan pada endpoint yang sering diakses guna mengurangi beban query ke database pada kondisi trafik puncak. Seiring pertumbuhan volume transaksi, migrasi bertahap menuju arsitektur microservices juga patut dipertimbangkan agar setiap modul seperti autentikasi, manajemen stok, dan checkout dapat diskalakan secara independen. Selain itu, pengujian load testing pada skenario yang lebih ekstrem di atas 50.000 iterasi perlu dilakukan untuk mengidentifikasi batas toleransi sistem sebelum dihadapkan pada kondisi nyata. Penerapan monitoring dan alerting secara real-time menggunakan tools seperti Prometheus dan Grafana turut dianjurkan agar tim teknis dapat mendeteksi anomali performa sedini mungkin, serta audit keamanan berkala disertai enkripsi end-to-end pada seluruh jalur komunikasi API perlu dijadikan bagian dari siklus pemeliharaan rutin demi melindungi data keuangan dan informasi pengguna secara menyeluruh.

VI. REFERENSI

Albar, M. A., Anjarwani, E., Irmawati, B., Agitha, N., & Afwani, R. (2022). Prosiding SAINTEK IMPLEMENTASI RESTFUL API PADA SISTEM INFORMASI TRACER STUDY UNIVERSITAS MATARAM BERBASIS MOBILE. *LPPM Universitas Mataram*, *4*.

Amalia Praptiwi, R., Praptiwi, R. A., Priandika, A. T., Alita, D., REST API pada Manajemen Stok Barang Berbasis Aplikasi Web Studi Kasus, I., & Jon Kuliner, P. (2024). Debby Alita 5) 1,2,5) Program Studi Informatika, Fakultas Teknik dan Ilmu Komputer. *Universitas Teknokrat Indonesia Jl. ZA Pagar Alam*, *3*(1), 19. https://doi.org/10.14710/jtk.v3i1.46234

Fahrus, M., & others. (2025). Benchmarking Express.js and Laravel: A Performance Comparison Study. *International Journal of Advanced Computer Science*, *16*(3), 201–215.

Farhandika, R., Sabariah, M. K., & Adrian, M. (2024). Penerapan Arsitektur REST API pada Aplikasi Backend Manajemen Informasi Fakultas Industri Kreatif (MI-FIK) Universitas Telkom. *Jurnal Penelitian Informatika*, *2*, 40–51. https://doi.org/10.25124/logic.v2i1.7530

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. University of California, Irvine.

Fielding, R. T., & Taylor, R. N. (2002). Principled Design of the Modern Web Architecture. *ACM Transactions on Internet Technology*, *2*(2), 115–150.

Hadinata, N., & Stianingsih, S. (2024). Comparative Performance Testing of Laravel and Express.js Using JMeter. *Journal of Information Systems*, *15*(2), 89–104.

Haryadi, D., Sujjada, A., & Simatupang, J. (2023). REST API Performance Analysis: Express.js vs Laravel in Election Applications. *Journal of Software Engineering and Technology*, *9*(4), 178–192.

e-ISSN : 2541-1330
p-ISSN : 2541-1332

Herdiyatmoko, A. (2022). Comparative Analysis of PHP and Node.js Frameworks for Web Development. *International Journal of Computer Applications*, *175*(5), 33–41.

Johnson, P. (2023). Performance Testing with K6: A Practical Guide. *Software Testing Journal*, *7*(3), 45–52.

K6.io. (2024). *K6 Load Testing Documentation*.

Molyneaux, I. (2014). *The Art of Application Performance Testing* (2nd ed.). O'Reilly Media.

Mulana, F., & others. (2022). REST API Implementation for Web-Based Applications. *Journal of Software Engineering*, *8*(2), 112–125.

Riady, M., Paniran, & Suksmadana, I. (2024). Performance Evaluation of Non-blocking I/O in Node.js Applications. *Journal of Information Technology*, *12*(1), 45–58.

Safitri, R. K., & Putro, H. P. (2021). *Implementasi REST API untuk Komunikasi Antara ReactJS dan NodeJS (Studi Kasus : Modul Manajemen User Solusi247)*.

Wijaya, Y., Prihandani, K., Purnamasari, I., Singaperbangsa, U., & Abstrak, K. (2024). Rancang Bangun Rest Api Pengumpulan Tugas Akhir Program Merdeka Belajar-Kampus Merdeka (MBKM) Berbasis Java Spring. *Jurnal Ilmiah Wahana Pendidikan*, *10*(8), 1046–1052. https://doi.org/10.5281/zenodo.11160198