

Peran *Quality Assurance Testing* Dalam Pengembangan *Website* Klasifikasi Kualitas Air Sungai Citarum

1st Rai Barokah Utari
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

raiuariiii@student.telkomuniversity.ac.id

2nd Meta Kallista
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

metakallista@telkomuniversity.ac.id

3rd Ig. Prasetya Dwi Wibowo
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

prasdwiwawa@telkomuniversity.ac.id

Abstrak— Sungai Citarum, yang merupakan sungai terpanjang di Jawa Barat, memiliki peran penting dalam kehidupan masyarakat di sekitarnya. Selain sebagai sumber air, sungai ini juga mendukung berbagai aktivitas domestik, pertanian, dan industri. Namun, pada tahun 2013, Sungai Citarum dinobatkan sebagai sungai terkotor di dunia. Oleh karena itu, diperlukan suatu inovasi untuk menganalisis kualitas air sungai menggunakan sistem informasi yang pastinya sebelum digunakan layanannya oleh masyarakat diperlukannya *Quality Assurance testing*. Hal ini bertujuan untuk memverifikasi dan menjamin kualitas produk perangkat lunak agar bisa digunakan dengan baik. *Quality Assurance testing* sangat diperlukan guna memastikan *User Interface (UI)* dan *User Experience (UX)* yang lancar.

Kata kunci— *Quality Assurance testing*, Sungai Citarum, Pencemaran Air, Kualitas Air

I. PENDAHULUAN

Sungai Citarum, sebagai salah satu sungai terpanjang di Indonesia, memiliki peran vital bagi kehidupan masyarakat di Jawa Barat. Namun, kualitas airnya kian terancam akibat pencemaran yang kian parah. Oleh karena itu, diperlukannya suatu inovasi untuk menjadi sumber informasi penting bagi masyarakat untuk memantau kondisi sungai dan mengambil langkah-langkah pencegahan. Melalui platform ini, data kualitas air diperbarui secara berkala menggunakan teknologi pembelajaran mesin, memberikan informasi yang akurat dan terkini. Masyarakat dapat mengakses informasi mengenai tingkat pencemaran, parameter kualitas air, dan rekomendasi tindakan yang perlu diambil. Dengan demikian, kolaborasi antara teknologi dan kesadaran masyarakat diharapkan dapat membantu memulihkan dan melestarikan ekosistem Sungai Citarum. Upaya ini sangat penting untuk memastikan keberlanjutan sumber daya air bagi generasi mendatang[1].

Diketahui bahwa sepanjang 127 km atau 47,1% dari panjang sungai telah dikategorikan tercemar berat[2]. Sungai Citarum dicemari oleh kurang lebih 20,000 ton sampah dan 340.000 ton air limbah dengan mayoritas penyumbang limbah tersebut berasal dari 2.000 industri tekstil[3].

Sebuah sistem informasi yang dirancang untuk mengklasifikasikan kualitas air Sungai Citarum perlu dikembangkan bersamaan dengan upaya pemetaan sungai

tersebut. Sistem ini bertujuan untuk mengatasi masalah pencemaran dan dampaknya terhadap lingkungan serta masyarakat sekitar. Melalui sistem informasi ini, masyarakat akan memiliki akses yang lebih baik terhadap data mengenai kualitas air sungai, yang diharapkan dapat meningkatkan kesadaran mereka tentang kondisi sungai. Dengan kesadaran yang meningkat, masyarakat diharapkan akan lebih proaktif dalam menjaga dan melindungi kesehatan sungai, serta menghindari aktivitas yang dapat merusaknya.

Dalam hal ini, *QA testing* (Quality Assurance testing) memegang peranan krusial untuk memastikan *website* kualitas air Sungai Citarum berfungsi dengan baik dan memberikan informasi yang akurat kepada penggunaannya. *QA testing* membantu mengidentifikasi dan memperbaiki bug, memastikan kompatibilitas dengan berbagai perangkat dan *browser*, serta meningkatkan kinerja *website*[4]. Berikut beberapa alasan harus diadakannya *QA testing* diantaranya yaitu:

1. Akurasi data, *QA testing* membantu memastikan bahwa data yang ditampilkan di *website* ini akurat. Hal ini sangat penting karena data tersebut digunakan oleh masyarakat untuk membuat keputusan yang tepat terkait dengan kesehatan dan keselamatannya;
2. Keandalan *website*, untuk memastikan *website* bisa diakses tanpa kendala dengan mudah dan lancar;
3. Kegunaan *website*, *QA testing* membantu memastikan bahwa *website* kualitas air sungai Citarum mudah digunakan dan dinavigasi oleh semua pengguna. Hal ini penting untuk memastikan bahwa semua orang dapat mengakses informasi yang mereka butuhkan dengan mudah;
4. Keamanan Data, pada bagian ini pastinya *QA testing* membantu memastikan bahwa data yang tersimpan pada *website* aman dan tidak ada kebocoran data, guna untuk melindungi data pengguna dan menjaga integritas data yang ditampilkan[5].

Dengan memastikan *website* berfungsi dengan baik dan memberikan informasi yang akurat, *QA testing* membantu meningkatkan kepercayaan dan keterlibatan masyarakat terhadap *website* ini. Hal ini pada akhirnya akan berkontribusi pada upaya pelestarian Sungai Citarum.

II. KAJIAN TEORI

Dalam proses pengembangan perangkat lunak, berbagai jenis pengujian perlu dilakukan untuk memeriksa fungsionalitas dan kegunaan aplikasi perangkat lunak. Di antara pengujian tersebut, dua pengujian yang sangat penting adalah pengujian alpha dan beta.

A. Alpha Testing

Alpha testing bertujuan untuk mengidentifikasi bug atau kekurangan lebih awal yang terdapat pada *website*, sehingga akan menjadi lebih baik saat diluncurkan untuk publik. Alpha testing adalah pengujian yang dilakukan oleh user pada lingkungan pengembangan. Pengujian alpha berlangsung di situs pengembang oleh tim internal, sebelum rilis kepada pelanggan eksternal. Agar nantinya ketika pelanggan menggunakan system ini tidak kecewa karena masalah cacat atau kegagalan aplikasi. Pengujian ini dilakukan tanpa keterlibatan tim pengembangan. Selain itu, alpha testing sering digunakan untuk *software* sebagai bentuk testing penerimaan internal sebelum *software* menuju beta testing[6]. Pada *Alpha Testing* dilakukan beberapa pengujian fungsi pada setiap halaman seperti, *Main*, *Account*, Kalkulator, Maps, dan Penjelasan.

Pengujian ini melibatkan berbagai skenario penggunaan untuk memastikan bahwa setiap fitur berfungsi sesuai dengan yang diharapkan. Selain itu, tim pengembang juga melakukan pengujian terhadap responsivitas *website* pada berbagai perangkat dan ukuran layar. Umpan balik dari pengujian ini sangat penting untuk mengidentifikasi area yang memerlukan perbaikan atau peningkatan.

Proses pengujian ini juga membantu dalam mengidentifikasi masalah kompatibilitas dengan berbagai *browser*. Dengan demikian, *Alpha Testing* memastikan bahwa *website* siap untuk tahap *Beta Testing* dengan lebih sedikit masalah yang signifikan[7].

B. Beta Testing

Beta testing merupakan metode untuk memeriksa dan mengesahkan suatu *software*. *Beta testing* digunakan untuk menggambarkan proses testing external dimana *software* dapat diedarkan kepada orang lain seperti user yang berpotensi menggunakan *software* untuk kehidupan sehari-hari. *Beta testing* biasanya berpengaruh pada tahap akhir pengembangan *software* dan biasanya menjadi suatu pengesahan bahwa *software* sudah siap untuk digunakan oleh *user*. Tujuan dari *beta testing* dapat beraneka ragam, seperti kesempatan media pers untuk menuliskan masukan dari user untuk mengatasi bugs dan kesalahan yang ada.

Beta testing merupakan sebuah *testing* yang dimana sebuah perusahaan memberikan sebuah akses kepada *user* – *user* untuk memakainya dan juga banyak dari perusahaan tidak memberikan akses secara umum kepada *user*, melainkan perusahaan membayar user tersebut dalam artian kontrak selama lebih 1 bulan maupun setahun. *Beta testing* juga dilakukan untuk agar para *user* yang memakainya dapat memberikan info mengenai kerusakan ataupun error yang terjadi pada aplikasi yang dibuat oleh developer tersebut sehingga, dan juga laporan mengenai error atau kerusakan tersebut akan diterima selama kurang lebih sampai selesainya *beta testing*. Dengan begitu masalah yang terjadi pada aplikasi tersebut akan diperbaiki.

C. Perbandingan Pengujian Alpha Testing & Beta Testing

Membandingkan pengujian alpha dengan pengujian beta, pengujian alpha dilakukan tepat setelah pengembangan perangkat lunak, dimana perangkat lunak menjalani pemeriksaan internal menggunakan teknik pengujian kotak hitam dan kotak putih untuk menemukan potensi bug.

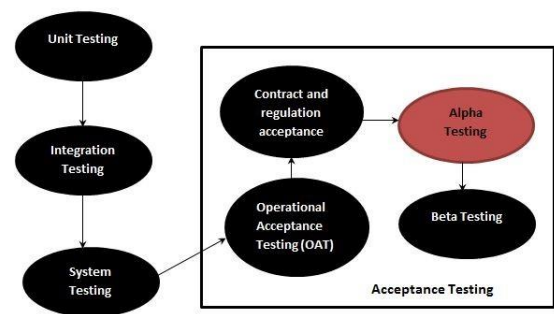
D. Stress Test

Stress Testing adalah salah satu performance test yang fokus untuk mengetes kekuatan, ketersediaan dan keandalan dari suatu aplikasi dalam kondisi yang ekstrim. Kondisi ekstrim tersebut di antaranya *heavy loads*, *high concurrency*, atau *limited computational resources*. Stress testing yang baik juga membantu dalam menemukan *bug* terkait sinkronisasi, *interlock problems*, *priority problems*, dan *resource loss bugs*[11].

III. METODE

A. Diagram Pengujian Alpha Testing

Pengujian alpha merupakan bagian dari pengujian penerimaan yang melibatkan pengujian aplikasi perangkat lunak selama *Software Development Life Cycle*. Pengujian ini dijalankan oleh karyawan internal organisasi yang memahami aplikasi perangkat lunak dan fungsinya. Pengujian alpha sering kali dilakukan dalam lingkungan yang terkendali untuk menangkap bug atau masalah yang mungkin terlewatkan selama pengembangan. Tujuan utamanya adalah untuk memastikan bahwa aplikasi memenuhi kebutuhan bisnis dan berfungsi sesuai harapan sebelum diluncurkan ke pengguna akhir. Pengujian ini juga membantu dalam mengidentifikasi masalah kinerja atau stabilitas yang mungkin timbul. Hasil dari pengujian alfa digunakan untuk melakukan perbaikan dan penyesuaian lebih lanjut sebelum melanjutkan ke tahap pengujian beta. Berikut adalah diagram yang menjelaskan proses Alpha test dalam siklus pengembangan *software*:



Gambar 3.1
Diagram proses Alpha Testing

Pada tahap pertama dari pengujian alpha, perangkat lunak diuji oleh pengembang secara *in-house* di mana tujuannya adalah untuk menangkap bug dengan cepat. Perangkat lunak yang diuji sebenarnya sudah diuji pada tahap unit test, maupun system test. Pada bagian akhir perangkat lunak tersebut diuji pada lingkungan yang sebenarnya tetapi masih dilingkungan developer. Lingkungan dibuat sedemikian rupa agar menyerupai lingkungan sebenarnya. Hal ini dilakukan agar seluruh modul yang ada dalam perangkat lunak tersebut berjalan sesuai aktifitas dan lingkungan sebenarnya. Pada

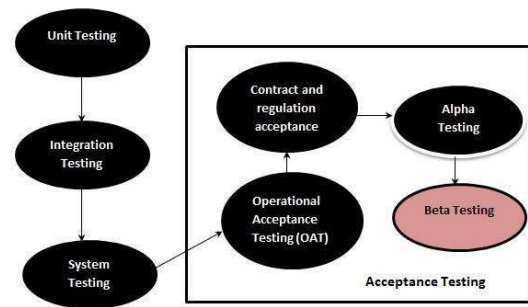
fase kedua pengujian alpha, perangkat lunak diberikan kepada tim QA perangkat lunak untuk pengujian tambahan.

1. Waktu Pengujian

- a. Tim pengembangan telah menyelesaikan pengkodean awal dan membangun fitur dasar. Pengujian alpha memungkinkan untuk melihat fungsionalitas keseluruhan terlebih dahulu sebelum penyempurnaan lebih lanjut.
 - b. Fungsi inti yang menentukan tujuan perangkat lunak harus dibangun dan berfungsi. Pengujian alpha membantu memvalidasi fungsi-fungsi ini dan memastikan fungsi tersebut berfungsi sebagaimana mestinya.
 - c. Lingkungan pengujian yang terkendali, terpisah dari lingkungan pengguna akhir, harus dibuat. Hal ini memungkinkan pengujian untuk fokus pada fungsi tertentu dan membuat ulang skenario untuk identifikasi masalah yang efisien.
 - d. Sekelompok pengujian internal khusus, seperti pengembang atau pengujian, harus tersedia untuk memberikan umpan balik selama pengujian alpha.
2. Proses Pengujian
- a. Memeriksa aplikasi perangkat lunak pada tingkat granular untuk memastikan keandalannya. Fokusnya kepada bagian *whitebox testing*, yang memeriksa struktur internal sistem untuk memverifikasi kode dan algoritma pembelajaran mesin
 - b. Penyerahan perangkat lunak kepada pengujian untuk pengujian lebih lanjut dalam lingkungan yang menyerupai pengaturan produksi. Disini, fokus utamanya adalah pengujian *black-box* yaitu memeriksa kinerja sistem tanpa mempertimbangkan struktur internal, seperti kemampuan sistem untuk mengklasifikasikan kualitas air dengan sangat akurat dan presisi[8].

B. Diagram Pengujian *Beta Testing*

Pengujian beta dikenal sebagai pengujian pengguna berlangsung di lokasi pengguna akhir oleh pengguna akhir untuk memvalidasi kegunaan, fungsi, kompatibilitas, dan uji reliabilitas dari *software* yang dibuat. Pengujian ini memungkinkan pengembang mendapatkan umpan balik nyata dari pengguna yang sesungguhnya menggunakan perangkat lunak dalam lingkungan mereka sehari-hari. Selain itu, pengujian beta dapat mengidentifikasi masalah yang mungkin tidak terdeteksi selama pengujian internal. Hasil dari pengujian ini sangat berharga untuk memastikan produk akhir memenuhi harapan dan kebutuhan pengguna. Selanjutnya, ini membantu memperbaiki bug dan meningkatkan performa sebelum peluncuran resmi. Pengujian ini juga berfungsi sebagai bentuk promosi awal untuk membangun antusiasme di antara pengguna potensial. [9].



Gambar 3.2 Diagram proses *Beta Testing*

Tujuan dari pengujian beta adalah untuk menempatkan aplikasi ditangan pengguna yang sebenarnya yang berada di luar tim teknik Guna untuk menemukan setiap kekurangan atau masalah dari perspektif pengguna akhir[10].

1. Waktu Pengujian

Waktu untuk pengujian beta testing yaitu dilakukan ketika software sudah melewati testing sebelumnya dan apabila ada kesalahan semuanya sudah diperbaiki. Dengan hal ini menunjukkan bahwa software tersebut sudah siap untuk digunakan oleh user.

2. Proses Pengujian

Pengujian beta ini dilakukan dalam bentuk kuesioner melalui *Google Form* dengan total responden sebanyak 81 orang. Responden atau target pengguna *website* ini adalah masyarakat umum, khususnya mahasiswa yang rumahnya dekat dengan sungai. Pengujian dilakukan dengan menggunakan skala Likert yang terdiri dari 5 tingkatan, masing-masing memiliki nilai yang berbeda. Persentase ini memberikan gambaran kuantitatif tentang distribusi tanggapan responden terhadap setiap tingkat pada skala likert. Dengan menghitung persentase untuk setiap kategori tanggapan bertujuan untuk memahami proporsi responden yang sangat setuju, setuju, netral, kurang setuju, dan sangat tidak setuju dengan pertanyaan yang dilakukan dalam kuesioner. Adapun rumus yang digunakan yaitu:

$$\text{Persentase} = \frac{\text{Jumlah frekuensi jawaban} \times \text{skala jawaban}}{\text{Jumlah jawaban tertinggi} \times \text{jumlah sampel}} \times 100\%$$

IV. HASIL DAN PEMBAHASAN

A. *Alpha Testing*

Alpha testing bertujuan untuk mengidentifikasi bug atau kekurangan lebih awal yang terdapat pada *website*, sehingga akan menjadi lebih baik saat diluncurkan untuk publik. Pada *Alpha Testing* dilakukan beberapa pengujian fungsi pada setiap halaman seperti, *Main*, *Account*, Kalkulator, Maps, dan Penjelasan.

TABEL 1
Alpha Testing Main

No	Deskripsi Pengujian	Hasil yang Diperoleh	Status
1	Mengakses <i>website</i> melalui link https://citasii.streamlit.a pp/	<i>Website</i> akan menampilkan halaman kalkulator manual	<i>Pass</i>

2	Menekan menu <i>Account</i>	Menampilkan halaman login <i>admin</i>	<i>Pass</i>
3	Menekan menu Kalkulator	Menampilkan halaman kalkulator manual	<i>Pass</i>
4	Menekan menu <i>Maps</i>	Menampilkan keterangan warna pada <i>maps</i> dan tampilan awal peta beserta titik persebarannya	<i>Pass</i>
5	Menekan menu Penjelasan	Menampilkan penjelasan mengenai Website Citasi	<i>Pass</i>

TABEL 2
Alpha Testing Account

No	Deskripsi Pengujian	Hasil yang Diperoleh	Status
1	Menekan menu <i>Account</i>	Menampilkan halaman <i>login admin</i>	<i>Pass</i>
2	Menekan opsi pegawai atau <i>admin</i>	Menampilkan halaman <i>login admin</i>	<i>Pass</i>
3	Menekan menu Kalkulator	Menampilkan halaman <i>login pegawai</i>	<i>Pass</i>
4	Memasukkan <i>email</i> dan <i>password</i> yang salah pada <i>login admin</i>	<i>Login</i> akan gagal dan akan menampilkan notifikasi ' <i>Login Failed: Invalid Admin Credentials</i> '	<i>Pass</i>
5	Memasukkan <i>email</i> dan <i>password</i> yang benar pada <i>login admin</i>	Akan masuk ke dalam halaman verifikasi anggota pegawai	<i>Pass</i>
6	Menekan <i>Sign Out</i> pada akun <i>admin</i>	Berhasil melakukan <i>sign out</i> dan kembali pada halaman <i>login admin</i>	<i>Pass</i>
7	Memilih <i>login</i> atau <i>sign up</i> pada halaman <i>login pegawai</i>	Akan menampilkan halaman <i>login pegawai</i>	<i>Pass</i>
8		Akan menampilkan halaman <i>sign up pegawai</i>	<i>Pass</i>

9	Memasukkan data pada halaman <i>sign up</i> pegawai dan menekan <i>create account</i>	Berhasil melakukan <i>sign up</i> lalu menampilkan notifikasi 'Akun berhasil dibuat. Silakan tunggu verifikasi dari <i>admin</i> .' dan 'Email verifikasi berhasil dikirim'	<i>Pass</i>
10	Memasukkan data yang sudah terdaftar pada halaman <i>sign up</i> pegawai	<i>Sign up</i> akan gagal dan akan muncul notifikasi 'Pembuatan akun gagal: Email sudah terdaftar'	<i>Pass</i>
11	Memasukkan <i>email</i> dengan tidak sesuai pada halaman <i>sign up</i> pegawai	<i>Sign up</i> akan gagal dan akan muncul notifikasi 'Pembuatan akun gagal: <i>Malformed email address string: ""</i> .'	<i>Pass</i>
12	Mengosongkan halaman <i>sign up</i> pegawai dan menekan <i>create account</i>	<i>Sign up</i> akan gagal dan akan muncul notifikasi 'Pembuatan akun gagal: <i>Invalid email: ""</i> . <i>Email must be a non-empty string</i> .'	<i>Pass</i>
13	Mengosongkan halaman login pegawai dan menekan <i>login</i>	<i>Login</i> akan gagal dan akan muncul notifikasi 'Pembuatan akun gagal: <i>Invalid email: ""</i> . <i>Email must be a non-empty string</i> .'	<i>Pass</i>
14	Memasukkan <i>email</i> dan <i>password</i> yang belum terverifikasi pada halaman <i>login pegawai</i>	<i>Login</i> akan gagal dan akan menampilkan notifikasi 'Akun Anda belum diverifikasi.'	<i>Pass</i>
15	Memasukkan <i>email</i> dan <i>password</i> yang sudah terverifikasi pada halaman <i>login pegawai</i>	<i>Login</i> akan berhasil dan akan menampilkan halaman nama akun pegawai	<i>Pass</i>

16	Menekan <i>Sign Out</i> pada akun pegawai	Berhasil keluar dan akan kembali pada halaman <i>login</i> pegawai	<i>Pass</i>
----	-------------------------------------------	--------------------------------------------------------------------	-------------

TABEL 3
Alpha Testing Kalkulator

No	Deskripsi Pengujian	Hasil yang Diperoleh	Status
1	Menekan menu kalkulator	Menampilkan halaman kalkulator manual	<i>Pass</i>
2	Menekan buku panduan pengguna	Menampilkan jendela yang berisikan <i>link</i> eksternal	<i>Pass</i>
3	Menekan <i>link</i> eksternal	Akan menampilkan buku panduan untuk pengguna	<i>Pass</i>
4	Memilih metode pembelajaran mesin yang disediakan	Hasil kalkulator akan menjadi sesuai dengan yang dipilih	<i>Pass</i>
5	Memilih penginputan data	Menampilkan halaman kalkulator manual	<i>Pass</i>
6	Menekan <i>Sign Out</i> pada akun <i>admin</i>	Menampilkan halaman kalkulator <i>upload file</i>	<i>Pass</i>
7	Tidak mengisikan data dan menekan klasifikasi pada kalkulator manual	Akan memunculkan notifikasi 'Mohon pastikan semua nilai sudah dimasukkan dengan benar Dan dalam format numerik.' dan 'Tidak boleh ada kolom yang kosong yes!'	<i>Pass</i>
8	Memasukkan data non numerik pada kalkulator manual	Akan memunculkan notifikasi 'Mohon pastikan semua nilai sudah dimasukkan dengan benar Dan dalam format numerik.' dan 'Tidak boleh ada kolom yang kosong yes!'	<i>Pass</i>

9	Memasukkan data dengan benar pada kalkulator manual	Menampilkan hasil klasifikasi, penjelasan kelas, grafik metode pembelajaran yang dipilih, dan penjelasan tambahan	<i>Pass</i>
10	Memasukkan file tidak lengkap pada Kalkulator <i>upload file</i>	Klasifikasi akan gagal dan memunculkan notifikasi 'Dataset tidak memiliki semua fitur yang dibutuhkan. Pastikan kolom BOD, COD, <i>FecalColiform</i> , dan IP ada dalam dataset.	<i>Pass</i>
11	Memasukkan file yang benar pada kalkulator <i>upload file</i>	Menampilkan tabel hasil klasifikasi dan <i>pie chart</i> hasil klasifikasi	<i>Pass</i>
12	Mengunduh hasil klasifikasi	Berhasil mengunduh dan akan tersimpan pada <i>file</i> komputer	<i>Pass</i>

TABEL 4
Alpha Testing Maps

No	Deskripsi Pengujian	Hasil yang Diperoleh	Status
1	Menekan menu <i>Maps</i>	Menampilkan keterangan warna pada <i>Maps</i> dan tampilan awal peta beserta titik persebarannya	<i>Pass</i>
2	Menekan buku panduan pengguna	Menampilkan jendela yang Berisikan <i>link</i> eksternal	<i>Pass</i>
3	Menekan <i>link</i> eksternal	Akan menampilkan buku panduan untuk pengguna	<i>Pass</i>
4	Memilih metode pembelajaran mesin yang disediakan	Akan menampilkan peta tanpa pembelajaran mesin	<i>Pass</i>
		Akan menampilkan peta dengan pembelajaran mesin <i>Weighted KNN</i>	<i>Pass</i>

		Akan menampilkan peta dengan pembelajaran mesin ANN	Pass
		Akan menampilkan peta dengan pembelajaran mesin GNB	Pass
5	Menceklis kontrol panel pada tampilan awal peta	Menampilkan jalur sungai Citarum	Pass
		Menampilkan jalur waduk	Pass
		Menampilkan daerah aliran sungai	Pass
6	Menekan salah satu titik pada sungai	Hanya menampilkan nama titik sungainya saja	Pass
		Menampilkan nama dan alamat sungai serta status sesuai dengan <i>Weighted KNN</i>	Pass
		Menampilkan nama dan alamat sungai serta status sesuai dengan <i>Artificial Neural Network</i>	Pass

TABEL 5
Alpha Testing Penjelasan

No	Deskripsi Pengujian	Hasil yang Diperoleh	Status
1	Menekan menu Penjelasan	Menampilkan penjelasan mengenai Website Citasi dan <i>video</i> penjelasan	Pass
2	Menekan <i>video</i>	Akan memutar <i>video</i> pengujian <i>website</i> Citasi	Pass
3	Menekan Baca Selengkapn ya <i>Weighted KNN</i>	Akan menampilkan <i>website</i> https://www.ibm.com/topics/knn yang menjelaskan KNN lebih dalam	Pass
4	Menekan Baca Selengkapn ya <i>Artificial Neural Network</i>	Akan menampilkan <i>website</i> https://www.analyticsvi dya.com/blog/2021/09/introduction-	Pass

		to-artificial-neural-networks/ yang menjelaskan ANN lebih dalam	
5	Menekan Baca Selengkapn ya <i>Gaussian Naive Bayes</i>	Akan menampilkan <i>website</i> https://builtin.com/artifi cial- intelligence/gau ssia n-naive- bayes yang menjelaskan GNB lebih dalam	Pass
6	Menekan Tempat Mendapatka n <i>Dataset</i>	Akan menampilkan <i>website</i> https://opendata .jabarpr ov.go.id/id	Pass
7	Menekan Baca Selengkapn ya Ph	Akan menampilkan <i>website</i> https://www.us gs.gov/s pecial- topics/water- scie nce- school/science/ ph-a nd- water#: ~:text=o il%20or%20alc ohol.-,pH%2 0is%20a%20me asure% 20of%20how% 20acidic %2Fbasic%20w ater%2 0is,hydroxyl%2 0ions% 20in%20the%2 0water. yang menjelaskan Ph lebih dalam	Pass
8	Menekan Baca Selengkapn ya Tss	Akan menampilkan <i>website</i> https://www.ha ndal- selar as.com/total- suspended- solids- tss/ yang menjelaskan TSS lebih dalam	Pass
9	Menekan Baca Selengkapn ya DO	Akan menampilkan <i>website</i> https://www.ep a.gov/na tional- aquatic- resourcesurveys /indicators- dissol ved-	Pass

TABEL 8
Beta Testing 2

Pertanyaan	Jawaban	Nilai	Frekuensi	Total Nilai
1	Sangat Tidak Setuju	1	1	1
	Kurang Setuju	2	1	2
	Netral	3	14	42
	Setuju	4	40	160
	Sangat Setuju	5	26	130
Jumlah			81	335
Persentase = $\frac{336}{405} \times 100\% = 82.72\%$				

3. Antarmuka pengguna *website* Citasi mudah digunakan dan intuitif untuk melakukan klasifikasi kualitas air sungai Citarum

TABEL 9
Beta Testing 2

Pertanyaan	Jawaban	Nilai	Frekuensi	Total Nilai
1	Sangat Tidak Setuju	1	0	0
	Kurang Setuju	2	1	2
	Netral	3	11	33
	Setuju	4	34	136
	Sangat Setuju	5	35	175
Jumlah			81	346
Persentase = $\frac{346}{405} \times 100\% = 85.43\%$				

4. Hasil klasifikasi dari model machine learning membantu dalam memahami kualitas air sungai Citarum

TABEL 10
Beta Testing 4

Pertanyaan	Jawaban	Nilai	Frekuensi	Total Nilai
1	Sangat Tidak Setuju	1	0	0
	Kurang Setuju	2	0	0
	Netral	3	9	27
	Setuju	4	37	148
	Sangat Setuju	5	35	175
Jumlah			81	350
Persentase = $\frac{350}{405} \times 100\% = 86.42\%$				

5. Data yang digunakan untuk klasifikasi kualitas air sungai Citarum cukup akurat dan terpercaya.

TABEL 10
Beta Testing 5

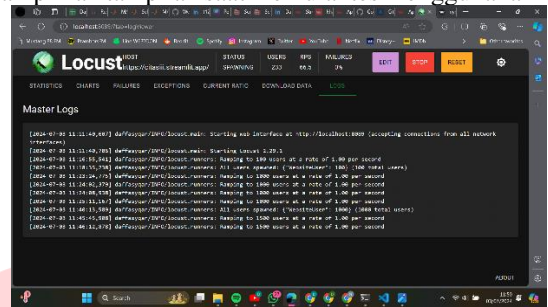
Pertanyaan	Jawaban	Nilai	Frekuensi	Total Nilai
1	Sangat Tidak Setuju	1	0	0
	Kurang Setuju	2	0	0
	Netral	3	19	57
	Setuju	4	34	136
	Sangat Setuju	5	28	140
Jumlah			81	333
Persentase = $\frac{333}{405} \times 100\% = 82.22\%$				

6. Secara keseluruhan, saya puas dengan performa *website* Citasi dalam melakukan klasifikasi kualitas air sungai Citarum

TABEL 11
Beta Testing 6

Pertanyaan	Jawaban	Nilai	Frekuensi	Total Nilai
1	Sangat Tidak Setuju	1	0	0
	Kurang Setuju	2	1	2
	Netral	3	19	57
	Setuju	4	34	136
	Sangat Setuju	5	28	140
Jumlah			81	335
Persentase = $\frac{335}{405} \times 100\% = 82.72\%$				

detik, waktu respon, dan jumlah pengguna aktif. Selain itu, informasi mengenai permintaan yang berhasil dan gagal dapat diamati untuk memahami kemampuan aplikasi dalam menangani beban tinggi. Dengan menggunakan dashboard ini, evaluasi terhadap kapasitas dan stabilitas aplikasi dapat dilakukan secara komprehensif. Gambar dibawah menampilkan tampilan saat memulai tes menggunakan locust



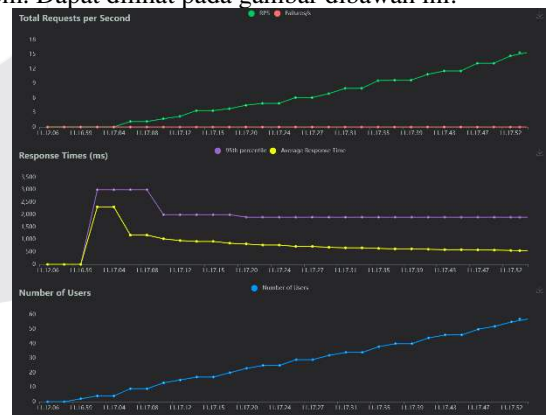
Gambar 4.2
Tampilan saat memulai tes menggunakan locust

Locust memungkinkan pengguna membuat skrip simulasi perilaku pengguna untuk mengevaluasi kinerja aplikasi di bawah beban. Halaman web ini menampilkan log utama dari pengujian beban, termasuk jumlah permintaan, waktu respons rata-rata, dan tingkat kesalahan, yang membantu mengidentifikasi masalah kinerja.

5. Memulai tes

Pada tahap memulai tes menggunakan locust dilakukan beberapa kali percobaan, dengan percobaan pertama yaitu a. Hasil *stress test* untuk 100 User

Hasil *stress test* untuk 100 pengguna menunjukkan bahwa sistem mampu menangani peningkatan beban dengan baik. Grafik permintaan per detik menunjukkan tidak ada kegagalan permintaan, dengan sistem berhasil menangani semua permintaan. Grafik waktu respon menunjukkan adanya peningkatan tajam pada awal pengujian, namun kemudian menurun dan stabil. Jumlah pengguna yang meningkat selama pengujian juga ditangani dengan baik oleh sistem. Dapat dilihat pada gambar dibawah ini.



GAMBAR 4.3
Hasil *stress test* untuk 150 user

Grafik di atas menunjukkan hasil dari *stress test* untuk 100 user. *Stress test* ini mengukur tiga metrik utama: Berikut adalah penjelasan masing-masing grafik:

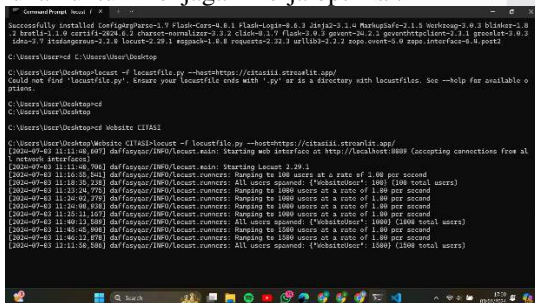
1. *Total Requests per Second* (RPS)
 - Grafik ini menunjukkan jumlah permintaan per detik yang diterima oleh sistem.
 - Garis Hijau (RPS): Menunjukkan jumlah permintaan per detik yang berhasil ditangani oleh sistem.

C. *Stress Test*

Pengujian ini dilakukan bertujuan untuk mengevaluasi kinerja dan stabilitas situs website. Pengujian ini membantu memahami batasan situs yang telah dibuat. Adapun langkah yang dilakukan untuk melakukan stress test yaitu:

1. Install Locust menggunakan pip dengan perintah “pip install locust”
2. Membuat file test
3. Menjalankan locust

Stress testing adalah metode penting untuk mengetahui bagaimana situs web menangani lonjakan lalu lintas yang tiba-tiba. Dengan melakukan pengujian ini, bisa mengidentifikasi potensi bottleneck dan area yang memerlukan peningkatan. Selain itu, pengujian ini juga memungkinkan untuk memastikan bahwa situs web tetap responsif dibawah beban berat, sehingga pengalaman pengguna tidak terganggu. Dengan pemahaman yang lebih baik tentang batasan sistem, merencanakan peningkatan yang diperlukan untuk menjaga kinerja optimal.



GAMBAR 4.1
Tampilan menjalankan locust

4. Akses *Dashboard locust*

Dashboard ini menyediakan visualisasi yang mendetail tentang berbagai metrik kinerja seperti jumlah permintaan per

- Garis Merah (Failures/s): Menunjukkan jumlah permintaan per detik yang gagal diproses oleh sistem. Terlihat bahwa tidak ada permintaan yang gagal selama stress test ini.

2. Response Times (ms)

- Grafik ini menunjukkan waktu respon dari sistem.

- Garis Ungu (95th percentile): Menunjukkan waktu respon pada persentil ke-95, artinya 95% dari permintaan memiliki waktu respon di bawah nilai ini.

- Garis Kuning (Average Response Time): Menunjukkan waktu respon rata-rata dari sistem.

- Dari grafik terlihat bahwa waktu respon meningkat tajam pada awal test (sekitar 11:17:00) dan kemudian menurun dan stabil.

3. Number of Users

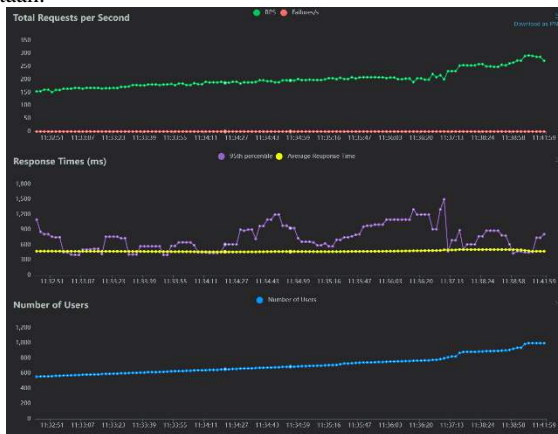
- Grafik ini menunjukkan jumlah pengguna yang menggunakan sistem pada setiap waktu.

- Garis Biru: Menunjukkan jumlah pengguna yang terus meningkat selama stress test.

Hasil *stress test* ini menunjukkan bahwa sistem mampu menangani peningkatan beban dengan baik, meskipun ada peningkatan waktu respon pada awalnya yang kemudian stabil.

b. Hasil stress test untuk 1.000 User

Hasil *stress test* untuk 1.000 user menunjukkan bahwa sistem mampu menangani beban yang meningkat dengan baik. Grafik menunjukkan jumlah permintaan per detik yang stabil di sekitar 250 RPS tanpa adanya permintaan yang gagal, menunjukkan keandalan sistem. Meskipun ada fluktuasi pada waktu respon persentil ke-95, waktu respon rata-rata tetap konstan sekitar 600-700 ms. Jumlah pengguna meningkat secara bertahap hingga 1000 pengguna selama pengujian, dan sistem tetap berfungsi dengan baik tanpa kegagalan permintaan.



Gambar 4.4
Hasil *stress test* untuk 1000 user

Grafik di atas menunjukkan hasil dari *stress test* dengan total pengguna mencapai 1000. Berikut adalah penjelasan masing-masing grafik:

1. Total Requests per Second (RPS)

- Garis Hijau (RPS): Menunjukkan jumlah permintaan per detik yang berhasil ditangani oleh sistem.

- Garis Merah (Failures/s): Menunjukkan jumlah permintaan per detik yang gagal diproses oleh sistem. Dari grafik ini, terlihat tidak ada permintaan yang gagal, menunjukkan bahwa sistem mampu menangani semua permintaan.

- Jumlah permintaan per detik meningkat secara bertahap dan mencapai puncaknya sekitar 250 RPS, kemudian stabil pada tingkat tersebut.

2. Response Times (ms)

- Garis Ungu (95th percentile): Menunjukkan waktu respon pada persentil ke-95. Ada fluktuasi signifikan dalam waktu respon, terutama setelah 11:34:27.

- Garis Kuning (Average Response Time): Menunjukkan waktu respon rata-rata dari sistem. Waktu respon rata-rata cukup stabil

sekitar 600-700 ms, tetapi ada lonjakan dan penurunan pada waktu respon persentil ke-95.

- Ada beberapa lonjakan pada waktu respon persentil ke-95, tetapi rata-rata waktu respon tetap relatif konstan.

3. Number of Users

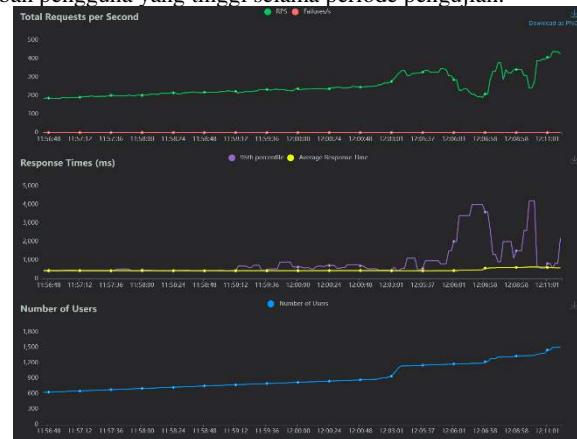
- Garis Biru: Menunjukkan jumlah pengguna yang menggunakan sistem pada setiap waktu.

- Jumlah pengguna meningkat secara bertahap dari 0 hingga 1000 pengguna selama periode stress test ini.

Hasil *stress test* ini menunjukkan bahwa sistem mampu menangani beban yang meningkat dengan baik, meskipun ada beberapa fluktuasi pada waktu respon persentil ke-95. Sistem dapat menangani hingga 1000 pengguna tanpa adanya permintaan yang gagal, yang menunjukkan kinerja yang baik di bawah beban tinggi.

c. Hasil stress test untuk 1.500 User

Hasil *stress test* menunjukkan bahwa sistem mampu menangani peningkatan jumlah permintaan per detik hingga sekitar 400 RPS tanpa adanya kegagalan proses. Jumlah permintaan per detik meningkat secara bertahap dengan beberapa fluktuasi di sekitar pukul 12:03 hingga 12:07. Waktu respon rata-rata cukup stabil antara 400-600 ms, meskipun ada beberapa lonjakan signifikan pada waktu respon persentil ke-95, terutama setelah pukul 12:03:01. Jumlah pengguna meningkat secara bertahap hingga mencapai 1500 pengguna, menunjukkan kemampuan sistem dalam menangani beban pengguna yang tinggi selama periode pengujian.



Gambar 4.5
Hasil *stress test* untuk 1.500 user

Grafik di atas menunjukkan hasil dari stress test dengan total pengguna mencapai 1.500. Berikut adalah penjelasan masing-masing grafik:

1. Total Requests per Second (RPS)

- Garis Hijau (RPS): Menunjukkan jumlah permintaan per detik yang berhasil ditangani oleh sistem.

- Garis Merah (Failures/s): Menunjukkan jumlah permintaan per detik yang gagal diproses oleh sistem. Dari grafik ini, terlihat tidak ada permintaan yang gagal, menunjukkan bahwa sistem mampu menangani semua permintaan.

- Jumlah permintaan per detik meningkat secara bertahap dan mencapai puncaknya sekitar 400 RPS sebelum stabil pada tingkat tersebut. Namun, terdapat fluktuasi di sekitar 12:03 hingga 12:07 sebelum stabil kembali.

2. Response Times (ms)

- Garis Ungu (95th percentile): Menunjukkan waktu respon pada persentil ke-95. Ada fluktuasi signifikan dalam waktu respon, terutama setelah 12:03:01.

- Garis Kuning (Average Response Time): Menunjukkan waktu respon rata-rata dari sistem. Waktu respon rata-rata cukup stabil sekitar 400-600 ms, tetapi ada lonjakan dan penurunan pada waktu respon persentil ke-95.

- Ada beberapa lonjakan pada waktu respon persentil ke-95 yang cukup signifikan, terutama setelah 12:03:01, dengan beberapa waktu respon yang mencapai hampir 5000 ms.

3. Number of Users

-Garis Biru: Menunjukkan jumlah pengguna yang menggunakan sistem pada setiap waktu.

-Jumlah pengguna meningkat secara bertahap dari 0 hingga 1500 pengguna selama periode stress test ini.

Secara keseluruhan, hasil stress test ini menunjukkan bahwa sistem mampu menangani beban yang meningkat dengan baik hingga 1500 pengguna, meskipun ada beberapa fluktuasi signifikan pada waktu respon persentil ke-95 selama periode tertentu. Sistem dapat menangani hingga 400 RPS tanpa adanya permintaan yang gagal, yang menunjukkan kinerja yang baik di bawah beban tinggi, meskipun perlu ada perhatian pada peningkatan waktu respon untuk beberapa permintaan.

V. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan dalam pengujian hasil *alpha testing website* menunjukkan bahwa sistem berfungsi sesuai harapan. Fitur utama seperti kalkulator dan maps berhasil menampilkan data dengan akurat sesuai dengan metode pembelajaran mesin yang digunakan. Selain itu, fitur *account* dan penjelasan juga sudah berfungsi dengan baik, memungkinkan admin dan pegawai untuk mengelola akun serta memberikan informasi tambahan mengenai pembelajaran mesin dan kualitas air. Selanjutnya, pada pengujian *betha testing* dengan menghitung persentase untuk setiap kategori tanggapan, dapat dipahami distribusi persepsi responden terhadap fitur-fitur *website* yang diuji, khususnya dalam hal klasifikasi kualitas air Sungai Citarum.

Berdasarkan hasil pengujian menggunakan skala Likert, mayoritas responden (rata-rata sekitar 82-86%) menunjukkan tingkat kepuasan yang tinggi terhadap performa model *machine learning*, kemudahan penggunaan antarmuka, serta akurasi dan keandalan data yang digunakan, mencerminkan kepercayaan mereka terhadap kehandalan informasi yang disajikan oleh *website*. Dan yang terakhir adanya *stress test* menggunakan locust yang bertujuan untuk mengetahui batasan maksimum kapasitas dari kinerja *website* saat dikenakan beban tinggi. Dalam percobaan ini dilakukan tiga kali yaitu 100 user, 1000 user, dan 1500 user. Hasilnya *website* mampu menampung bahkan saat percobaan untuk 1500 user, meskipun beberapa beberapa fluktuasi signifikan pada waktu respon persentil ke-95 selama periode tertentu. Sistem dapat menangani hingga 400 RPS tanpa adanya permintaan yang gagal, yang menunjukkan kinerja yang baik di bawah beban tinggi, meskipun perlu ada perhatian pada peningkatan waktu respon untuk beberapa permintaan.

REFERENSI

- [1] M. Marselina, F. Wibowo, and A. Mushfiroh, "Water quality index assessment methods for surface water: A case study of the Citarum River in Indonesia," *Heliyon*, vol. 8, no. 7, Jul. 2022, doi: 10.1016/j.heliyon.2022.e09848.
- [2] "Sekilas Citarum Kondisi Fisik dan Spasial," Cita-Citarum. Diakses: 8 Oktober 2023. [Daring]. Tersedia pada: <http://citarum.org/tentang-kami/sekilas-citarum/kondisi-fisik-dan-spasial.html>
- [3] R. Desriyan, "Identifikasi pencemaran logam berat timbal (Pb) pada perairan Sungai Citarum Hulu segmen Dayeuhkolot sampai Nanjung," *Jurnal Reka Lingkungan*, vol. 3, no. 1, hlm. 41–52, 2015.
- [4] Arkatama. (n.d.). Peran Penting Quality Assurance dalam Software Tester. Retrieved from <https://arkatama.id/peran-penting-quality-assurance-dalam-software-tester/>.
- [5] Robyanti, Z. F. (2021). "Pencemaran Sungai Citarum dan Tanggung Jawab Sosial Perusahaan." Diakses dari \[Online\]. Available: <https://kumparan.com/zahra-fani-robyanti/pencemaran-sungai-citarum-dan-tanggung-jawab-sosial-perusahaan-1urHoN1LdOb>
- [6] BINUS University. (2020, June 30). Alpha dan beta testing. BINUS University School of Computer Science. <https://socs.binus.ac.id/2020/06/30/alpha-dan-beta-testing/>.
- [7] RevoU. (2024). Apa itu UAT (User Acceptance Testing)? Pengertian dan contoh. Retrieved from <https://revou.co/kosakata/uat>.
- [8] "Alpha Testing," BINUS University, 16 Dec. 2016. [Online]. Available: <https://sis.binus.ac.id/2016/12/16/alpha-testing/>. [Accessed: 01-Jul-2024].
- [9] "Beta Test," SIS Binus, Dec. 16, 2016. [Online]. Available: <https://sis.binus.ac.id/2016/12/16/beta-test/>. [Accessed: Jul. 1, 2024].
- [10] Lambdatest, "Alpha Testing vs Beta Testing," [Online]. Available: <https://www.lambdatest.com/blog/alpha-testing-vs-beta-testing/>. [Accessed: 01-Jul-2024].
- [11] I. Hidayat, "Stress Testing Web Applications," *Medium*, Apr. 16, 2020. [Online]. Available: <https://medium.com/ppl-a1-iebs/stress-testing-web-applications-82ed156acd67>. [Accessed: Jul. 4, 2024].
- [12] Q. Hasanah, H. Oktavianto, dan Y. D. Rahayu, "Analisis Algoritma Gaussian Naive Bayes Terhadap Klasifikasi Data Pasien Penderita Gagal Jantung," *Jurnal Smart Teknologi*, vol. 3, no. 4, 2022.
- [13] E. A. W. Sanad, "Pemanfaatan Realtime Database di Platform Firebase Pada Aplikasi E-Tourism

Kabupaten Nabire,” Jurnal Penelitian Enjiniring, vol. 22, no. 1, 2019, doi: 10.25042/jpe.052018.04.

- [14] R. Hamidi, M. T. Furqon, dan B. Rahayudi, “Implementasi Learning Vector Quantization (LVQ) untuk Klasifikasi Kualitas Air Sungai,” Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 1, no. 12, 2017.
- [15] [1] InMotion Hosting, "How to Stress Test Your Website," InMotion Hosting, 2023. [Online].

Available:

<https://www.inmotionhosting.com/support/server/server-usage/how-to-stress-test-your-website/>.
[Accessed: 04-Jul-2024].

