

RESEARCH ARTICLE

Pengembangan Aplikasi Web Full-Stack untuk Manajemen Sampah: Implementasi *Frontend*, *Backend*, *Cloud Deployment*

Muhammad Irfan Al Rasyid, Meta Kalista* and Casi Setianingsih

Fakultas Teknik Elektro, Universitas Telkom, Bandung, 40257, Jawa Barat, Indonesia

*Corresponding author: metakalista@student.telkomuniversity

Abstrak

Indonesia menghadapi tantangan besar dalam pengelolaan sampah, termasuk pertumbuhan populasi yang cepat dan infrastruktur yang tidak memadai. Bank Sampah Bersinar, sebagai salah satu inisiatif pengelolaan sampah, menghadapi kesulitan dalam manajemen data pelanggan, transparansi harga sampah, dan proyeksi harga sampah di masa depan. Penelitian ini berfokus pada pengembangan dan peningkatan website Bank Sampah Bersinar melalui pendekatan full-stack. Proses pengembangan mencakup desain antarmuka pengguna (*frontend*), pengembangan *backend*, integrasi prediksi harga sampah menggunakan algoritma *machine learning*, serta *deployment* ke *cloud*. Pengujian aplikasi dilakukan untuk memastikan fungsionalitas dan kinerja yang optimal. Hasil pengujian menunjukkan bahwa peningkatan pada website ini berhasil meningkatkan efisiensi dan kemudahan penggunaan bagi para pengguna.

Key words: Pengembangan Web, Bank Sampah, *FullStack Development*, *Cloud Deployment*, Prediksi Harga Sampah.

Pendahuluan

Indonesia, sebagai negara dengan populasi yang terus bertambah, menghadapi tantangan signifikan dalam pengelolaan sampah[1]. Salah satu upaya untuk mengatasi tantangan ini adalah melalui Bank Sampah Bersinar, yang berfungsi sebagai wadah bagi masyarakat untuk mengelola sampah secara lebih terstruktur dan berkelanjutan. Namun, dengan semakin kompleksnya kebutuhan dan permasalahan dalam manajemen sampah, website Bank Sampah Bersinar menghadapi beberapa kendala, antara lain manajemen data pelanggan yang kurang efisien, keterbatasan transparansi harga sampah, serta fluktuasi harga sampah yang sulit diprediksi secara akurat. Untuk menjawab tantangan tersebut, penelitian ini berfokus pada pengembangan website Bank Sampah Bersinar melalui pendekatan *full-stack development*. Pengembangan ini mencakup perancangan dan implementasi antarmuka pengguna yang intuitif (*frontend*), pengembangan *backend* yang handal untuk manajemen data, serta integrasi algoritma *machine learning* guna memprediksi harga sampah yang dinamis. Selain itu, *deployment* ke *cloud* dilakukan untuk memastikan aksesibilitas website bagi pengguna secara optimal. Melalui pengembangan ini, diharapkan website Bank Sampah Bersinar mampu memberikan solusi yang lebih efisien dan mudah digunakan bagi masyarakat dalam mengelola sampah, serta mendukung transparansi dan akurasi prediksi harga sampah.

Tinjauan Pustaka

Full-Stack Development

Full-stack development adalah pendekatan dalam pengembangan aplikasi web yang mencakup pengembangan bagian *frontend* dan *backend* secara menyeluruh. Pengembang *full-stack* harus memiliki pemahaman yang baik tentang teknologi dan alat-alat yang digunakan di kedua sisi ini, seperti HTML, CSS, dan JavaScript untuk *frontend*, serta bahasa pemrograman seperti *Python*, *Java*, atau *Node.js* untuk *backend*. Kemampuan untuk mengelola seluruh siklus pengembangan aplikasi dari antarmuka pengguna hingga pengelolaan data dan logika bisnis memberikan fleksibilitas yang tinggi dalam merancang dan mengimplementasikan solusi yang efisien dan efektif.

Selain itu, *full-stack development* juga melibatkan integrasi antara berbagai komponen teknologi untuk menciptakan pengalaman pengguna yang mulus dan konsisten. Dengan memahami kedua sisi pengembangan ini, seorang pengembang dapat memastikan bahwa setiap elemen dalam aplikasi saling berintegrasi dengan baik, dari interaksi pengguna di antarmuka hingga pemrosesan data di server. Pendekatan ini juga memungkinkan pengembang untuk lebih cepat dalam iterasi dan perbaikan aplikasi, karena mereka memiliki kendali penuh atas seluruh aspek teknis dari proyek tersebut [2].

Antarmuka Pengguna

Antarmuka pengguna (*frontend development*) merupakan aspek krusial dalam pengembangan aplikasi web, karena ini adalah bagian yang berinteraksi langsung dengan pengguna. Desain yang responsif dan intuitif tidak hanya membuat aplikasi mudah digunakan, tetapi juga

meningkatkan kepuasan pengguna secara keseluruhan. Prinsip-prinsip *User Experience* (UX) dan *User Interface* (UI) sangat penting dalam *frontend development*. UX menitikberatkan pada bagaimana pengguna merasakan dan berinteraksi dengan aplikasi, sementara UI lebih fokus pada tata letak visual dan elemen-elemen grafis yang membentuk pengalaman pengguna.

Selain itu, penerapan ini dalam *frontend development* sangat penting untuk memastikan bahwa antarmuka pengguna dapat beradaptasi dengan berbagai perangkat, mulai dari ponsel hingga layar desktop yang lebih besar. Ini melibatkan penggunaan teknik seperti media queries dan grid yang fleksibel untuk menyesuaikan tata letak antarmuka berdasarkan ukuran layar. Penerapan konsep *Accessibility* juga harus diperhatikan untuk memastikan bahwa aplikasi dapat diakses oleh semua pengguna, termasuk mereka dengan keterbatasan fisik atau sensorik, sehingga menciptakan pengalaman yang inklusif bagi semua kalangan [3].

Pengembangan Backend

Backend development adalah inti dari fungsionalitas sebuah aplikasi web, di mana logika bisnis, pengelolaan data, dan komunikasi dengan basis data terjadi. Pada *backend*, pengembang menggunakan bahasa pemrograman server-side seperti *Python*, *Ruby*, atau *Java* untuk menulis logika aplikasi yang mengelola bagaimana data diproses dan disimpan. Pengelolaan basis data juga merupakan bagian penting dari *backend development*, dengan sistem manajemen basis data seperti *MySQL*, *PostgreSQL*, atau *MongoDB* yang digunakan untuk menyimpan, mengambil, dan memanipulasi data secara efisien.

Selain itu, *backend development* mencakup pembuatan API (*Application Programming Interface*), yang memungkinkan komunikasi antara *frontend* dan *backend*. Salah satu pendekatan yang populer adalah *RESTful API*, yang memungkinkan pertukaran data dengan cara yang terstruktur dan efisien. Dengan *RESTful API*, data dapat dikirim dan diterima melalui protokol *HTTP*, memungkinkan *frontend* untuk melakukan operasi seperti mengambil data dari server atau mengirim data ke server untuk disimpan. Pengaturan server juga merupakan aspek penting dalam *backend development*, di mana server harus dikonfigurasi untuk mendukung skalabilitas, keamanan, dan kinerja yang optimal [3].

Cloud Deployment

Cloud deployment merujuk pada proses penyebaran aplikasi web di lingkungan *cloud* agar dapat diakses secara luas oleh pengguna melalui internet. Teori yang relevan termasuk konsep *Scalability*, *Elasticity*, dan *High Availability*. *Scalability* mengacu pada kemampuan sistem untuk menangani peningkatan beban kerja dengan memperbesar sumber daya yang tersedia, sementara *Elasticity* mengacu pada kemampuan sistem untuk menyesuaikan sumber daya dengan beban kerja yang berubah-ubah. *High Availability* adalah konsep yang memastikan bahwa aplikasi tetap dapat diakses oleh pengguna meskipun terjadi kegagalan pada salah satu komponen sistem [4].

Metodologi Penelitian

Desain Rancangan Solusi

Tujuan dari sub bab Desain Rancangan Solusi adalah untuk mengevaluasi dan membandingkan berbagai alternatif teknologi dan solusi yang dapat digunakan dalam pengembangan proyek tugas akhir, guna menentukan opsi terbaik yang memenuhi kebutuhan dan tujuan proyek. Sub bab ini bertujuan untuk membandingkan *platform* pengembangan, yaitu antara website dan aplikasi *mobile*, untuk menentukan media yang paling sesuai untuk mencapai tujuan fungsional dan pengalaman pengguna yang diinginkan. Selain itu, akan dilakukan perbandingan antara berbagai *framework backend* dan *frontend* untuk memilih yang

paling efektif dan efisien dalam mendukung pengembangan aplikasi. Sub bab ini juga akan mengevaluasi opsi server, baik *on-premise* maupun *cloud*, untuk menentukan solusi yang paling cocok dari segi biaya, skalabilitas, dan manajemen. Dengan menganalisis dan membandingkan berbagai solusi ini, diharapkan dapat diidentifikasi pendekatan terbaik yang akan diadopsi dalam rancangan solusi, memastikan bahwa sistem yang dikembangkan dapat memenuhi kebutuhan pengguna dan tujuan proyek secara optimal.

Implementasi

Proyek ini menggunakan stack MERN (*MongoDB*, *Express.js*, *React*, *Node.js*) untuk membangun aplikasi berbasis web yang terintegrasi. *Frontend* dikembangkan dengan *React* untuk menciptakan antarmuka pengguna yang interaktif dan responsif, sedangkan *backend* menggunakan *Express.js* dan *Node.js* untuk mengelola API dan logika server-side. *MongoDB* dipilih sebagai sistem *database* *NoSQL* untuk menyediakan penyimpanan data yang fleksibel dan skalabel. Fitur utama aplikasi meliputi integrasi *YOLOv8* untuk deteksi sampah dan penerapan algoritma *Decision Tree Regression* untuk prediksi harga sampah. Selain itu, proses implementasi mencakup pembuatan halaman-halaman utama seperti registrasi, login, *dashboard* pengguna, serta fitur-fitur tambahan seperti deteksi sampah dan prediksi harga. Pendekatan ini bertujuan untuk memenuhi kebutuhan fungsional aplikasi dan meningkatkan efisiensi dalam pengelolaan sampah.

Pengujian

Pengujian aplikasi merupakan tahap kritis dalam memastikan kualitas dan keandalan sistem yang dikembangkan. Dalam proyek ini, tiga jenis pengujian dilakukan untuk mengevaluasi aplikasi secara menyeluruh: beta testing, validasi kuisioner, dan pengujian beban server.

1. Pengujian Beta
Beta testing dilakukan dengan melibatkan nasabah Bank Sampah Bersinar sebagai pengguna utama untuk menguji aplikasi dalam lingkungan nyata. Pengujian ini berfokus pada pengalaman pengguna dengan website dan mengidentifikasi potensi masalah serta area yang perlu perbaikan. Nasabah akan diberikan kuisioner yang berisi pertanyaan terkait penggunaan aplikasi, dan mereka akan menilai pengalaman mereka menggunakan skala nilai berikut: 1 (Sangat Tidak Setuju), 2 (Kurang Setuju), 3 (Netral), 4 (Setuju), dan 5 (Sangat Setuju). Data dari beta testing digunakan untuk memperbaiki aplikasi berdasarkan umpan balik yang diterima [5].
2. Validasi Kuisioner
Validasi kuisioner dilakukan untuk memastikan bahwa instrumen pengumpulan data yang digunakan dalam penelitian ini adalah valid dan dapat diandalkan. Validasi ini dilakukan dengan metode korelasi produk momen Pearson, yang digunakan untuk mengukur validitas setiap butir pertanyaan dalam kuisioner. Tingkat signifikansi yang diterapkan adalah 0.05 dengan uji dua sisi, dan nilai derajat kebebasan (df) yang digunakan adalah $df = n - 2$, di mana n adalah jumlah responden. Nilai tabel r untuk korelasi kemudian digunakan sebagai acuan; jika nilai korelasi (r) antara butir pertanyaan dan skor total kuisioner sama dengan atau lebih besar dari nilai r tabel, maka butir pertanyaan tersebut dianggap valid. Perhitungan lebih lengkap mengenai validitas kuisioner dapat ditemukan pada tabel yang terlampir, yang menunjukkan hasil evaluasi validitas dari setiap butir pertanyaan yang digunakan dalam penelitian ini.
3. Pengujian Beban
Pengujian beban server bertujuan untuk mengevaluasi performa aplikasi di bawah kondisi beban tinggi. Pengujian ini melibatkan

Table 1. labelformat=empty

Komponen	Solusi Terpilih	Alasan
Platform	Website	Memungkinkan aksesibilitas yang luas dan kemudahan integrasi dengan berbagai fitur serta teknologi web modern [7].
Frontend	React.js	Framework yang efisien untuk membangun antarmuka pengguna yang responsif dan dinamis, dengan dukungan komunitas dan ekosistem yang matang [8].
Backend	Express.js	Framework yang cepat dan efisien dalam membangun aplikasi web, serta integrasi yang baik dengan Node.js[9].
Database	MongoDB	Sistem manajemen basis data yang fleksibel dalam menangani data tidak terstruktur dan mudah untuk skala aplikasi yang berkembang[10].
Server	Cloud	Keunggulan server <i>cloud</i> dalam hal skalabilitas, fleksibilitas, dan biaya berbasis penggunaan memberikan kerangka kerja yang optimal untuk mendukung fitur-fitur pada proyek ini [10].

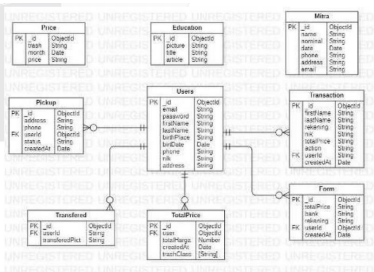
simulasi berbagai skenario beban untuk memastikan aplikasi dapat menangani volume pengguna dan transaksi yang tinggi tanpa mengalami penurunan performa atau kegagalan sistem. Hasil dari pengujian beban server memberikan informasi mengenai kapasitas dan kestabilan sistem, serta membantu dalam perencanaan skala dan manajemen sumber daya.

Hasil dan Pembahasan

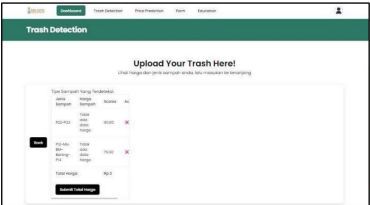
Pada penelitian kali ini, menghasilkan berbagai data untuk menunjang alat *swimming lap counter*. Berhasilnya integrasi antara mikrokontroler ESP32 dan sensor IR *obstacle avoidance* menghasilkan data pengujian waktu *respons* sekaligus jarak deteksi objek sensor dan tingkat kesalahan perhitungan durasi pada saat *race* renang.

Desain Rancangan Solusi

Pada sub bab ini, dipaparkan hasil dari evaluasi dan pemilihan solusi yang dilakukan untuk proyek tugas akhir ini. Setelah melalui pertimbangan dan perbandingan berbagai alternatif solusi, solusi terpilih yang diimplementasikan adalah sebagai berikut: 3 Sebagai bagian dari desain rancangan solusi, diagram hubungan entitas (ERD) untuk basis data disertakan untuk menggambarkan struktur data yang digunakan dalam aplikasi. Diagram ini menunjukkan entitas utama, atribut mereka, dan hubungan antar entitas, memberikan gambaran visual mengenai skema basis data yang mendukung keseluruhan solusi yang diimplementasikan.



Gambar 1. Entity Relationship Diagram.



Gambar 2. Tampilan pada layar laptop.

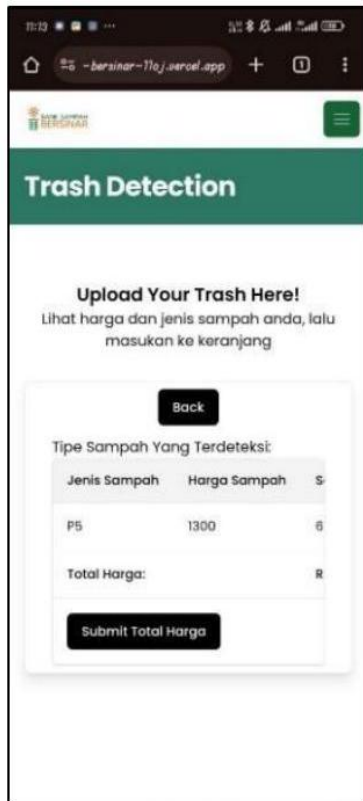
Implementasi

Implementasi aplikasi berbasis MERN melibatkan beberapa aspek kunci yang berhasil dicapai dalam proyek ini. Berikut adalah hasil-hasil utama dari proses implementasi:

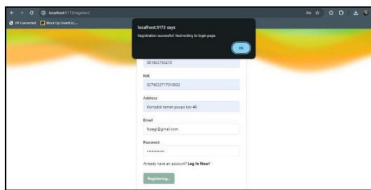
- 1. Halaman Website Responsif
Aplikasi web yang dikembangkan berhasil dioptimalkan untuk tampil dengan baik pada berbagai ukuran layar, termasuk layar laptop dan perangkat *mobile*. Dengan menggunakan *React.js* untuk *frontend*, antarmuka pengguna dirancang secara responsif, memastikan pengalaman pengguna yang konsisten dan optimal di semua perangkat. Desain responsif ini mencakup penyesuaian elemen UI dan tata letak yang dinamis, sehingga pengguna dapat mengakses dan menggunakan aplikasi dengan mudah, baik melalui desktop maupun perangkat seluler.
- 2. Data Berhasil Masuk ke Database
Implementasi *backend* menggunakan Express.js dan MongoDB memastikan bahwa data yang dikumpulkan dari pengguna berhasil disimpan dan dikelola dengan baik. Setiap interaksi dengan aplikasi, seperti pendaftaran pengguna atau pengumpulan data, diuji untuk memastikan bahwa data tersebut berhasil dikirim dan tersimpan di basis data. Pengujian ini mencakup verifikasi integritas dan konsistensi data, serta kemampuan basis data dalam menangani data secara efisien.
- 3. Frontend dan Backend Terdeploy ke Cloud
Deploy aplikasi ke *cloud* berhasil dilakukan dengan *frontend* dihosting di Vercel dan *backend* di-hosting di Azure Web App Service. *Deployment* ini memastikan aksesibilitas aplikasi yang luas dan skalabilitas. Konfigurasi lingkungan *cloud* untuk hosting aplikasi melibatkan pengaturan yang tepat pada Vercel untuk *frontend* dan pada Azure Web App Service untuk *backend*, memastikan bahwa semua komponen berfungsi dengan baik dalam lingkungan produksi.

Pengujian

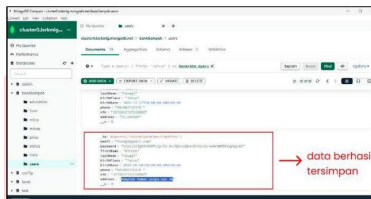
Pada sub bab ini, dibahas hasil dari berbagai jenis pengujian yang dilakukan untuk memastikan bahwa aplikasi berfungsi sesuai dengan



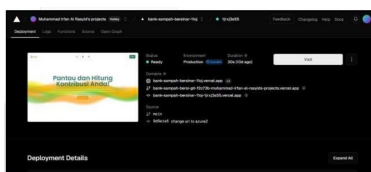
Gambar 3. Tampilan pada layar handphone.



Gambar 4. Hasil actual registrasi nasabah berhasil tersimpan.

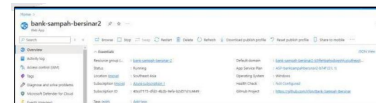


Gambar 5. Verifikasi penyimpanan data pada MongoDB.

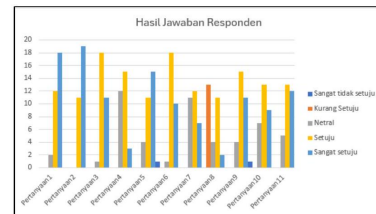


Gambar 6. Deployment pada Vercel.

spesifikasi dan memenuhi kebutuhan pengguna. Pengujian yang dilakukan meliputi pengujian beta, validasi kuesioner, dan pengujian beban server.



Gambar 7. Deployment pada Azure.



Gambar 8. Hasil Pengujian Beta.

1. Pengujian Beta

Pengujian beta testing melibatkan nasabah dari Bank Sampah Bersinar yang diundang untuk berpartisipasi dalam evaluasi produk secara luring. Nasabah yang berpartisipasi diminta untuk mengisi formulir penilaian yang mencakup berbagai aspek dari produk, dengan skala penilaian dari 1 hingga 5, di mana 1 berarti sangat buruk dan 5 berarti sangat baik. Dalam sub bab ini, kami akan menampilkan detail pengujian beta testing, termasuk data penilaian serta umpan balik yang diberikan oleh nasabah. Detail pengujian ini akan menjadi dasar bagi kami untuk melakukan perbaikan dan penyempurnaan sebelum produk diluncurkan secara resmi. Setelah melaksanakan pengujian beta 8 testing dengan melibatkan nasabah dari Bank Sampah Bersinar, kami memperoleh data penilaian yang menunjukkan respons dan umpan balik pengguna terhadap berbagai aspek produk kami. Hasil pengujian menunjukkan bahwa fitur login dan autentikasi admin berfungsi dengan sangat baik, di mana pengguna dapat login dengan *username* dan *password* yang *valid* serta sistem menampilkan pesan kesalahan ketika login gagal.

Navigasi *dashboard* juga dinilai mudah digunakan, termasuk kemampuan untuk tetap login setelah merefresh halaman, mengarahkan ke *dashboard* setelah login berhasil, dan *logout* dari *dashboard* dengan mudah. Pengelolaan data nasabah dan mitra ditampilkan dengan baik pada halaman *dashboard*, dan informasi rinci nasabah dapat diakses dengan mudah. Fungsi pencarian nasabah dan informasi mengenai penyetoran nasabah juga berfungsi dengan baik, membantu admin dalam menjalankan tugas mereka. Skor yang konsisten tinggi menunjukkan bahwa produk ini mudah digunakan dan memenuhi harapan pengguna. Umpan balik positif ini akan menjadi dasar untuk penyempurnaan akhir sebelum produk diluncurkan secara resmi ke pasar.

2. Validasi Kuisiонер

Berdasarkan hasil pengujian validitas kuesioner pada tabel 2, sebagian besar butir pertanyaan dapat dikatakan valid. Dengan demikian, dapat dikatakan bahwa kuesioner yang digunakan pada tugas akhir ini memiliki validitas yang baik. Beberapa pertanyaan yang tidak valid mungkin perlu direvisi atau dihapus untuk meningkatkan keandalan kuesioner.

3. Pengujian Beban

Pengujian beban 9 dilakukan menggunakan Postman untuk mengevaluasi performa aplikasi web yang telah dideploy di *Azure Cloud* tipe *Standard 1*. Pengujian ini menggunakan profil beban "ramp up" dengan pengguna virtual sebanyak 20, dimulai dari beban awal

Table 2. Hasil Validasi Kuisisioner

No. Pertanyaan	rxxy	r tabel	Status
Pertanyaan 1	0.47	0.306	Valid
Pertanyaan 2	0.52	0.306	Valid
Pertanyaan 3	0.48	0.306	Valid
Pertanyaan 4	0.39	0.306	Valid
Pertanyaan 5	0.45	0.306	Valid
Pertanyaan 6	0.40	0.306	Valid
Pertanyaan 7	0.42	0.306	Valid
Pertanyaan 8	0.37	0.306	Valid
Pertanyaan 9	0.50	0.306	Valid
Pertanyaan 10	0.44	0.306	Valid
Pertanyaan 11	0.46	0.306	Valid

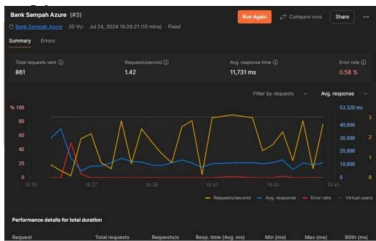
response), tingkat kesalahan (*error rate*), dan pengguna virtual yang aktif (*virtual users*) selama periode pengujian.

Kesimpulan

Kesimpulan dari penelitian ini menunjukkan bahwa solusi teknis yang dirancang menggunakan teknologi MERN stack (MongoDB, Express, *React*, Node.js) berhasil diimplementasikan dengan baik. Proses implementasi mencakup pengembangan website dan integrasi dengan *database* MongoDB untuk penyimpanan data. Pengujian aplikasi mengonfirmasi bahwa semua fitur berfungsi sesuai ekspektasi, dengan integrasi antara *frontend* dan *backend* berjalan lancar dan data disimpan dengan akurat. Sistem ini siap digunakan dalam konteks nyata, menunjukkan efektivitas solusi yang dirancang dan kesesuaian dengan spesifikasi yang ditetapkan.

Daftar Pustaka

1. Kadang JM, Sinaga N. Pengembangan Teknologi Konversi Sampah Untuk Efektifitas Pengolahan Sampah dan Energi Berkelanjutan. 2024.
2. Muliada IMP, Paramitha AAI, Purnama IN. Pengembangan Booking Engine Pada Gripastudio. Jurnal RESTIKOM: Riset Teknik Informatika dan Komputer. 2024;6(2):261-70.
3. Sanubari T, Prianto C, Riza N. Odol (One Desa One Product Unggulan Online) Penerapan Metode Naive Bayes pada Pengembangan Aplikasi E-Commerce Menggunakan CodeIgniter. Kreatif. 2020;1.
4. Armbrust M, et al. A View of Cloud Computing. Communications of the ACM. 2010;53(4):50-8.
5. Mohd CKNCK, Shahbodin F. Personalized Learning Environment: Alpha Testing, Beta Testing & User Acceptance Test. Procedia - Social and Behavioral Sciences. 2015;195:837-43.
6. Hiles M, Agha M. Knowledge-Based ITS for Teaching Mongo Database. 2017.
7. Kamthan P. User Interface Design Principles;.
8. Mardan A, Mardan A. In: Using Express.js to Create Node.js Web Apps; 2018. p. 51-387.
9. Ouzzani M, Hammady H, Fedorowicz Z, Elmagarmid A. Rayyan: A Web and Mobile App for Systematic Reviews. Systematic Reviews. 2016;5:210.
10. Pahl C, Xiong H, Walshe R. A Comparison of On-Premise to Cloud Migration Approaches. In: Service-Oriented and Cloud Computing: Second European Conference, ESOC 2013, Málaga, Spain, September 11–13, 2013. Proceedings. Springer; 2013. p. 212-26.
11. Saks E. JavaScript Frameworks: Angular vs React vs Vue; 2019.



Gambar 9. Pengujian Beban Pada Postman.

Table 3. Hasil Pengujian Beban

Parameter	Hasil
Total Permintaan yang Dikirim	861
Permintaan per Detik	1,42
Waktu Respons Rata-rata	11,731 ms
Tingkat Kesalahan	0.58 %
Waktu Respons Minimum	90 ms
Waktu Respons Maksimum	53,320

sebanyak 5 pengguna, dan berlangsung selama 10 menit. Berikut adalah hasil dari pengujian beban tersebut dalam bentuk tabel:

Grafik hasil pengujian 3 menunjukkan variasi jumlah permintaan per detik (*requests/second*), waktu respons rata rata (*avg*).