

Implementasi Web Service Pada Sistem Akuakultur Berbasis *Internet of Things*

Rika Rahmawati, Muhammad Nasir*, Hari Toha Hidayat, Amri, Ayu Ramadhani

Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe

Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA

*E-mail : muhnasir.tmj@pnl.ac.id

Abstract

Article history:

Received: 13-11-2025

Accepted: 08-12-2025

Published: 31-12-2025

Keywords:

aquaculture;

internet of things;

jmeter;

waiting time;

web service.

Aquaculture is oriented towards a sustainable environment that requires more attention, fish can live and breed well if their environmental requirements have been met or are close to their natural habitat. The main parameter in carrying out fish farming is water quality. The main parameter in carrying out fish farming is water quality. Water quality needs to be monitored regularly so that success in aquaculture is achieved. API that functions to integrate between applications to facilitate communication between the two. The method used, namely rest web service, is expected to be used by IoT device developers and web application developers for monitoring and controlling freshwater fish farming. The results of the implementation of web services in an internet-based aquaculture system of things with this to make it easier to do. With this system, monitoring and control can also be done remotely (online) and in real time by cultivators. The results obtained from testing using jmeter conducting load testing tests were obtained using virtual users and there were response times in milliseconds. Testing the response times based on the number of users, 250 users entered the 1.0 category with a description of some lags, but users are unlikely to see the disturbance and when entering thousands of users experience a long delay, which is more than a maximum waiting time of 10 seconds so that this causes users to experience interruptions due to the resulting waiting time with an error rate above 50.00%.

1. Pendahuluan

Ikan merupakan salah satu makhluk hidup air yang sangat dibutuhkan oleh manusia[1, 2]. Beberapa jenis ikan, seperti ikan laut dan ikan air tawar, adalah makanan penting untuk pertumbuhan tubuh dan sumber protein yang baik. 18% protein ikan terdiri dari asam amino esensial yang tidak mudah rusak selama proses pengolahan. Sangat disarankan untuk dikonsumsi dalam jumlah besar karena kandungan gizinya yang tinggi dan hampir sama dengan ikan air laut, sehingga memenuhi kebutuhan tubuh untuk pertumbuhan dan penurunan kolesterol darah[3, 4]. Ikan air tawar memiliki kandungan protein dan vitamin yang tinggi, yang membuatnya mudah dibudidayakan dan dikembangbiakkan[4, 5].

Keberhasilan budidaya perikanan bergantung pada perhatian yang lebih besar untuk melestarikan lingkungan hidup. Ikan dapat hidup dan berkembang biak dengan baik jika persyaratan lingkungannya telah dipenuhi atau jika mereka lebih dekat dengan habitat aslinya. Untuk mencapai tujuan ini, kualitas air harus dipantau secara berkala.

Sistem akuakultur adalah salah satu cara untuk meningkatkan kualitas budidaya ikan. Akuakultur adalah jenis budidaya yang menggabungkan sistem rekayasa manusia

dengan memanfaatkan energi untuk meningkatkan produksi organisme akuatik, termasuk pemeliharaan, penanganan, pengelolaan, dan pemasaran[6, 7]. Namun, penerapan sistem akuakultur memerlukan pengawasan ketat, jadi akan sulit jika diterapkan jauh dari lokasi pelaku budidaya ikan.

Pengawasan yang perlu dilakukan meliputi pengawasan pakan, kualitas air, dan filtrasi air. Pengawasan saat ini dilakukan secara manual, melakukan pengecekan secara langsung di tempat pembudidayaan, yang membuat budidaya tidak efektif. Oleh karena itu, diperlukan pengawasan dari jarak jauh daripada melakukan pengecekan secara langsung. Aplikasi yang disebut web service terdiri dari database, perangkat lunak, atau bagian dari program perangkat lunak yang dapat diakses dari jarak jauh melalui perangkat dengan perantara yang dimaksud untuk memudahkan interaksi antar aplikasi[8, 9].

Berdasarkan permasalahan tersebut, maka terciptalah suatu pemikiran pembuatan implementasi *web service* pada sistem akuakultur berbasis IoT (*Internet of things*). Sistem dirancang untuk pemantauan dan pengendalian pembudidayaan jarak jauh serta *real time* dengan medium internet untuk memfasilitasinya, melihat dan memberikan *alert*

pemberian pakan, monitoring kekeruhan air, monitoring kadar pH pada air, dan monitoring penyaringan air. Seluruh sistem bekerja secara berkala sesuai dengan kondisi dan waktu yang telah ditetapkan. Ini memungkinkan mereka untuk memberikan informasi kepada peternak ikan tentang cara meningkatkan kualitas dan kuantitas hasil panen mereka. Seluruh layanan web diharapkan dapat diakses dengan mudah pada berbagai perangkat[10, 11].

2. Metode

2.1 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data yang digunakan untuk membuat implementasi *Web Service* pada sistem akuakultur berbasis *Internet of things*.

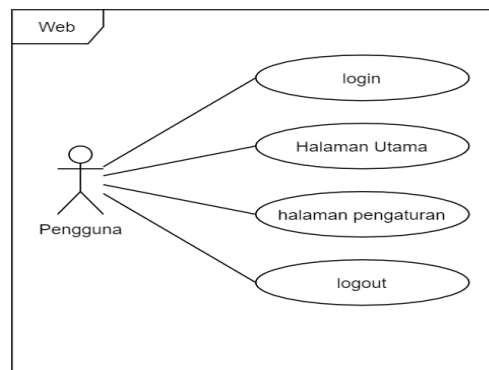
Teknik pengumpulan data yang digunakan untuk dilakukan penelitian adalah teknik pengumpulan data observasi, dengan melakukan pengamatan langsung agar mendapatkan data akurat yang akan dibuatkan sistem dan teknik pengumpulan data melalui wawancara, yaitu pengambilan data didapatkan secara langsung dengan mengajukan pertanyaan antara pewawancara dengan narasumber. Pengambilan data juga perlu dipersiapkan pertanyaan yang baik, agar kualitas data yang didapat efektif.

2.2 Rancangan Sistem (Software)

Tujuan dari rancangan sistem adalah untuk mendapatkan gambaran tentang apa yang akan dilakukan sistem. Dalam kajian ini rancangan sistem disusun dalam blok diagram sistem, diagram kasus, dan antarmuka pengguna.

2.3 Use Case Diagram

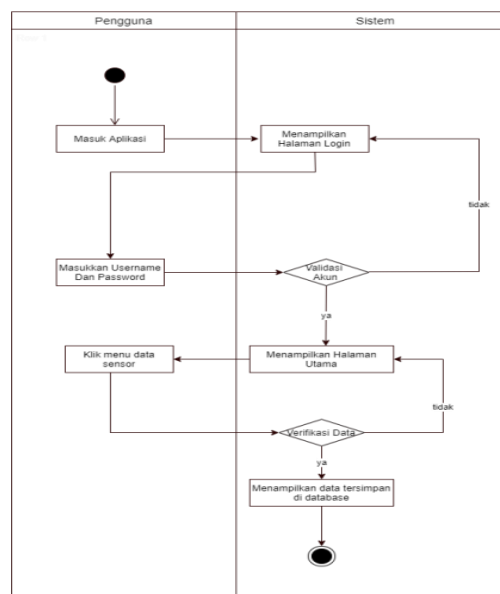
Use case diagram dibuat untuk melihat cara kerja sistem yang berjalan, terutama pada penggun. Rancangan use case diagram diperlihatkan pada Gambar 1. Dari Gambar 1 hanya terdapat aktor pengguna yang akan digunakan untuk membangun sistem ini dengan melakukan pengelolaan pada data sensor yang didapat dari sistem monitoring yang telah dibuat.



Gambar 1. Use case diagram pengguna

2.4 Diagram Activity

Tahap ini merupakan diagram aktivitas terdiri dari alur kerja, yang mencakup aktivitas untuk menjelaskan alur aktifitas sistem dan pengguna. Diagram activity pengguna diperlihatkan pada Gambar 2.



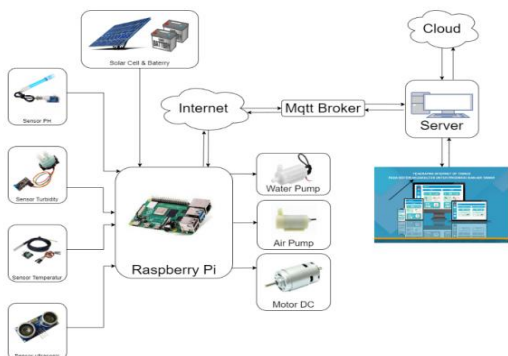
Gambar 2. Diagram activity pengguna

Setelah melakukan login, pengguna (pembudidaya) akan memiliki kemampuan untuk memantau atau mengontrol budidayanya. Untuk memantau, pengguna akan mengakses data di *restful web service*, dan untuk mengendalikan, pengguna akan mengirimkan data tertentu melalui *restful web service* ke *Raspberry Pi*, yang kemudian akan diteruskan ke alat pendukung.

2.5 Block Diagram

Pada tahap ini, diagram aktivitas memuat alur kerja. Di dalamnya, terdiri dari aktivitas yang menjelaskan alur aktivitas sistem dan

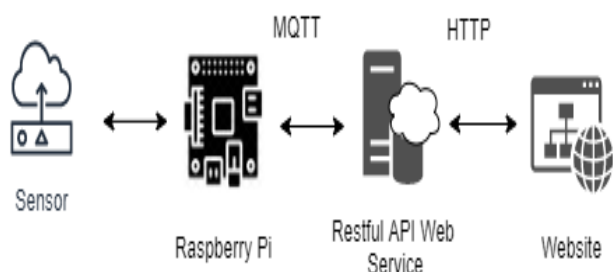
pengguna. Block diagram system diperlihatkan pada Gambar 3.



Gambar 3. Block diagram sistem

Dari Gambar 3 terlihat bahwa block diagram sistem, dimana sensor yang terdiri dari sensor pH, turbidity, temperatur dan ultrasonic akan mengirimkan data ke raspberry pi kemudian data tersebut akan dikirim melalui internet dengan sebuah protokol MQTT maka data akan dikirim pada database sebagai report dari masing-masing sensor yang akan menjalankan water pump, air pump dan motor dc sesuai dengan data yang telah diolah sebelumnya.

Block diagram web service diperlihatkan pada Gambar 4. Dari gambar 4 menjelaskan proses pertukaran data antara sistem pengawasan dan pengguna dimana perantara MQTT akan menyimpan semua output data alat di dalam cloud. Selanjutnya pengguna akan diinformasikan melalui website [12, 13].



Gambar 4. Block diagram web service

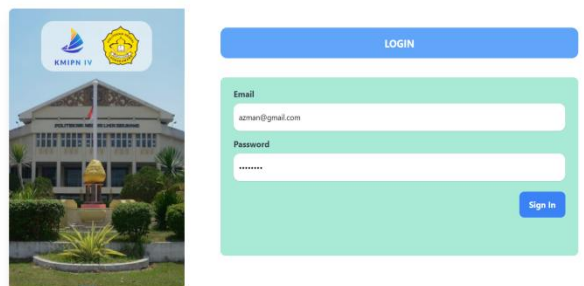
3. Hasil dan Pembahasan

3.1 Implementasi User Interface

Setiap halaman utama user interface yang berkaitan dengan pengelolaan data sistem monitoring akan ditampilkan di bagian perancangan user interface. User Interface diimplementasikan sesuai dengan perancangan ini.

3.1.1 Implementasi pada Halaman Login

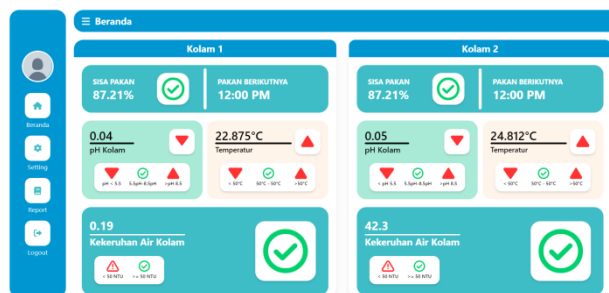
Halaman login ini digunakan untuk memasukkan email dan password pengguna yang digunakan untuk mengakses. Rancangan user-interface halaman login diperlihatkan diperlihatkan pada Gambar 5.



Gambar 5. Halaman login

3.1.2 Implementasi pada Halaman Utama

User interface halaman utama ditunjukkan pada Gambar 6. Gambar 6 menampilkan halaman utama dengan hasil sensor yang didapatkan sehingga memberikan data dan informasi. Data yang dihasilkan dari sensor akan ditampilkan dalam tampilan web Kolam 1 dan 2 memiliki data berbeda. Pada kolom 1, keterangan sisa pakan adalah 87,21%; pakan berikutnya datang pada 8:00 AM dengan pH kolam 0,04, suhu 22,875 derajat celcius, dan nilai kekeruhan air 0,19. Di kolom 2, pH adalah 0,05, suhu 24,812 derajat celcius, dan nilai kekeruhan air adalah 42,3.

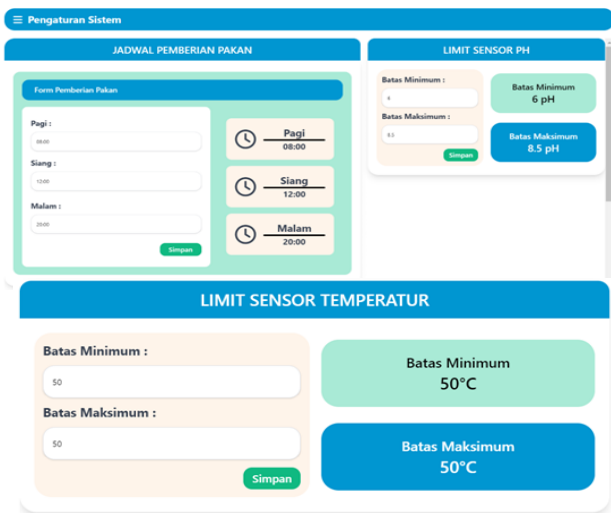


Gambar 6. Halaman utama

3.1.3 Implementasi pada Halaman Pengaturan

User interface halaman pengaturan diperlihatkan pada Gambar 7. Dari Gambar 7 terlihat bahwa terdapat jadwal pemberian pakan pada halaman pengaturan sistem. Pengguna dapat mengatur jadwal pemberian pakan pada 3 waktu yaitu pagi, siang, dan malam. Limit sensor pH dan limit sensor temperatur. Limit sensor pH menunjukkan batas minimum dan maksimum,

dan limit sensor temperatur juga dapat diatur, dengan batas minimum pH 6 dan batas maksimum pH 8,5. Jika semua diatur sesuai dengan kebutuhan pengguna, data dapat disimpan.



Gambar 7. Halaman pengaturan

3.1.4 Implementasi pada Halaman Report

Halaman report memberikan hasil pembacaan secara *real time* dari data sensor turbidity, pH, dan temperatur. Hasil pembacaan sensor turbidity diperlihatkan pada Gambar 8.



Gambar 8. Hasil report sensor turbidity

Gambar 8 memperlihatkan grafik dan data yang tercatat dari sensor *turbidity*, dilakukan selama lebih kurang 30 menit sehingga sebanyak 54 data yang ada dikirim. Data yang didapat mengalami perubahan karena ditampilkan dalam persatuan detik (*real time*). Dijelaskan standar dalam mengukur kekeruhan pada air atau yang dikenal dengan *NTU* (*Nephelometric*

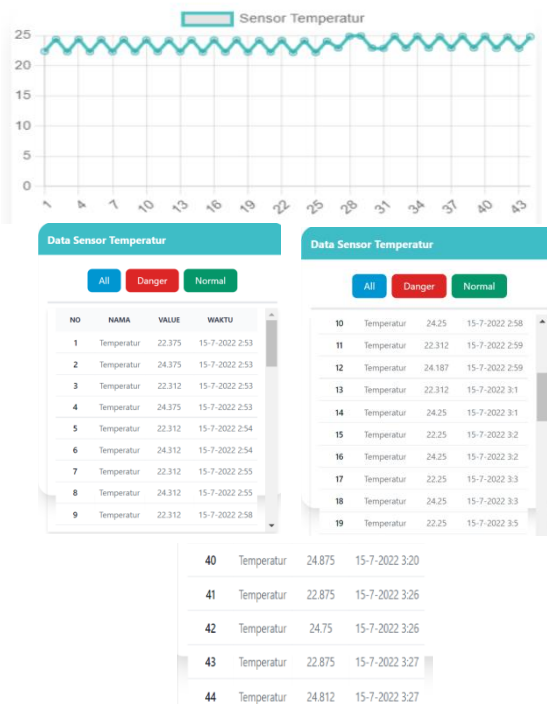
Turbidity Unit). Standar kekeruhan air ditetapkan antara 5-25 *NTU*. Sehingga apabila nilai melebihi 50 maka grafik akan naik dan bila kurang dari 50 grafik akan turun. Oleh karena itu grafik dibuat sesuai dengan *output* dari sensor yang muncul, data mulai meningkat dari data ke 42-54.

Hasil pembacaan sensor pH diperlihatkan pada Gambar 9. Gambar 9 merupakan data dari sensor pH, yang dilakukan selama lebih kurang 30 detik didapatkan *output* sebanyak 36 data maka menampilkan grafik. Adapun batasan pH yang ada didalam air bahwa apabila $pH < 6$ maka air mengalami peringatan dan untuk pH stabil $6pH-8,5pH$ serta pH diatas 8,5 juga akan mengalami peringatan. Kisaran data yang didapat tidak stabil terlihat dari grafik kenaikan angka pada kisaran 11-15 dan mengalami penurunan mencapai angka 0.02. Grafik mengalami peningkatan pada data ke 2-28 dan selanjutnya mengalami penurunan sehingga data tersebut mengirim peringatan kepada user bahwa pH air tidak stabil.



Gambar 9. Sensor pH

Hasil pembacaan sensor temperatur diperlihatkan pada Gambar 10. Temperatur memiliki batas maksimum dan minimum sebesar 50 derajat celsius. Dibawah 50 derajat celsius akan menampilkan peringatan untuk netral 50 derajat celsius dan lebih besar dari 50 derajat celsius akan mengalami peringatan.



Gambar 10. Sensor temperatur

Gambar 10 menunjukkan bahwa pengamatan sensor temperatur dilakukan selama 30 menit dan diperoleh data sebanyak 44 data. Untuk data terlihat dari 22 – 25 derajat celsius hal ini menandakan bahwa temperatur pada saat itu mengalami kondisi dibawah dari 50 derajat Celsius maka dikatakan bahwa temperatur pada saat itu mengalami penurunan dari kondisi netral, sehingga grafik mengalami penurunan.

3.2 Implementasi pada Sistem

Dalam pembuatan API, menggunakan bahasa pemrograman *java script* dengan *runtime node js*, *Vuejs* memiliki kemudahan untuk dilakukan intergrasi antar *framework* atau *library javascript* lain dan memiliki kemampuan menjalankan *single-page application* yang canggih[14].

3.2.1 Pengujian Sistem

Pengujian ini akan dilakukan pada web dengan menggunakan aplikasi *jmeter* untuk mengetahui keluaran berupa *respon time*, *throughput* dan *error*. Tahapan ini menggunakan 4 API yaitu *get feeding times*, *limit*, *reports* dan *post reports*. Pengujian sistem untuk menguji *response time* untuk melihat apakah API tersebut layak atau tidak, memakai *POSTMAN* dan *APACHE Jmeter*. *POSTMAN* digunakan untuk memvalidasi respon data dari tiap fungsi yang

diuji sedangkan *APACHE Jmeter* akan digunakan untuk mengetahui seberapa banyak *request* yang bisa dilayani dengan menggunakan jumlah *request* dan *response time* yang didapat.

3.2.2 Pengujian Request Fungsi

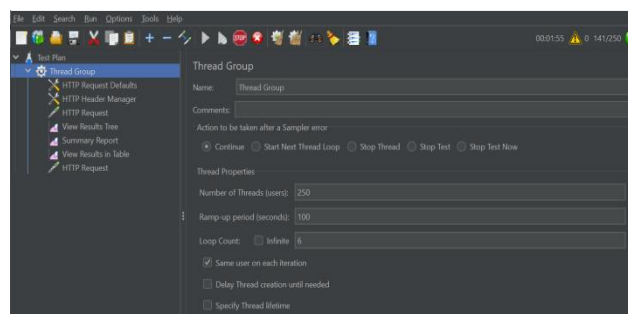
Hasil pengujian request fungsi ditunjukkan dalam Tabel 1. Pengujian pertama yang dilakukan ditampilkan menggunakan *POSTMAN* dengan beberapa fungsi *response time* rata-rata diatas 300 ms. Hasil kajian terdahulu menunjukkan bahwa *response time* data *attribute* masyarakat seperti *NIK*, tempat tanggal lahir dan sejenisnya membutuhkan waktu sekitar 168 ms untuk single request dan 1032 ms untuk 100 request secara bersamaan[12]. Jika dibandingkan, maka hasil pengujian yang diperoleh pada penelitian ini sudah dalam kategori baik

Tabel 1. Pengujian request fungsi

Sample	Function	Time
1	Get Feeding Times	781 ms
2	Get Limits	64 ms
4	Get Reports	73 ms
5	Post Reports	891 ms
6	Get User	51 ms

3.2.3 Pengujian Load Testing

Pengujian selanjutnya dengan menggunakan aplikasi *apache jmeter* untuk melihat *performance* atau ketahanan *server* meminta *request* yang dilakukan yang akan menghasilkan keluaran *respon time* (ms) dan *error rate* (%). Jumlah *user* sebanyak 250 orang, setiap 2 detik (100/50) akan mengirimkan 6 request ke server. Sehingga jumlah sample sebanyak 300 (50x6) terlihat pada gambar 11 tampilan dari *jmeter*.



Gambar 11. Tampilan Jmeter

Selanjutnya dilakukan pengujian dengan menggunakan metode *load testing* dengan menggunakan *request sample* 250, 500, 1000, 1200 dan 2000 *user*. Secara berurut mengisi nilai

user sesuai dengan *request sample* dengan *rump up periods* 100 dan nilai *loop count* sebanyak 6 seperti diperlihatkan dalam Table 2.

Tabel 2. Pengujian pada *load testing*

API	Jumlah Request (vu)	Respon Time (ms)	Error Rate %
GET/Feeding_Times	250	4203	0.00%
	500	15395	0.03%
	1000	24665	30.28%
	1200	27274	37.72%
	2000	26961	52.20%
GET/Limit	250	6380	50.00%
	500	18452	50.02%
	1000	23626	72.23%
	1200	27833	70.94%
	2000	28203	80.93%
GET/Reports	250	4897	50.00%
	500	17748	50.00%
	1000	26690	69.12%
	1200	35854	72.15%
	2000	30080	81.35%
POST/Reports	250	5996	50.00%
	500	17310	50.00%
	1000	24839	72.32%
	1200	29146	71.85%
	2000	28484	80.94%

Dari hasil pengujian *load testing*, percobaan server pada *GET/feeding_times* dengan permintaan 250, 500, 1000, 1200, dan 2000 user memiliki rentang 4203-37274 *millisecond* (ms) dengan rata-rata 19.699 ms. Selanjutnya, percobaan server pada *GET/Limit* memiliki rentang 6380-28203 *millisecond* (ms) dengan rata-rata 20.898 ms, dengan kenaikan waktu dari sebelumnya. *GET/Reports* memiliki rentang 4897-35854 *millisecond* (ms) dengan rata-rata 23.054 ms, dan POST.

Berdasarkan hasil pengujian yang dilakukan pada *load testing* setiap API dengan jmeter, maka API akan memberikan bentuk perilaku berdasarkan dari *avg.response time* yang dihasilkan pada API mengalami peningkatan seiring dengan bertambahnya jumlah *request*. Hal itu dibuktikan bahwa API akan berkurang seiring dengan jumlah user yang masuk menghasilkan *error rate* diatas 50.00%. Pada pengujian diatas merupakan nilai yang diambil untuk melihat virtual user yang sebesar 250, 500, 1000, 1200 dan 2000. Kategori jumlah pengguna sesuai dengan jumlah maksimum hal ini agar tingkat *error* yang didapatkan dengan nilai yang bervariasi. *Response time* akan menunjukkan waktu yang dibutuhkan oleh server untuk melakukan respon dengan *client*,

karena semakin cepat *response time* maka server akan semakin bagus[15]. Hasil dari uji tersebut pada setiap API dijalankan dalam waktu 1 detik ketika 250 user namun ketika diberikan *virtual user* yang lebih besar maka mengalami respon yang melebihi dari 6 detik sehingga mengalami keterlambatan dan waktu tunggu yang lama.

3.2.4 Pengujian Response Time

Pada pengujian ini akan melihat kategori yang didapatkan dari beberapa API dengan melihat *response time* dan mengubah dari *millisecond* ke *second*. *Response times testing* memiliki dua karakteristik yang penting yaitu waktu response rata-rata dan waktu response maksimum. Ketentuan uji *response time* diperlihatkan dalam Tabel 3.

Tabel 3. Ketentuan untuk uji *response time*

Response Time	Penjelasan
0.1 detik	Waktu response yang paling baik. Jika waktu response adalah 0.1 pengguna akan merasa sistem atau aplikasi merespon secara instan sehingga tidak merasakan gangguan apapun.
1.0 detik	Batas maksimum waktu respons yang dapat diterima. Pengguna tidak merasakan gangguan meskipun mengalami keterlambatan sedikit, tetapi <i>response</i> lebih dari 1 detik dapat mengganggu pengalaman pengguna.
10 detik	waktu <i>response</i> yang telah melampaui batas yang dapat diterima. Jika pengguna menerima waktu respons lebih dari 6 detik, dia akan meninggalkan halaman atau memuatnya kembali. Ini adalah batas paling atas yang dilintasi untuk waktu respons.

Tabel 4 memperlihatkan pengujian *response time* dalam kategori. Dari Tabel 4 terlihat bahwa tekanan pengguna yang diberikan untuk setiap API memasukkan 250 user ke dalam kategori 1 detik, yang berarti pengguna tidak merasa terlambat karena tetap dalam kategori cepat, sehingga mereka tidak merasa terganggu. Namun, ketika tekanan pengguna berjumlah 500, 1000, 1200, dan 2000, maka kategori tersebut dimasukkan ke dalam 10 detik, yang membuat pengguna merasa terganggu dan tidak dapat melakukan apa-apa.

Tabel 4. Pengujian *response time* dalam kategori

API	Jumlah Request (Vu)	Response Time (s)	Kategori		
			0.1	1	10
GET/Fe eding.T imes	250	4.203		√	
	500	15.395			√
	1000	24.665			√
	1200	27.274			√
	2000	26.961			√
GET/Li mit	250	6.38		√	
	500	18.452			√
	1000	23.626			√
	1200	27.833			√
	2000	28.203			√
GET/Re ports	250	4.897		√	
	500	17.748			√
	1000	26.69			√
	1200	35.854			√
	2000	30.08			√
POST/R eports	250	5.996		√	
	500	17.31			√
	1000	24.839			√
	1200	29.146			√
	2000	28.484			√

3.2.5 Pengujian Selenium Web

Untuk memastikan bahwa fitur yang telah dibangun sesuai, *selenium webdriver* digunakan untuk melakukan pengujian. Dengan menampilkan data tentang pH pakan, parameter, suhu, dan kekeruhan air kolam pada url beranda, dapat dianggap berhasil. Kemudian, dengan menggunakan url yang diarahkan, juga dapat menampilkan dan melakukan pengaturan untuk pengontrol pH dan suhu, jam pemberian pakan, sesuai dengan batas yang telah ditetapkan sebelumnya. Hasil pengujian dengan selenium ditunjukkan dalam Tabel 5.

Tabel 5. Pengujian dengan selenium

URL	Berhasil Diakses	Tidak Berhasil Di Akses	Keterangan
Http://localhost:8080/login	√		Testing sesuai dengan harapan
Http://localhost:8080/beranda	√		Testing sesuai dengan harapan
Http://localhost:8080/setting	√		Testing sesuai dengan harapan
Http://localhost:8080/reports	√		Testing sesuai dengan harapan
Http://localhost:8080/logout	√		Testing sesuai dengan harapan

Tabel 5 menampilkan hasil yang didapat dari aplikasi selenium dan dapat dilihat bahwa uji yang dilakukan pada halaman tersebut berhasil atau sesuai dengan harapan karena tidak mengalami kendala dalam saat dilakukan uji.

4. Kesimpulan

Setelah melakukan perancangan dan pengujian pada implementasi *web service* pada sistem akuakultur berbasis *internet of things*, maka didapatkan simpulan bahwa:

Pengujian dilakukan dengan melihat persentase kecepatan respons waktu pada *webservice* dari nilai rata-rata respons waktu dan nilai throughput. Dalam penelitian, beberapa API tersebut berhasil mengirimkan data. Menguji respons waktu berdasarkan jumlah user, dengan antara 250 dan 2000 user, beberapa user masuk ke dalam kategori 1.0 dengan keterangan bahwa ada kelambatan, tetapi pengguna tidak dapat melihatnya. Waktu respons lebih dari satu detik dapat memengaruhi pengalaman pengguna. Namun, ketika waktu respons ditambahkan untuk kategori lebih dari sepuluh detik, tingkat kesalahan juga meningkat, dengan rata-rata lebih dari lima puluh persen.

Pengujian dengan *selenium web* memungkinkan untuk melihat kinerja fitur web yang diuji; setiap fitur memiliki hasil yang baik, artinya tidak ada kegagalan.

Ucapan Terima Kasih

Penghargaan dan ungkapan terima kasih kepada pihak-pihak yang telah berkontribusi dalam penelitian yang telah dilakukan. Bila tidak ada bagian ini tidak diperlukan.

Daftar Pustaka

- [1] Lumentut, H. B.& Hartati, S., 2015. *Sistem pendukung keputusan untuk memilih budidaya ikan air tawar menggunakan af-topsis*. IJCCS (Indonesian Journal of Computing and Cybernetics Systems), Vol. 9, No. 2, pp. 197-206.
- [2] Andhikawati, A., Junianto, J., Permana, R., & Oktavia, Y., 2021. *Komposisi gizi ikan terhadap kesehatan tubuh manusia*. Marinade, Vol. 4, No. 02, pp. 76-84.
- [3] Astawan, M., "Ikan air tawar kaya protein dan vitamin. Artikel departement of food science and technology bogor agricultural university," ed, 2016.

- [4] Rasmi, D. A. C., Sedijani, P., & Sativa, D. Y., 2023. *Ayo penuhi zat gizi protein dengan konsumsi ikan*. *Jurnal Pengabdian Magister Pendidikan IPA*, Vol. 6, No. 4, pp. 1348-1352.
- [5] Alimaturahim, F. *et al.*, 2024. *Ekosistem kolam ikan air tawar*. TOHAR MEDIA.
- [6] Rejeki, S., Aryati, R. W., & Widowati, L. L., "Pengantar akuakultur," ed: Undip Press, 2019.
- [7] Haikal, M., Mursyidah, M., & Nasir, M., 2023. *Penerapan iot (internet of thing) pada sistem monitoring dan kontrol aquarium berbasis web service*. *Jurnal Teknologi Rekayasa Informasi dan Komputer*, Vol. 6, No. 2.
- [8] Salsabila, F.& Intani, S. M., 2020. *Sejarah web service beserta arsitektur dan penggunaannya*. Matrik: *Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*.
- [9] Kurniadi, D., 2017. *Sistem kendali jarak jauh perangkat elektronik rumah berbasis cloud computing*. *Jurnal Algoritma*, Vol. 14, No. 2, pp. 333-342.
- [10] Darmawansyah, D.& Syaryadhi, M., 2018. *Restful web service untuk pemantauan dan pengendalian peternakan ayam broiler*. *Karya Ilmiah Mahasiswa Teknik*.
- [11] Hanifah, S.& Nasir, M., 2023. *Penerapan cloud computing pada sistem monitoring dan kontrol sensor ph dan sensor suhu untuk aquaculture berbasis internet of things*. *Journal of TIK*, Vol. 3, No. 2, pp. 130-135.
- [12] Sitorus, Z., 2018. *Kebutuhan web service untuk sinkronisasi data antar sistem informasi dalam universitas: Zulham sitorus, meri sri wahyuni, supina batubara*. *Jurnal Teknik Dan Informatika*, Vol. 5, No. 2, pp. 87-90.
- [13] Nasir, M., Hidayat, H. T., Anwar, A., & Rudi, F. Y., Year. *Penerapan iot pada smart laundry berbasis rfid (e-ktp)*. in *Prosiding Seminar Nasional Politeknik Negeri Lhokseumawe*, Vol. 4, No. 1.
- [14] Ozdemir, E., 2020. *A general overview of restful web services*. *Applications and approaches to object-oriented software design: emerging research and opportunities*, pp. 133-165.
- [15] Sulisty, W., 2023. *Analisis kehandalan global server load balancing dengan algoritma fixed weighting*. *Jurnal sosial dan sains*, Vol. 3, No. 5, pp. 477-482.