

Website Development for IoT-Based Clean Water Reservoir Control System at Bandung City View I

Pengembangan Website untuk Sistem Pengontrolan Reservoir Air Bersih Berbasis IoT pada Bandung City View I

Muhammad Fharist¹, Faisal Candrasyah Hasibuan², Purba Daru Kusuma³

^{1,2,3} Teknik Komputer, Universitas Telkom, Indonesia

^{1*} muhammadfharist1203@gmail.com, ² faicanhasfcb@telkomuniversity.ac.id, ³ purbodaru@telkomuniversity.ac.id

*: Penulis korespondensi (corresponding author)

Informasi Artikel

Received: January 2025

Revised: February 2025

Accepted: June 2025

Published: June 2025

Abstract

Purpose: This research aims to overcome the problem of overflowing water in the upper reservoir in Bandung City View I (BCV I) Housing, developed an IoT-based water reservoir control system that can operate continuously for 24 hours.

Design/methodology/approach: This system uses secure hardware and can be accessed through web and mobile applications to facilitate the process of monitoring and controlling water reservoirs remotely and in real-time. Website development for this system involves the use of HTML, CSS, and JavaScript on the frontend, and Laravel framework on the backend. Firebase Realtime Database is used for data storage and real-time synchronization.

Findings/result: System testing shows that the developed website can effectively control pumps and monitor reservoir conditions, reduce water wastage, and lower operational costs.

Originality/value/state of the art: These results indicate that an IoT-based reservoir control system accessed through a website can be an efficient and effective solution in managing water resources in BCV I housing.

Abstrak

Keywords: water control; real-time; website

Kata kunci: pengontrolan air; waktu nyata; website

Tujuan: Penelitian ini bertujuan untuk mengatasi masalah meluapnya air di reservoir atas di Perumahan Bandung City View I (BCV I), dikembangkan sebuah sistem pengontrolan reservoir air berbasis IoT yang dapat beroperasi secara kontinu selama 24 jam.

Perancangan/metode/pendekatan: Sistem ini menggunakan perangkat keras yang aman dan dapat diakses melalui web dan aplikasi *mobile* untuk mempermudah proses pemantauan dan pengontrolan reservoir air secara jarak jauh dan *real-time*. Pengembangan *website* untuk sistem ini melibatkan penggunaan HTML, CSS, dan JavaScript di bagian *frontend*, serta Laravel *framework* di bagian *backend*. Firebase Realtime Database digunakan untuk penyimpanan data dan sinkronisasi *real-time*.

Hasil: Pengujian sistem menunjukkan bahwa *website* yang dikembangkan dapat mengontrol pompa dan memonitor kondisi reservoir dengan efektif, mengurangi pemborosan air, dan menurunkan biaya operasional.

Keaslian/ *state of the art*: Hasil ini mengindikasikan bahwa sistem pengontrolan reservoir berbasis IoT yang diakses melalui *website* dapat menjadi solusi yang efisien dan efektif dalam mengelola sumber daya air di perumahan BCV I.

1. Pendahuluan

Pengelolaan air bersih merupakan isu penting di berbagai perumahan, termasuk di perumahan Bandung City View I (BCV I). Seringkali terjadi pemborosan air akibat kurangnya pengawasan dan pengelolaan yang efisien, terutama pada malam hari saat penggunaan air warga rendah. Hal ini mengakibatkan meluapnya air di reservoir atas yang terus menerima suplai air dari pompa di reservoir bawah. Sistem pengontrolan air yang ada saat ini belum mampu mengatasi masalah ini secara optimal yang berdampak pada peningkatan biaya operasional dan pemborosan sumber daya air.

Dalam beberapa tahun terakhir, teknologi Internet of Things (IoT) telah berkembang pesat dan diterapkan di berbagai bidang, termasuk pengelolaan sumber daya air. IoT memungkinkan integrasi sensor dan perangkat untuk memantau dan mengendalikan sistem secara *real-time* melalui jaringan internet. Berbagai penelitian telah menunjukkan efektivitas IoT dalam meningkatkan efisiensi dan akurasi sistem pengelolaan air [1]. Implementasi sistem berbasis IoT diharapkan dapat menjadi solusi yang tepat untuk mengatasi masalah pemborosan air di BCV I.

Dalam pengembangan sistem ini, teknologi pengembangan web memainkan peran penting. Sistem yang akan dikembangkan terdiri dari dua komponen utama, yaitu *frontend* dan *backend*. *Frontend* menggunakan teknologi seperti HTML, CSS, dan JavaScript untuk membangun antarmuka pengguna yang interaktif dan responsif, sementara *backend* menggunakan *framework* Laravel untuk mengelola logika bisnis, menyimpan, dan mengambil data dari *database*, serta menangani permintaan *server-side* [2]. Laravel menyediakan berbagai fitur seperti *routing*, *middleware*, dan ORM (Eloquent) yang mempermudah pengembangan aplikasi web yang skalabel dan maintainable [3]. Integrasi antara *frontend* dan *backend* dilakukan melalui API untuk pertukaran data secara efisien.

Selain itu, sistem ini akan menggunakan Firebase Realtime Database sebagai solusi penyimpanan *cloud* yang memungkinkan sinkronisasi data secara *real-time* [4]. Firebase akan digunakan untuk menyimpan data kondisi reservoir dan status pompa sehingga pemantauan dan pengontrolan dapat dilakukan secara langsung melalui web dan aplikasi *mobile*.

Keamanan sistem juga menjadi prioritas utama dalam implementasi proyek ini. Laravel Sanctum akan digunakan untuk otentikasi token yang memungkinkan kontrol akses yang aman dan manajemen sesi pengguna [5]. Praktik keamanan terbaik, seperti enkripsi data dan penggunaan HTTPS, akan diterapkan untuk melindungi data pengguna.

Untuk memastikan aplikasi dapat diakses dengan baik di berbagai perangkat, prinsip-prinsip desain responsif dan pengalaman pengguna (UX) akan diterapkan. *Framework* seperti Bootstrap akan digunakan untuk memastikan tampilan yang adaptif, sementara navigasi yang intuitif dan waktu respon yang cepat akan diupayakan untuk meningkatkan pengalaman pengguna.

Pengujian sistem merupakan tahap penting dalam pengembangan melibatkan metode seperti pengujian fungsional untuk memastikan setiap fitur bekerja sesuai spesifikasi, serta pengujian keamanan untuk mengidentifikasi potensi celah keamanan. *Black Box Testing* akan diterapkan untuk memverifikasi fitur tanpa melihat ke dalam kode program [6], [7]. Selain itu, performa sistem akan dimonitor melalui Google Cloud Platform (GCP), menggunakan metrik seperti *CPU utilization* untuk memastikan keandalan sistem saat beroperasi.

Sistem ini akan di-*host* di Google Cloud Platform (GCP), yang menyediakan infrastruktur *cloud* yang aman dan skalabel untuk penyimpanan data, analisis big data, dan *hosting* aplikasi web [8]. Google Compute Engine akan digunakan untuk menjalankan aplikasi dengan konfigurasi yang fleksibel sesuai kebutuhan.

Tujuan dari penelitian ini adalah untuk mengembangkan sistem pengontrolan reservoir air bersih berbasis IoT yang dapat beroperasi secara kontinu selama 24 jam setiap harinya. Sistem ini dirancang untuk memudahkan proses pemantauan dan pengontrolan reservoir air melalui web dan aplikasi mobile. Dengan adanya sistem ini, diharapkan pengelolaan air di BCV I menjadi lebih efisien dan efektif, mengurangi pemborosan air, serta menurunkan biaya operasional. Sistem yang dikembangkan juga diharapkan dapat meningkatkan keamanan dan ketersediaan perangkat keras melalui penyimpanan yang aman di dalam *control room* dan pos satpam. Hasil akhir dari penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam pengelolaan sumber daya air yang lebih baik dan efisien di perumahan BCV I.

2. Metode/Perancangan

2.1. Perancangan UI

Langkah pertama dalam pengembangan website adalah perancangan antarmuka pengguna (UI). Proses ini dimulai dengan membuat *mockup*. *Mockup* dibuat untuk memberikan gambaran yang lebih realistis tentang tampilan akhir dari antarmuka pengguna, termasuk penggunaan warna, tipografi, dan gambar. Alat desain seperti Figma digunakan untuk membuat *mockup* ini.

Desain UI harus responsif, artinya antarmuka harus dapat menyesuaikan diri dengan berbagai ukuran layar dan perangkat, seperti desktop, tablet, dan ponsel. Responsivitas desain dicapai dengan menggunakan framework CSS seperti Bootstrap, yang menyediakan komponen UI siap

pakai dan sistem grid yang fleksibel. Dengan Bootstrap, pengembang dapat dengan cepat membangun antarmuka yang konsisten dan responsif tanpa harus menulis banyak kode CSS dari awal.

2.2. Pengembangan *Frontend*

Setelah desain UI selesai, tahap berikutnya adalah pengembangan *frontend*. *Frontend* adalah bagian dari *website* yang berinteraksi langsung dengan pengguna. Pengembangan *frontend* melibatkan penggunaan HTML, CSS, dan JavaScript.

HTML5 digunakan untuk struktur konten, memastikan bahwa elemen-elemen di halaman web diatur dengan cara yang semantik dan sesuai standar terbaru [9]. HTML5 memungkinkan penggunaan elemen-elemen seperti <header>, <nav>, <section>, dan <footer> yang membuat struktur halaman lebih mudah dibaca dan dikelola.

CSS3 digunakan untuk *styling*, dengan menerapkan desain visual yang konsisten dan menarik. CSS3 memungkinkan penggunaan fitur-fitur canggih seperti animasi, transisi, dan *media queries* untuk membuat desain yang dinamis dan responsif. Preprocessor CSS seperti SASS atau LESS dapat digunakan untuk memudahkan pengelolaan *stylesheet* yang kompleks.

JavaScript digunakan untuk menambahkan interaktivitas ke dalam antarmuka pengguna. Library seperti jQuery dapat digunakan untuk memudahkan manipulasi DOM dan penanganan event [10]. *Framework* JavaScript modern seperti React atau Vue.js juga dapat digunakan untuk membangun antarmuka pengguna yang dinamis dan reaktif.

2.3. Pengembangan *Backend*

Backend adalah bagian dari aplikasi web yang mengelola logika bisnis, penyimpanan data, dan interaksi dengan *server*. Dalam penelitian ini, framework Laravel digunakan untuk mengembangkan *backend*. Laravel dipilih karena menyediakan berbagai fitur yang mempermudah pengembangan aplikasi web yang skalabel dan *maintainable*, seperti *routing* yang memungkinkan pengembang untuk mengarahkan permintaan HTTP ke kontroler yang tepat dengan cara yang mudah dan fleksibel, *middleware* digunakan untuk memproses permintaan sebelum mencapai kontroler, seperti otentikasi, *logging*, dan validasi permintaan, ORM (Eloquent) yang memungkinkan pengembang untuk berinteraksi dengan database menggunakan model dan *query builder* yang intuitif, dan Database MySQL digunakan untuk menyimpan data pengguna dan informasi penting lainnya. Struktur *database* dirancang dengan hati-hati untuk memastikan data dapat diakses dan dikelola dengan efisien. Eloquent ORM memudahkan pengelolaan hubungan antara tabel *database* dan operasi CRUD (*Create, Read, Update, Delete*) pada data.

2.4. Integrasi API

API (*Application Programming Interface*) adalah antarmuka yang memungkinkan komunikasi antara *frontend* dan *backend*. Dalam penelitian ini, API RESTful dikembangkan untuk mempercepat pertukaran data antara klien (*frontend*) dan server (*backend*) [11]. API RESTful digunakan karena kemudahannya dalam mengelola permintaan HTTP dan kemampuan untuk mengembalikan respons dalam format JSON [12]. Langkah-langkah dalam pengembangan API meliputi perancangan *endpoint*, implementasi *endpoint*, keamanan API. Menentukan *endpoint* yang diperlukan untuk mengelola berbagai operasi seperti membaca data, menambahkan data, memperbarui data, dan menghapus data. Mengimplementasikan *endpoint* menggunakan

kontroler di Laravel yang menangani logika bisnis untuk setiap operasi. API dirancang dengan pertimbangan keamanan, termasuk penggunaan token otentikasi melalui Laravel Sanctum. Setiap permintaan ke API divalidasi dan diautentikasi sebelum diproses lebih lanjut, memastikan hanya pengguna yang sah yang dapat mengakses API.

2.5. Integrasi dengan Firebase Realtime Database

Integrasi dengan Firebase Realtime Database adalah langkah penting dalam pengembangan sistem pengontrolan reservoir air bersih berbasis IoT. Firebase Realtime Database dipilih karena kemampuannya untuk menyimpan dan menyinkronkan data secara *real-time* yang sangat penting untuk aplikasi yang memerlukan pembaruan data secara cepat dan konsisten [13].



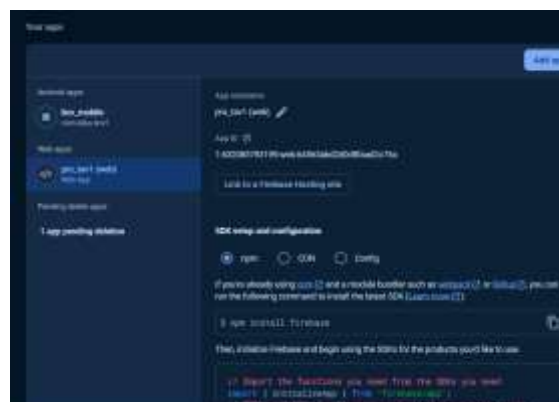
Gambar 1. Halaman Utama Firebase

Langkah pertama dalam integrasi dengan Firebase adalah membuat proyek di Firebase Console. Proses ini melibatkan beberapa langkah utama seperti:



Gambar 2. Membuat Project di Firebase

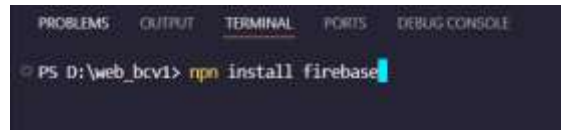
Di Firebase Console, buat proyek baru dengan memberikan nama proyek yang sesuai. Setelah proyek dibuat, Firebase akan memberikan konfigurasi yang diperlukan seperti apiKey, authDomain, databaseURL, projectId, storageBucket, dan messagingSenderId.



Gambar 3. Menambahkan Aplikasi Web ke Firebase

Tambahkan aplikasi web ke proyek Firebase dengan mengikuti langkah-langkah yang disediakan di Firebase Console. Firebase akan memberikan konfigurasi SDK yang diperlukan untuk menghubungkan aplikasi dengan Firebase Realtime Database.

Setelah mendapatkan konfigurasi dari Firebase Console, langkah berikutnya adalah menginisialisasi Firebase di proyek Laravel. Hal ini dilakukan dengan menggunakan Firebase SDK untuk JavaScript dan mengintegrasikannya ke dalam aplikasi Laravel.



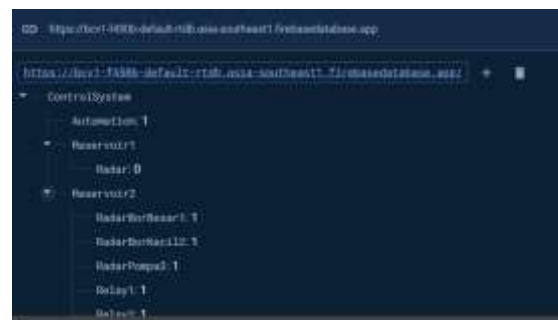
Gambar 4. Command Install Firebase SDK pada Laravel

Firebase SDK diinstalasi dengan menggunakan npm atau yarn. Command untuk menginstal Firebase SDK adalah “npm install firebase” atau “yarn add firebase”.



Gambar 5. Konfigurasi Firebase pada Laravel

Buat file JavaScript di proyek Laravel untuk menginisialisasi Firebase dan masukkan semua konfigurasi yang telah didapat dari Firebase tadi.

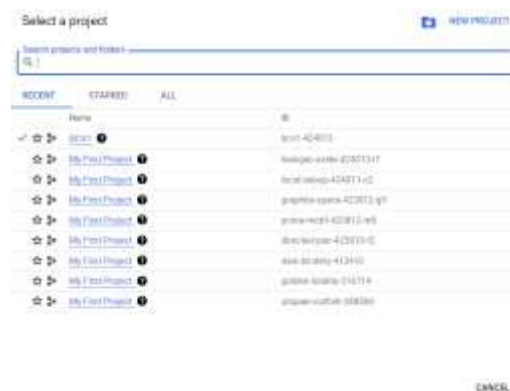


Gambar 6. Struktur Database di Firebase untuk Alat

Setelah Firebase diinisialisasi, langkah berikutnya adalah mengatur Realtime Database di Firebase Console. Di Firebase Console, buat struktur database yang sesuai dengan kebutuhan aplikasi. Struktur ini mencakup node untuk kondisi reservoir, status pompa, dan data pengguna.

2.6. Hosting ke Google Cloud Platform

Setelah pengembangan dan integrasi sistem selesai, langkah berikutnya adalah hosting website ke Google Cloud Platform (GCP). GCP dipilih karena menawarkan skalabilitas, keamanan, dan performa tinggi untuk aplikasi web.



Gambar 7. Tampilan Membuat Project pada GCP

Langkah pertama dalam proses ini adalah membuat proyek baru di Google Cloud Console. Proyek ini akan menjadi wadah untuk semua sumber daya GCP yang diperlukan untuk menjalankan aplikasi Laravel.



Gambar 8. Menyiapkan Virtual Machine

Setelah proyek dibuat, langkah berikutnya adalah menyiapkan instance Virtual Machine (VM) di Compute Engine untuk menjalankan aplikasi Laravel. Proses ini melibatkan membuat instance VM dengan spesifikasi yang sesuai, seperti jenis mesin, jumlah CPU, dan ukuran RAM, serta mengonfigurasi akses SSH untuk mengelola VM dari jarak jauh.



Gambar 9. Memindahkan Kodingan Laravel Memakai SSH

Setelah VM siap, kode aplikasi Laravel diupload ke VM menggunakan SSH dan diinstal semua dependensi menggunakan Composer. Konfigurasi lingkungan dilakukan dengan menyalin file `.env.example` menjadi `.env` dan menyesuaikan pengaturan yang diperlukan, seperti pengaturan database dan kunci aplikasi. Selanjutnya, migrasi database dijalankan untuk membuat tabel yang diperlukan.

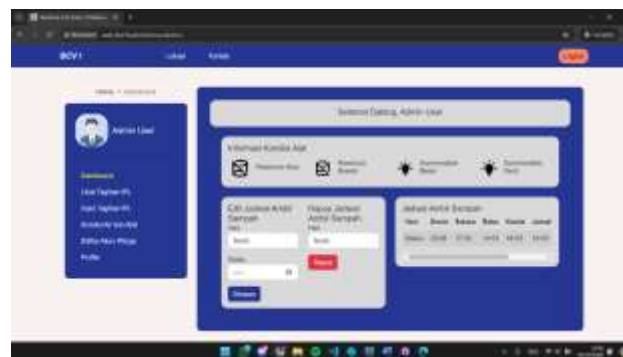
Langkah terakhir adalah mengatur kustom domain dan mengonfigurasi SSL untuk memastikan semua komunikasi terenkripsi. Konfigurasi DNS dilakukan untuk mengarahkan domain custom ke alamat IP VM. Certbot digunakan untuk mendapatkan sertifikat SSL gratis dari Let's Encrypt, dan server web dikonfigurasi untuk menggunakan sertifikat SSL. Sertifikat SSL diperbarui secara otomatis menggunakan cron job untuk memastikan keamanan komunikasi terus terjaga.

Dengan langkah-langkah ini, aplikasi Laravel di-hosting di Google Cloud Platform dengan kustom domain dan komunikasi terenkripsi menggunakan SSL.

3. Hasil dan Pembahasan

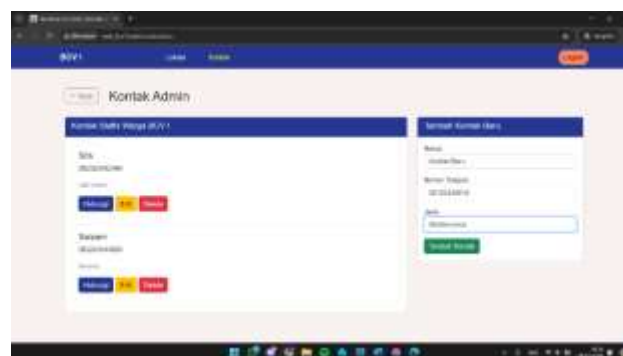
Pada proyek ini, dikembangkan perangkat lunak berbasis web menggunakan framework Laravel, sebuah framework PHP yang memudahkan dalam pembuatan aplikasi web yang efisien dan terstruktur. Website ini dirancang untuk memantau dan mengontrol kondisi reservoir air bersih di Perumahan Bandung City View I secara real-time, dengan integrasi Firebase Realtime Database untuk sinkronisasi data yang cepat dan akurat.

Website ini dilengkapi dengan berbagai fitur, seperti pemantauan kondisi reservoir atas dan bawah, kontrol pompa submersible dan pompa dorong, serta manajemen tagihan IPL warga. Fitur pemantauan memungkinkan pengguna untuk melihat level air dan status pompa secara langsung, sementara fitur kontrol memudahkan pengguna dalam mengoperasikan pompa dari jarak jauh. Selain itu, website ini juga menyediakan halaman khusus bagi admin untuk mengelola tagihan IPL, menambahkan data warga baru, dan melihat informasi kontak staf BCV I. Berikut merupakan beberapa tampilan dari fitur-fitur utama di website khusus admin:



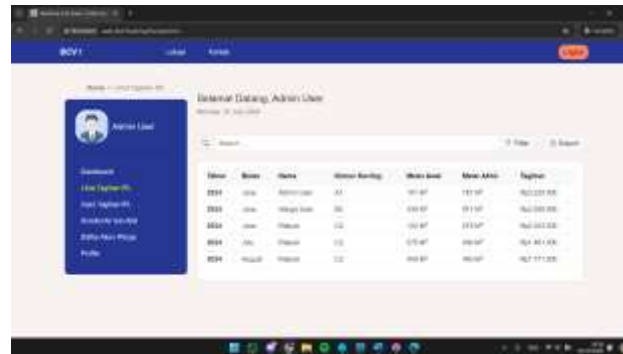
Gambar 10. Halaman Dashboard Admin

Pengguna yang memiliki role admin dapat melihat informasi beberapa kondisi alat yang terdiri dari Reservoir Atas, Reservoir Bawah, Submersible Besar, dan Submersible Kecil. Admin juga dapat mengelola jadwal pengambilan sampah, dengan kemampuan untuk menambahkan atau menghapus jadwal pada hari tertentu. Selain itu, ada daftar jadwal pengambilan sampah yang sudah ditetapkan.



Gambar 11. Halaman Kontak Admin

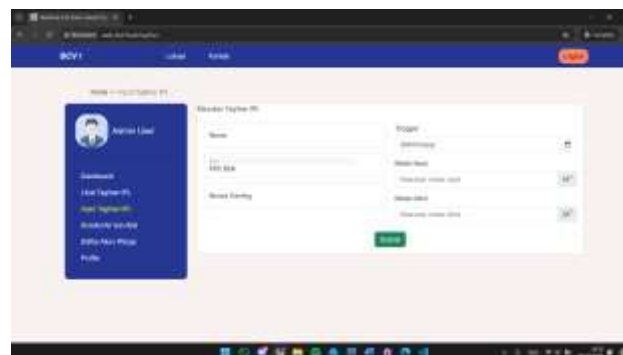
Pengguna yang terautentikasi sebagai admin dapat melihat daftar kontak staf yang bekerja di BCV I pada halaman kontak admin ini. Admin dapat mengedit atau menghapus setiap kontak melalui tombol yang tersedia. Selain itu, admin dapat menambahkan kontak baru dengan mengisi formulir di sebelah kanan halaman yang mencakup nama, nomor telepon, dan jenis tugas.



Tahun	Bulan	Nama	Nomor Kavling	Meter Awal	Meter Akhir	Tagihan
2024	Jan	Adhitya Pratomo	01	101.87	101.90	162.000.000
2024	Jan	Adhitya Pratomo	02	100.87	101.90	162.000.000
2024	Jan	Adhitya Pratomo	03	100.87	101.90	162.000.000
2024	Jan	Adhitya Pratomo	04	100.87	101.90	162.000.000
2024	Jan	Adhitya Pratomo	05	100.87	101.90	162.000.000

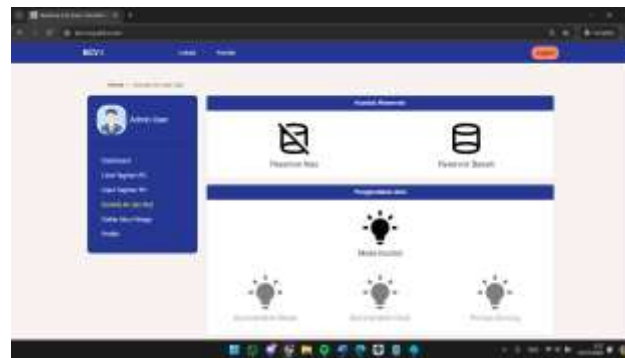
Gambar 12. Halaman Lihat Tagihan IPL

Admin dapat melihat daftar lengkap semua tagihan IPL yang dimiliki warga BCV I pada halaman "Lihat Tagihan IPL". Daftar ini mencakup informasi seperti tahun, bulan, nama, nomor kavling, meter awal, meter akhir, dan jumlah tagihan. Mereka juga dapat menggunakan fitur pencarian untuk mencari tagihan tertentu, dan mereka juga dapat menggunakan fitur filter dan ekspor untuk memfilter dan mengunduh data tagihan sesuai kebutuhan.



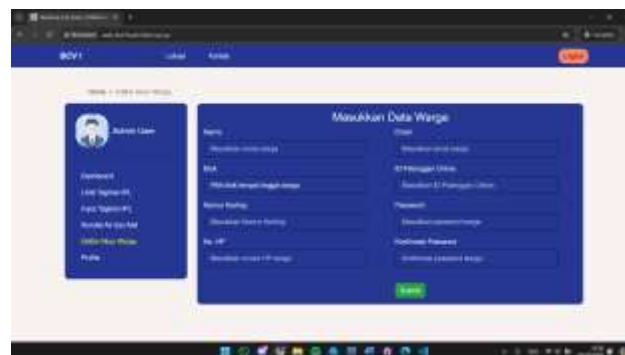
Gambar 13. Halaman Input Tagihan IPL

Pada halaman "Input Tagihan IPL", manajer dapat memasukkan tagihan IPL untuk setiap warga BCV I. Manajer hanya perlu memilih blok dan nomor kavling yang benar, maka nama warga yang tinggal di rumah tersebut akan muncul secara otomatis jika data tersebut valid. Selain itu, meter awal juga akan muncul secara otomatis berdasarkan data dari meteran bulan lalu. Setelah memasukkan tanggal dan meter akhir, manajer kemudian dapat mengklik tombol "Submit" untuk menyimpan tagihan tersebut.



Gambar 14. Halaman Kondisi Air dan Alat

Pada halaman "Kondisi Air dan Alat", manajer dapat melihat dan mengontrol kondisi peralatan yang terhubung dengan sistem pengelolaan air di perumahan BCV I. Manajer dapat melihat status reservoir atas dan bawah serta mengontrol mode kontrol (mengubah alat menjadi beroperasi secara manual atau otomatis), submersible besar, submersible kecil, dan pompa dorong dengan mengklik UI yang tersedia.



Gambar 15. Halaman Pendaftaran Akun Warga

Untuk memasukkan data warga baru yang tinggal di perumahan BCV I, administrator harus mengisi formulir berikut: nama warga, blok tempat tinggal, nomor kavling, nomor HP, email, ID pelanggan online, password, dan konfirmasi password. Setelah semua data dimasukkan, administrator dapat menekan tombol "Submit" untuk menyimpan data warga baru ke dalam sistem.

Selanjutnya, pengujian dilakukan dengan cara pengujian UI menggunakan metode blackbox. Metode ini digunakan untuk memastikan bahwa setiap fitur antarmuka pengguna berfungsi sesuai dengan spesifikasi tanpa melihat ke dalam kode internal. Pengujian dilakukan dengan memberikan berbagai input dan mengamati output yang dihasilkan untuk memverifikasi bahwa sistem beroperasi sebagaimana mestinya. Berikut adalah hasil dari pengujian *blackbox*:

Tabel 1. Pengujian *Blackbox Role Admin*

Judul Tes	Skenario Tes	Kondisi Awal	Endpoint	Langkah	Output yang diharapkan	Output yang keluar	Status
<i>Login Berhasil</i>	<i>User login dengan kredensial yang benar</i>	<i>User sudah terdaftar dan memiliki kredensial yang valid</i>	<i>https://bcv1.my.id/login</i>	1.Pergi ke halaman login 2.Masukkan <i>username</i> dan <i>password</i> yang benar 3.Klik tombol “ <i>Sign In</i> ”	<i>User</i> diarahkan ke <i>dashboard</i> dari masing-masing <i>role</i> akun	<i>User</i> diarahkan ke <i>dashboard</i> dari masing-masing <i>role</i> akun	<i>PASSED</i>
<i>Login Gagal</i>	<i>User login dengan kredensial yang salah</i>	<i>User sudah terdaftar namun menggunakan kredensial yang salah</i>	<i>https://bcv1.my.id/login</i>	1.Pergi ke halaman <i>login</i> 2.Masukkan <i>username</i> dan <i>password</i> yang salah 3.Klik tombol “ <i>Sign In</i> ”	<i>User</i> gagal <i>login</i> dan muncul pesan error “ <i>Invalid username or password</i> ”	<i>User</i> gagal <i>login</i> dan muncul pesan error “ <i>Invalid credentials. Please Try Again.</i> ”	<i>PASSED</i>
<i>Load Dashboard Admin</i>	<i>Dashboard admin dimuat dengan benar</i>	<i>User login sebagai admin</i>	<i>https://bcv1.my.id/dashboardadmin</i>	1.Buka halaman	Halaman <i>dashboard</i> admin	Halaman <i>dashboard</i> admin	<i>PASSED</i>

Judul Tes	Skenario Tes	Kondisi Awal	Endpoint	Langkah	Output yang diharapkan	Output yang keluar	Status
				<i>Dashb oard</i>	dimuat dengan informasi kondisi alat, jadwal ambil sampah, dan pilihan edit/hapus jadwal	dimuat dengan informasi kondisi alat, jadwal ambil sampah, dan pilihan edit/hapus jadwal	
<i>Load</i> Jadwal Ambil Sampah	Jadwal ambil sampah dimuat dengan benar	<i>User login</i> sebagai admin	https://bcv1.my.id/dashboardadmin	1. Buka halaman <i>Dashb oard</i>	Jadwal ambil sampah ditampilkan dengan benar di tabel	Jadwal sampah yang tertampil ialah jadwal sampah yang sudah diperbarui oleh admin	<i>PASS ED</i>
Edit Jadwal Ambil Sampah	Admin mengedit jadwal ambil sampah	<i>User login</i> sebagai admin	https://bcv1.my.id/dashboardadmin	1. Masukkan hari dan waktu yang valid pada bagian "Edit Jadwal Ambil	Jadwal ambil sampah diperbarui dan tampil di tabel	Tampilan <i>dashbo ard</i> dengan jadwal ambil sampah yang sudah diperbarui	<i>PASS ED</i>

Judul Tes	Skenario Tes	Kondisi Awal	Endpoint	Langkah	Output yang diharapkan	Output yang keluar	Status
				Sampah"			
				2.Klik tombol "Simpan"			
Hapus Jadwal Ambil Sampah	Admin menghapus jadwal ambil sampah	User login sebagai admin	https://bcv1.my.id/dashboardadmin	1.Masukkan hari yang ingin dihapus pada bagian "Hapus Jadwal Ambil Sampah"	Jadwal ambil sampah dihapus dari tabel berdasar hari pilihan	Pada tabel jadwal ambil sampah, ada jam yang terhapus pada pilihan hari sebelumnya	PASSED
Load Informasi Kondisi Alat	Informasi kondisi alat dimuat dengan benar	User login sebagai admin	https://bcv1.my.id/dashboardadmin	1.Buka halaman Dashboard	Informasi kondisi alat (Reservoir Atas/Bawah, Submersible Besar/Kecil, Pompa)	Informasi kondisi alat (Reservoir Atas/Bawah, Submersible Besar/Kecil, Pompa)	PASSED

Judul Tes	Skenario Tes	Kondisi Awal	Endpoint	Langkah	Output yang diharapkan	Output yang keluar	Status
					dengan status terkini	dengan status terkini	
<i>View Bill Details</i>	<i>User</i> menavigasi ke halaman Lihat Tagihan IPL	<i>User</i> sudah <i>login</i>	https://bcv1.my.id/t agihanipladmin	1. <i>Login</i> sebagai Admin 2. Klik menu "Lihat Tagihan IPL"	Halaman menampilkan daftar tagihan IPL dengan benar	Halaman menampilkan daftar tagihan IPL dengan benar	<i>PASSED</i>
<i>Search Bill</i>	<i>User</i> menggunakan fitur pencarian untuk menemukan tagihan tertentu	<i>User</i> berada di halaman tagihan	https://bcv1.my.id/t agihanipladmin	1. <i>Login</i> sebagai Admin 2. Klik menu "Lihat Tagihan IPL" 3. Masukkan kata kunci ke dalam kotak pencarian dan tekan enter	Hasil pencarian menampilkan pilkan tagihan yang sesuai dengan kata kunci	Hasil pencarian menampilkan pilkan tagihan yang sesuai dengan kata kunci	<i>PASSED</i>
<i>Filter Bill</i>	<i>User</i> menggunakan fitur filter	<i>User</i> berada di halaman tagihan	https://bcv1.my.id/t agihanipladmin	1. <i>Login</i> sebagai Admin 2. Klik menu	Daftar tagihan ditampilkan sesuai dengan	Tidak menampilkan data apapun	<i>FAILED</i>

Judul Tes	Skenario Tes	Kondisi Awal	Endpoint	Langkah	Output yang diharapkan	Output yang keluar	Status
				"Lihat Tagihan IPL" 3. Klik tombol "Filter" dan pilih kriteria filter	kriteria filter yang dipilih		
<i>Export Bill Data</i>	<i>User</i> mengeksport data tagihan ke <i>file</i>	<i>User</i> berada di halaman tagihan	https://bcv1.my.id/tagihanipladmin	1. <i>Login</i> sebagai Admin 2. Klik menu "Lihat Tagihan IPL" 3. Klik tombol "Export"	Data tagihan diekspor ke file dengan format yang benar	Data tagihan berhasil diekspor dengan format .xlsx	<i>PASSED</i>
<i>Fetch Nama dan Meter Awal</i>	Nama warga dan meter awal otomatis terisi berdasarkan blok dan nomor kaveling	<i>User login</i> sebagai admin	https://bcv1.my.id/tagihan	1. Pilih blok yang valid pada <i>form</i> blok 2. Isi nomor kaveling yang valid 3. Nama warga	Nama warga dan meter awal otomatis terisi sesuai data yang ada di <i>database</i>	Nama warga dan meter awal otomatis terisi sesuai data yang ada di <i>database</i>	<i>PASSED</i>

Judul Tes	Skenario Tes	Kondisi Awal	Endpoint	Langkah	Output yang diharapkan	Output yang keluar	Status
				dan meter awal otomatis terisi			
Load Input Tagihan IPL	Halaman input tagihan IPL dimuat dengan benar	User login sebagai admin	https://bcv1.my.id/tagihan	1. Buka halaman input tagihan IPL	Halaman input tagihan IPL dimuat dengan form input lengkap	Muncul banyak form untuk mengisi kepengantagihan IPL warga	PASSED
Isi Data Tanggal	Admin mengisi tanggal tagihan IPL	User login sebagai admin	https://bcv1.my.id/tagihan	1. Pilih tanggal pada input tanggal	Tanggal terisi dengan format yang benar	Muncul kalender dan user bisa memilih tanggal yang diinginkan	PASSED
Isi Meter Akhir	Admin mengisi meter akhir dengan data valid	User login sebagai admin	https://bcv1.my.id/tagihan	1. Isi meter akhir pada input text	Bila admin mengisi meter akhir yang kurang dari meter awal, maka	Admin berhasil mendaftarkan tagihan IPL jika mengisi meter akhir	PASSED

Judul Tes	Skenario Tes	Kondisi Awal	Endpoint	Langkah	Output yang diharapkan	Output yang keluar	Status
					pendaftaran akan gagal, apabila admin mengisi meter akhir yang lebih dari meter awal, maka pendaftaran tagihan akan berhasil	yang lebih dari meter awal	
<i>Submit Form</i>	Admin men-submit <i>form</i> dengan data yang valid	<i>User login</i> sebagai admin	https://bcv1.my.id/tagihan	1.Isi semua data pada form dengan data yang valid 2.Klik tombol " <i>Submit</i> "	Tagihan IPL berhasil ditambah dan muncul pesan sukses	Tagihan IPL berhasil ditambah dan muncul pesan "Pendaftaran Tagihan Berhasil"	<i>PASSED</i>
Validasi Data	Admin men-submit <i>form</i> dengan data	<i>User login</i>	https://bcv1.my.id/tagihan	1.Biarkan beberapa	Muncul pesan error	Muncul pesan error	<i>PASSED</i>

Judul Tes	Skenario Tes	Kondisi Awal	Endpoint	Langkah	Output yang diharapkan	Output yang keluar	Status
Kosong	yang kosong atau tidak lengkap	sebagai admin		pa atau semua input form kosong 2.Klik tombol "Submit"	validasi data yang kosong	validasi data yang kosong seperti "Please Fill Out This Form"	
Validasi Nomor Kavling	Admin mengisi nomor kavling yang tidak valid	User login sebagai admin	https://bcv1.my.id/t agihan	1. Isi nomor kavling dengan data tidak valid (misal nomor kavling yang tidak terdaftar) 2. Klik tombol "Submit"	Muncul pesan error validasi nomor kavling	Tidak muncul nama warga dan tidak muncul meteran akhirnya	PASS ED
View Reservoir Status	User menavigasi ke halaman Kondisi Air dan Alat	User sudah login	https://bcv1.my.id/ kondisi	1. Login sebagai Admin 2. Klik menu "Kondisi Air dan Alat"	Halaman menampilkan status reservoir atas dan bawah dengan benar	Halaman menampilkan status reservoir atas dan bawah dengan benar	PASS ED

Judul Tes	Skenario Tes	Kondisi Awal	Endpoint	Langkah	Output yang diharapkan	Output yang keluar	Status
<i>Control Mode</i>	<i>User</i> menavigasi ke halaman Kondisi Air dan Alat dan melihat mode kontrol	<i>User</i> sudah <i>login</i>	https://bcv1.my.id/ kondisi	1. <i>Login</i> sebagai Admin 2. Klik menu "Kondisi Air dan Alat"	Mode kontrol ditampilkan dengan benar	Mode kontrol ditampilkan dengan benar	<i>PASSED</i>
<i>Control Submersible Besar</i>	<i>User</i> mencoba mengaktifkan /deaktivasi Submersible Besar	<i>User</i> berada di halaman kondisi	https://bcv1.my.id/ kondisi	1. <i>Login</i> sebagai Admin 2. Klik menu "Kondisi Air dan Alat" 3. Klik tombol "Submersible Besar"	Status Submersible Besar berubah sesuai dengan tindakan yang dilakukan	Status Submersible Besar berubah sesuai dengan tindakan yang dilakukan	<i>PASSED</i>
<i>Control Submersible Kecil</i>	<i>User</i> mencoba mengaktifkan /deaktivasi Submersible Kecil	<i>User</i> berada di halaman kondisi	https://bcv1.my.id/ kondisi	1. <i>Login</i> sebagai Admin 2. Klik menu "Kondisi Air dan Alat" 3. Klik tombol "Subm	Status Submersible Kecil berubah sesuai dengan tindakan yang dilakukan	Status Submersible Kecil berubah sesuai dengan tindakan yang dilakukan	<i>PASSED</i>

Judul Tes	Skenario Tes	Kondisi Awal	Endpoint	Langkah	Output yang diharapkan	Output yang keluar	Status
				ersible Kecil"			
Control Pompa Dorong	User mencoba mengaktifkan /deaktivasi Pompa Dorong	User berada di halaman kondisi	https://bcv1.my.id/ kondisi	1. Login sebagai Admin 2. Klik menu "Kondisi Air dan Alat" 3. Klik tombol "Pompa Dorong"	Status Pompa Dorong berubah sesuai dengan tindakan yang dilakukan	Status Pompa Dorong berubah sesuai dengan tindakan yang dilakukan	PASSED
Load Halaman Daftar Warga	Halaman daftar warga dimuat dengan benar	User login sebagai admin	https://bcv1.my.id/ daftarwarga	1. Buka halaman daftar warga	Halaman daftar warga dimuat dengan form input lengkap	Halaman daftar warga dimuat dengan form untuk mengisi data warga	PASSED
Isi Data Nama	Admin mengisi nama warga yang valid	User login sebagai admin	https://bcv1.my.id/ daftarwarga	1. Isi nama warga pada input text	Nama warga terisi dengan benar	Nama warga terisi dengan benar	PASSED
Isi Data Blok	Admin memilih blok yang valid	User login	https://bcv1.my.id/ daftarwarga	1. Pilih blok pada	Blok terisi	Blok terpilih	PASSED

Judul Tes	Skenario Tes	Kondisi Awal	Endpoint	Langkah	Output yang diharapkan	Output yang keluar	Status
	dari dropdown	sebagai admin		dropdo wn	dengan benar	dengan benar	
Isi Data Nomor Kavling	Admin mengisi nomor kavling yang valid	User login sebagai admin	https://bcv1.my.id/daftarwarga	1. Isi nomor kavling pada form kavling	Nomor kavling terisi dengan benar	Nomor kavling terisi dengan benar	PASS ED
Isi Data Email	Admin mengisi email yang valid	User login sebagai admin	https://bcv1.my.id/daftarwarga	1. Isi email warga pada form email	Email terisi dengan benar	Email terisi dengan benar	PASS ED
Isi Data ID Pelanggan	Admin mengisi ID pelanggan yang valid	User login sebagai admin	https://bcv1.my.id/daftarwarga	1. Isi ID pelanggan online pada input text	ID pelanggan terisi dengan benar	ID pelanggan terisi dengan benar	PASS ED
Isi Data No HP	Admin mengisi nomor HP yang valid	User login sebagai admin	https://bcv1.my.id/daftarwarga	1. Isi nomor HP warga pada input text	Nomor HP terisi dengan benar	Nomor HP terisi dengan benar	PASS ED
Isi Data Password	Admin mengisi password yang valid dan konfirmasi password yang sama	User login sebagai admin	https://bcv1.my.id/daftarwarga	1. Isi password pada form password 2. Isi konfirmasi	Password dan konfirmasi password terisi dengan benar	Password dan konfirmasi password terisi dengan benar	PASS ED

Judul Tes	Skenario Tes	Kondisi Awal	Endpoint	Langkah	Output yang diharapkan	Output yang keluar	Status
				password pada form konfirmasi password			
Submit Form	Admin men-submit form dengan data yang valid	User login sebagai admin	https://bcv1.my.id/daftarwarga	1. Isi semua data pada form 2. Klik tombol "Submit"	Data warga berhasil ditambah dan muncul pesan sukses	Data warga berhasil ditambah dan muncul pesan "Warga Berhasil Terdaftar"	PASSED
Validasi Data Kosong	Admin men-submit form dengan data yang kosong atau tidak lengkap	User login sebagai admin	https://bcv1.my.id/daftarwarga	1. Biarkan beberapa atau semua input form kosong 2. Klik tombol "Submit"	Muncul pesan error validasi data yang kosong	Muncul pesan error validasi data yang kosong dan meminta kita untuk mengisinya	PASSED
View Profile	User menavigasi ke halaman Profile	User sudah login	https://bcv1.my.id/profileadmin	1. Login sebagai Admin	Halaman menampilkan informasi	Halaman menampilkan informasi	PASSED

Judul Tes	Skenario Tes	Kondisi Awal	Endpoint	Langkah	Output yang diharapkan	Output yang keluar	Status
				2. Klik menu "Profil"	asi profil dengan benar	asi profil dengan benar	

Tabel 2. Pengujian *Blackbox Role* Warga

Judul Tes	Skenario Tes	Kondisi Awal	Endpoint	Langkah	Output yang diharapkan	Output yang keluar	Status
<i>View Dashboard</i> Warga	<i>User</i> menavigasi ke halaman Dashboard Warga	<i>User</i> sudah login	https://bcv1.my.id/dashboard	1. <i>Login</i> sebagai Warga 2. Klik menu "Dashboard"	Halaman menampilkan informasi dashboard berupa nama warga, jadwal ambil sampah, dan IPL bulan ini dengan benar	Halaman menampilkan informasi dashboard berupa nama warga, jadwal ambil sampah, dan IPL bulan ini dengan benar	PASSED
<i>Navigate to Payment</i>	<i>User</i> menavigasi ke halaman cara bayar	<i>User</i> berada di halaman dashboard	https://bcv1.my.id/dashboard	1. <i>Login</i> sebagai Warga 2. Klik menu "Dashboard" 3. Klik tombol	Halaman cara bayar ditampilkan dengan benar	Halaman cara bayar ditampilkan dengan benar	PASSED

Judul Tes	Skenario Tes	Kondisi Awal	Endpoint	Langkah	Output yang diharapkan	Output yang keluar	Status
				"Cara Bayar"			
Load Halaman Detail Tagihan IPL	Halaman detail tagihan IPL dimuat dengan benar	User login sebagai warga	https://bcv1.my.id/detailtagihan	1. Buka halaman detail tagihan IPL	Halaman detail tagihan IPL dimuat dengan tabel tagihan IPL yang berisi data yang relevan untuk warga tersebut	Muncul tabel berisi data tagihan yang dimiliki oleh warga	PASSED
Filter Data Tagihan	Filter data tagihan IPL berdasarkan kriteria tertentu (tahun, bulan, meter awal, meter akhir)	User login sebagai warga	https://bcv1.my.id/detailtagihan	1. Buka halaman detail tagihan IPL 2. Gunakan fitur filter untuk mencari data tagihan berdasarkan kriteria tertentu	Data tagihan IPL difilter dan hanya menampilkan data yang sesuai dengan kriteria yang dipilih	Tidak menampilkan data apapun	FAILED
Export Data	Export data tagihan	User login sebagai	https://bcv1.my.id/detailtagihan	1. Buka halaman detail	Data tagihan IPL	Terdownload file	PASSED

Judul Tes	Skenario Tes	Kondisi Awal	Endpoint	Langkah	Output yang diharapkan	Output yang keluar	Status
Tagihan	IPL ke file excel	ai warga		tagihan IPL 2. Klik tombol "Export"	diekspor ke file excel	excel yang berisi tabel dari data tagihan IPL	
Tampilkan Cara Bayar	Menampilkan cara bayar tagihan IPL	User login sebagai ai warga	https://bcv1.my.id/detailtagihan	1. Buka halaman detail tagihan IPL 2. Klik tombol "Cara Bayar"	Muncul <i>pop-up</i> yang menampilkan cara bayar tagihan IPL	Halaman cara bayar ditampilkan dengan benar	PASSED
Load Halaman Profil Warga	Halaman profil warga dimuat dengan benar	User login sebagai warga	https://bcv1.my.id/profilewarga	1. Buka halaman profil warga	Halaman profil warga dimuat dengan data pribadi warga (nama, nomor kavling, nomor HP)	Halaman profil warga dimuat dengan data pribadi warga (nama, nomor kavling, nomor HP)	PASSED
Edit Profil Warga	Warga dapat mengubah data profilnya	User login sebagai warga	https://bcv1.my.id/profilewarga	1. Buka halaman profil warga 2. Klik tombol "Edit Profile" 3. Ubah	Data profil warga diperbarui dan ditampilkan kembali di halaman	Data profil warga diperbarui dan ditampilkan kembali di halaman	PASSED

Judul Tes	Skenario Tes	Kondisi Awal	Endpoint	Langkah	Output yang diharapkan	Output yang keluar	Status
				data profil 4.Simpan perubahan	profil dengan informasi yang telah diubah	profil dengan informasi yang telah diubah	

Untuk *role* admin, pengujian menunjukkan bahwa halaman *dashboard* berhasil menampilkan informasi dengan tepat, dan semua input data (nomor HP, *password*, nama, blok, nomor kavling, email, dan ID pelanggan) divalidasi dan disimpan dengan benar. Formulir berhasil di-submit dengan data valid, dan sistem memberikan pesan sukses, sementara form dengan data kosong memberikan pesan error yang sesuai. Kontrol perangkat (Submersible Besar, Submersible Kecil, dan Pompa Dorong) berfungsi dengan baik, menunjukkan bahwa integrasi kontrol perangkat dilakukan dengan benar.

Pengujian dengan metode blackbox menunjukkan bahwa hampir semua fitur berfungsi dengan baik. Namun, terdapat satu fitur yang gagal, yaitu fitur filter, yang tidak berfungsi sesuai dengan harapan.

Pengujian performa sistem dilakukan menggunakan Postman untuk mengukur respon API *login* dan Google Cloud Platform (GCP) untuk memonitor penggunaan CPU. Pengujian dilakukan dengan tiga skenario beban berbeda, yaitu 20, 55, dan 80 *virtual users* untuk mengevaluasi bagaimana sistem menangani permintaan yang meningkat. Setiap pengujian menggunakan Postman untuk mengirim permintaan secara simultan dari *virtual users*, sementara GCP digunakan untuk memantau dan mencatat penggunaan CPU selama pengujian.



Gambar 16. Skenario Pengujian 20 *Virtual Users* dalam Waktu 5 Menit

Selama pengujian beban menggunakan 20 *virtual users* di Postman, grafik menunjukkan peningkatan yang signifikan dalam penggunaan CPU pada *instance* VM di Google Cloud

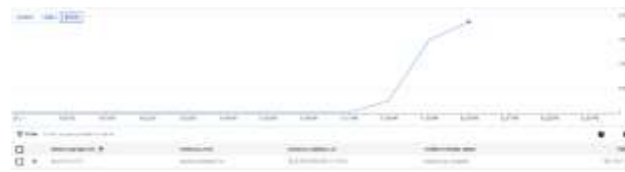
Platform (GCP). Pada awal pengujian, penggunaan CPU mulai meningkat tajam sekitar pukul 11:11 AM, mencapai puncak sekitar pukul 11:12 AM. Hal ini menunjukkan bahwa beban pada VM meningkat secara signifikan akibat banyaknya permintaan yang datang secara bersamaan. Setelah mencapai puncak, penggunaan CPU mulai menurun dan stabil pada kisaran tertentu, meskipun tetap tinggi, hal ini mengindikasikan bahwa VM terus bekerja di bawah beban berat. Fluktuasi dalam grafik menunjukkan beberapa penurunan penggunaan CPU yang signifikan pada pukul 11:19 AM dan 11:23 AM, yang mungkin disebabkan oleh berkurangnya jumlah permintaan atau penyelesaian batch permintaan tertentu oleh VM. Secara keseluruhan, hasil pengujian ini menunjukkan bahwa VM dapat menangani beban kerja yang berat dari 20 *virtual users*, namun tetap beroperasi pada kapasitas yang tinggi yang mengindikasikan perlunya optimisasi lebih lanjut untuk menjaga performa yang konsisten dan efisien.



Gambar 17. Skenario Pengujian 55 *Virtual Users* dalam Waktu 5 Menit

Pada pengujian dengan 55 *virtual users* menggunakan Postman, data menunjukkan total 1,268 permintaan yang dikirimkan dengan rata-rata 4.09 permintaan per detik dan waktu respons rata-rata mencapai 11,559 milidetik. Meskipun tidak ada *error rate* yang tercatat, waktu respons yang tinggi menunjukkan bahwa beban tambahan dari 55 *virtual users* menyebabkan penurunan kinerja yang signifikan. Grafik *monitoring CPU utilization* di Google Cloud Platform (GCP) menunjukkan bahwa penggunaan CPU meningkat secara drastis selama pengujian. Pada pukul 11:31 AM, penggunaan CPU mulai naik dan mencapai puncak pada pukul 11:33 AM. Setelah mencapai puncaknya, penggunaan CPU menunjukkan fluktuasi tetapi tetap tinggi, menunjukkan bahwa VM berada di bawah tekanan beban yang berat selama pengujian. Hasil pengujian ini menunjukkan bahwa sistem mengalami peningkatan waktu respons yang signifikan, mengindikasikan bahwa *server* menghadapi kesulitan dalam menangani jumlah permintaan yang lebih tinggi yang menyebabkan penurunan kinerja.





Gambar 18. Skenario Pengujian 80 *Virtual Users* dalam Waktu 5 Menit

Pada pengujian dengan 80 *virtual users* menggunakan Postman, grafik menunjukkan lonjakan yang signifikan dalam waktu respons rata-rata dan stabilitas permintaan per detik. Data menunjukkan total 1,253 permintaan yang dikirimkan dengan rata-rata 4.05 permintaan per detik dan waktu respons rata-rata mencapai 17,147 milidetik. Meskipun masih tidak ada *error rate* yang tercatat, waktu respons yang tinggi menunjukkan bahwa beban tambahan dari 80 *virtual users* menyebabkan penurunan kinerja yang sangat signifikan. Grafik *monitoring CPU utilization* di Google Cloud Platform (GCP) menunjukkan bahwa penggunaan CPU melonjak tajam selama pengujian. Pada pukul 11:57 AM, penggunaan CPU mulai naik dan mencapai puncak pada pukul 12:00 PM, melebihi 187% penggunaan. Setelah mencapai puncaknya, penggunaan CPU menunjukkan sedikit penurunan tetapi tetap tinggi, menunjukkan bahwa VM berada di bawah tekanan beban yang sangat berat selama pengujian. Hasil pengujian ini menunjukkan bahwa sistem mengalami penurunan kinerja yang drastis saat menghadapi jumlah permintaan yang sangat tinggi yang menandakan bahwa *server* mencapai batas kemampuannya dalam menangani permintaan tambahan dari 80 *virtual users*.

4. Kesimpulan dan Saran

Penelitian ini berhasil mengembangkan sistem pengontrolan reservoir air bersih berbasis IoT untuk Perumahan Bandung City View I yang dapat diakses melalui *website* menggunakan *framework* Laravel dan Firebase Realtime Database untuk sinkronisasi data *real-time*. Pengujian fungsional menunjukkan bahwa sebagian besar fitur berfungsi sesuai spesifikasi, termasuk kontrol perangkat dan pemantauan kondisi reservoir, dengan pengecualian fitur filter yang belum berfungsi optimal. Sistem ini terbukti efektif dalam mengurangi pemborosan air dan menurunkan biaya operasional yang memberikan solusi yang efisien dan dapat diandalkan untuk pengelolaan sumber daya air di lingkungan perumahan. Hasil pengujian dengan 20, 55, dan 80 *virtual users* menunjukkan peningkatan signifikan dalam waktu respons dan penggunaan CPU seiring bertambahnya beban, meskipun tidak ada *error rate* yang tercatat. Hal ini menunjukkan bahwa sistem memerlukan optimisasi lebih lanjut untuk memastikan kinerja yang konsisten dan efisien. Disarankan untuk mengoptimalkan kode *backend* dan *database query*, mengaktifkan *auto scaling* di GCP untuk penyesuaian kapasitas otomatis [14], menggunakan *load balancer* untuk mendistribusikan permintaan [15], serta melakukan *monitoring* berkelanjutan untuk mengidentifikasi *bottleneck* dan meningkatkan efisiensi sistem.

Daftar Pustaka

- [1] P. F. Laravel *et al.*, “Pemanfaatan Framework Laravel dan Framework Bootstrap pada Pembangunan Aplikasi Penejualan Hijab Berbasis Web,” *Jurnal Media Infotama*, vol. 18, no. 1, p. 2022.

- [2] A. Ratino, R. Astri, and P. Anggraini, “Implementasi Framework Laravel Dalam Pengembangan Aplikasi E-Commerce Untuk Toko Jago Software,” 2023.
- [3] A. M. Yudha and A. B. Cahyono, “Pengembangan Back End Menggunakan Laravel Lumen (Studi Kasus: Teknologi.id Event).” [Online]. Available: <https://www.exabytes.co.id/blog/pengertian-api>
- [4] E. A. W. Sanad, “Pemanfaatan Realtime Database di Platform Firebase Pada Aplikasi E-Tourism Kabupaten Nabire,” *Jurnal Penelitian Enjiniring*, vol. 22, no. 1, pp. 20–26, May 2019, doi: 10.25042/jpe.052018.04.
- [5] D. Eko Nugroho, J. Christian Chandra, U. Budiyanto, and T. Fatimah, “Implementasi Web Service dengan Metode RESTFUL pada Sistem Penjualan Barang di PT. WELLCOM GROUP,” *SENAFTI*, vol. 2, no. 1, pp. 534–543, 2023.
- [6] M. S. Mustaqbal, R. F. Firdaus, and H. Rahmadi, “Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus : APlikasi Prediksi Kelulusan SNMPTN),” 2015.
- [7] A. D. Frayudha, I. R. Pande, and M. B. Juwita, “Implementation of Black Box Testing with the Application of Equivalence Partitioning Techniques in the M-Magazine Android Application at Semen Gresik High School,” *Elinvo (Electronics, Informatics, and Vocational Education)*, vol. 9, no. 1, pp. 134–143, Jun. 2024, doi: 10.21831/elinvo.v9i1.70382.
- [8] N. Ramsari and A. Ginanjar, “Implementasi Infrastruktur Server Berbasis Cloud Computing Untuk Web Service Berbasis Teknologi Google Cloud Platform,” in *SENANTIK 2021*, 2022, pp. 169–182. doi: 10.28989/senatik.v7i1.472.
- [9] H. Sama and E. Hartanto, “Studi Deskriptif Evolusi Website Dari Html1 Sampai Html5 Dan Pengaruhnya Terhadap Perancangan Dan Pengembangan Website,” in *Conference on Management, Business, Innovation, Education, and Social Science*, 2021, pp. 589–596.
- [10] M. Sulehu and Markani, “Integrasi Pemrosesan Bahasa Alami dalam Bot Telegram untuk Pengajaran Mata Kuliah Website Lebih Interaktif,” *Jurnal Minfo Polgan*, vol. 13, no. 1, pp. 759–771, 2024.
- [11] M. K. Naufal, F. Affrianto, and A. B. Cahyono, “Implementasi REST API Untuk Fitur Rencana Strategis Program Pada SIMPEDA.”
- [12] M. F. Ramadhan and Z. Zukhri, “Muhammad Fachri Ramadhan Pengembangan REST API Sistem Pengembangan REST API Sistem UIIAdmisi dengan Menggunakan Pendekatan Domain Driven Design.”
- [13] R. A. Setyawan, “Penerapan Firebase Realtime Database Pada Aplikasi Catatan Harian Diabetes Melitus,” *Bisnis dan Manajemen*, vol. 22, no. 1, pp. 1–9, 2024.
- [14] Haeruddin, “Ketersediaan Tinggi Infrastruktur Elearning Berbasis Komputasi Awan,” *Jurnal Teknik Informatik dan Sistem Informasi*, vol. 10, no. 1, pp. 670–682, 2023.

- [15] J. Teknologi Informasi Dan Komputer Politeknik Negeri Lhokseumawe, “Analisis Kinerja Load Balancing Round Robin pada Website Skalabel,” *Journal of Information System Management (JOISM) e-ISSN*, vol. 5, no. 2, pp. 2715–3088, 2024.