

INTEGRASI *FRAMEWORK BOOTSTRAP* DAN *CHART.JS* UNTUK VISUALISASI DATA SENSOR PADA SISTEM HIDROPONIK BERBASIS *INTERNET OF THINGS (IOT)*

¹Muhammad Pasha Toppany, ²Deni Kurnia, ³Janizal, ⁴R. Priyo Hartono Adji

^{1,2,3}Teknologi Rekayasa Mekatronika, Politeknik Enjineri Indorama, Purwakarta, Indonesia

⁴Research Center for Telecommunication (PRT), National Research and Innovation Agency (BRIN), Bandung, Indonesia

¹mpasha.toppany@gmail.com, ²deni.kurnia@pei.ac.id, ³janizal@pei.ac.id, ⁴rpri001@brin.go.id

Co-author: ¹mpasha.toppany@gmail.com

Abstrak

Kurangnya penerapan teknologi menjadi salah satu hambatan dalam pengembangan sektor pertanian khususnya hidroponik. Penelitian ini bertujuan untuk mengembangkan sistem visualisasi data sensor pada sistem hidroponik dengan mengintegrasikan *framework bootstrap* dan *chart.js*. Sistem ini bertujuan untuk memperluas kemampuan aksesibilitas dan fungsionalitas dari sistem sebelumnya yang terbatas pada integrasi dengan aplikasi Android. Sistem ini melibatkan pengembangan alat monitoring yang mencakup sensor-sensor yang terpasang pada mikrokontroler ESP-32, yang tidak hanya memantau parameter tanaman hidroponik, tetapi juga menyediakan fungsi otomasi untuk pemberian pupuk. Melalui pendekatan ini, sistem mampu memberikan aksesibilitas yang lebih luas melalui berbagai platform perangkat, meningkatkan ketersediaan informasi bagi pengguna yang menggunakan perangkat selain Android. Berdasarkan hasil perancangan dan pengujian pada sistem, diperoleh kesimpulan bahwa implementasi *framework bootstrap* dan *chart.js* untuk visualisasi parameter tanaman hidroponik berhasil dilakukan dengan ditunjukkan waktu untuk memuat halaman web hanya 2 detik dan kemampuan website dalam menangani user request hanya sebesar 354 ms, serta data yang divisualisasikan dalam website sesuai dengan yang dikirim ke dalam database.

Kata kunci : ESP-32, WiFi, IoT, Hidroponik, Monitoring, Sensor.

1. PENDAHULUAN

Mengacu kepada data-data penelitian tentang tanaman hidroponik, proses pengelolaan tanaman ini masih menggunakan cara-cara tradisional, seperti pemberian pupuk, penyiraman, pengecekan kadar nutrisi, dll. Kondisi ini tentu cukup merepotkan para petani hidroponik, karena harus terus selalu melakukan aktivitas tersebut setiap waktunya, ditambah faktor *human error*, seperti kelelahan atau lupa, sehingga memicu terjadinya panen yang tidak sesuai dengan harapan [1]. Oleh karena itu, diperlukan teknologi yang memungkinkan menjadi bagian solusi terhadap permasalahan tersebut.

Saat ini, teknologi yang memungkinkan dapat membantu para petani adalah pengembangan *platform Internet of Things (IoT)* [2]. IoT adalah sebuah konsep dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer [3].

Salah satu sistem yang telah dikembangkan berkaitan dengan pengelolaan tanaman hidroponik adalah *platform* monitoring parameter tanaman menggunakan aplikasi pada OS Android [4][5]. Namun, dalam penggunaannya, sistem hanya dapat diakses melalui aplikasi OS Android, tanpa kemampuan akses melalui *website*. Sehingga hal ini membatasi fleksibilitas pengguna dalam mengakses dan monitoring data, terutama bagi pengguna yang menggunakan platform selain OS Android [6][7].

Oleh karena itu untuk meningkatkan aksesibilitas sistem monitoring perlu dilengkapi dengan *platform website*. Dengan pendekatan ini, pengguna dapat mengakses sistem monitoring dari berbagai perangkat, termasuk perangkat *android*, *iOS*, dan *desktop*, sehingga memberikan kemudahan akses yang lebih luas dan meningkatkan jangkauan pengguna secara keseluruhan.

Makalah dikirim 22 Februari 2024; Revisi 22 Maret 2024; Diterima 22 April 2024

Tujuan penelitian ini berfokus pada pengembangan *platform IoT* yang mampu menampilkan parameter-parameter secara *real-time* pada *website* menggunakan *framework bootstrap* dan *chart.js*. Adapun dipilihnya *framework* ini karena fiturnya yang dapat memenuhi kebutuhan pengembangan sistem serta dukungan *plugin JavaScript* yang membantu membuat pengembangan *interface* menjadi lebih mudah [8][9].

2. METODE PENELITIAN

Penelitian ini dilakukan melalui beberapa serangkaian proses diantaranya : analisa kebutuhan sistem, studi literatur, perancangan dan uji coba sistem.

2.1. Analisa Kebutuhan Sistem

2.1.1. Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak merupakan *tools* yang akan digunakan untuk merancang dan membangun sebuah sistem *platform IoT* berbasis *website*, adapun beberapa kebutuhan perangkat lunak yang digunakan pada sistem ini yaitu :

- 1) Bahasa pemrograman *PHP* yang akan digunakan untuk pertukaran data dengan *database*.
- 2) Menggunakan *framework bootstrap* dan *chart.js* untuk membuat *interface* visual halaman *website*.
- 3) Menggunakan *database* MySQL sebagai tempat penyimpanan data.
- 4) Baris program dibuat dengan menggunakan aplikasi *Visual Code* untuk membuat *interface website*.

2.1.2. Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang digunakan dalam membuat sistem ini yaitu :

- 1) Mikrokontroler ESP32.
- 2) Sensor suhu tipe DS18B20.
- 3) Sensor PH-4502C
- 4) Sensor TDS DFROBOT V1.0
- 5) Modul relay 2 *channel*
- 6) *Peristaltic pump*
- 7) Motor DC 12V
- 8) *Pushbutton*
- 9) *Liquid Crystal Display* (LCD)
- 10) *Buzzer*
- 11) Potensiometer

2.2. Studi Literatur

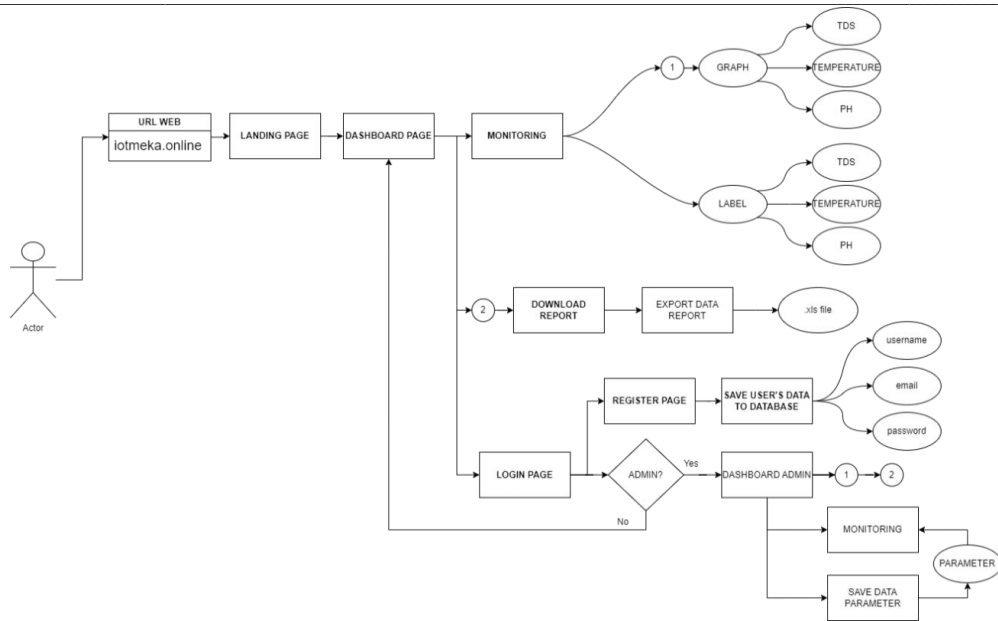
Studi literatur dilakukan untuk mendalami penggunaan *framework chart.js* dan *bootstrap* dalam pengembangan *platform interface website*. Referensi seperti dokumentasi resmi dan jurnal ilmiah digunakan untuk mendalami konsep dasar dan teknik penggunaan kedua *framework* tersebut.

Menurut penelitian yang dilakukan oleh G. Supriatmaja [10], penggunaan *framework bootstrap* menyediakan tipe, tombol, navigasi, dan elemen lain yang, bersama dengan *JavaScript*, membuat pengembangan *interface* menjadi lebih mudah. Oleh karena itu, *bootstrap* adalah salah satu *framework front-end* yang paling banyak digunakan [11]. Sementara itu, *framework chart.js* menurut Ranto [12], *framework chart.js* dapat digunakan untuk menampilkan informasi data dalam bentuk grafik sehingga lebih mudah dipahami dibandingkan menampilkan data dalam bentuk tabel.

2.3. Use Case Diagram

Use case diagram dibuat untuk memberikan pemahaman bagi pengguna terhadap sistem yang akan dibuat, dengan menggunakan perspektif sebagai *user*. *Use case diagram* tersebut dapat dilihat pada Gambar 1.

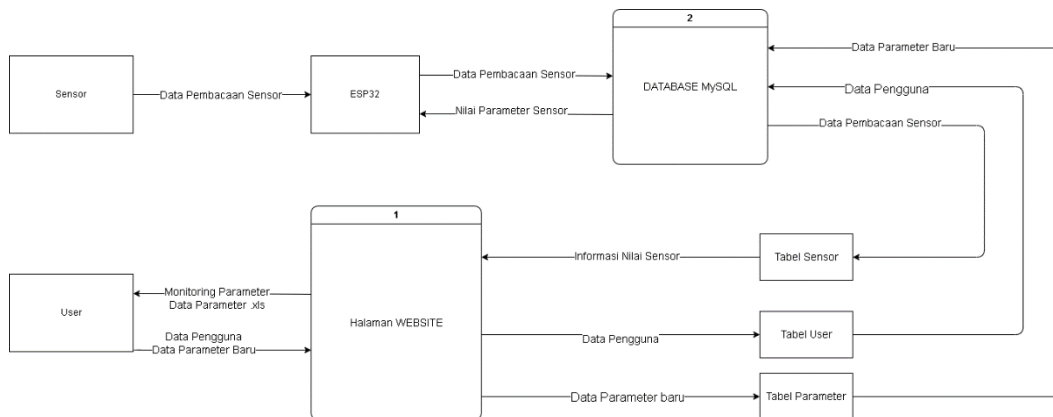
Dari *use case diagram* dapat dijelaskan bahwa pengguna dapat masuk ke *platform* monitoring yang terhubung dengan perangkat sensor pada sistem hidroponik, kemudian user dapat melihat parameter yang divisualisasikan seperti nilai suhu, pH, dan TDS air tanaman hidroponik. Selain itu, data parameter dapat diunduh oleh *user*.



Gambar 1. Use case diagram.

2.4. Data Flow Diagram

Data flow diagram digunakan untuk menunjukkan pertukaran data dari setiap proses yang terjadi pada sistem yang dibuat, hasil dari perancangan dapat dilihat pada Gambar 2.



Gambar 2. Data flow diagram system.

Gambar 2 menunjukkan setiap proses yang terjadi pada sistem secara terperinci yang membentuk aliran data. Aliran data ini diproses oleh file Hypertext Preprocessor (PHP), yang merupakan bahasa pemrograman server-side yang digunakan untuk melakukan pertukaran data dengan database melalui perintah Structured Query Language (SQL) [13]. Dengan database yang digunakan pada sistem ini adalah MySQL yang dikenal sebagai manajemen basisdata yang stabil digunakan sebagai media penyimpanan data. MySQL merupakan turunan dari SQL yang berfungsi mengatur dan mengelola data-data pada database [14].

2.5. Perancangan *Interface Website*

Perancangan *interface website* menampilkan desain visual, *layout*, elemen tombol, area parameter sensor, dan navigasi yang akan diimplementasikan dalam pengembangan halaman *website*. konsep *interface website* dapat dilihat pada Gambar 3.



Gambar 3. Perancangan *interface website*.

Gambar 3 merupakan perancangan *interface website* yang akan diimplementasikan, Gambar 3(A) merupakan konsep halaman *landing page*, Gambar 3(B) konsep halaman *monitoring dashboard*, Gambar 3(C) konsep halaman *admin dashboard*, Gambar 3(D) konsep halaman *Login Page*, Gambar 3(E) konsep halaman *register page*. Untuk implementasinya akan menggunakan *framework bootstrap* dan *chart.js*. dengan pertimbangan bahwa *framework* ini menyediakan banyak *class* dan *plugin Cascading Style Sheet (CSS)* yang siap pakai untuk membantu dalam pengembangan tampilan *interface website*.

2.6. Pengujian *Sistem Website*

2.6.1. Pengujian File PHP

Pengujian *file* PHP dilakukan untuk melihat respon *file* dalam melakukan perintah pengambilan dan penerimaan data dari *web* kedalam *database* atau sebaliknya [15]. Pengujian ini mencakup keberhasilan *file* dalam menyimpan dan memuat data yang dapat dilihat pada Tabel 1.

Tabel 1. Pengujian respon file PHP.

No	Nama File	Fungsi	Status
1.	connection.php	Menerima data sensor dari ESP32 dan menyimpan kedalam <i>database</i>	<i>File</i> berjalan dengan baik
2.	GrafikpH.php	Mengambil data sensor pH pada <i>database</i> dan dimuat dalam bentuk grafik	<i>File</i> berjalan dengan baik
3.	GrafikSuhu.php	Mengambil data sensor suhu pada <i>database</i> dan dimuat dalam bentuk grafik	<i>File</i> berjalan dengan baik
4.	Grafiktds.php	Mengambil data sensor tds pada <i>database</i> dan dimuat dalam bentuk grafik	<i>File</i> berjalan dengan baik
5.	Grafiksensor.php	Mengambil data seluruh sensor pada <i>database</i> dan dimuat dalam bentuk grafik	<i>File</i> berjalan dengan baik
6.	NilaiH.php	Mengambil data sensor pH dan ditampilkan dalam bentuk <i>value</i>	<i>File</i> berjalan dengan baik
7.	NilaiSuhu.php	Mengambil data sensor suhu dan ditampilkan dalam bentuk <i>value</i>	<i>File</i> berjalan dengan baik
8.	Nilaitds.php	Mengambil data sensor tds dan ditampilkan dalam bentuk <i>value</i>	<i>File</i> berjalan dengan baik
9.	nilaiparameter.php	Mengambil data parameter dan ditampilkan pada <i>web</i>	<i>File</i> berjalan dengan baik
10.	Export_to_excel.php	Mengambil data dari <i>database</i> dan dimuat kedalam <i>file</i> excel	<i>File</i> berjalan dengan baik
11.	delete.php	Menghapus data dari <i>database</i> <i>table</i> sensor	<i>File</i> berjalan dengan baik
12.	sendparameter.php	Mengirim nilai parameter kedalam <i>database</i>	<i>File</i> berjalan dengan baik
13.	connlog.php	Melakukan verifikasi data <i>user</i> saat melakukan <i>login</i>	<i>File</i> berjalan dengan baik
14.	connreg.php	Menyimpan data <i>user</i> dari halaman <i>register</i> kedalam <i>database</i>	<i>File</i> berjalan dengan baik
15.	Kirimparameter.php	Mengirim respon data parameter kedalam ESP32	<i>File</i> berjalan dengan baik

2.6.2. Pengujian User Handling

Pengujian ini dilakukan untuk melihat kemampuan *website* dalam mengelola permintaan *user* dalam jumlah banyak, metode yang digunakan adalah *stress test* dimana *website* akan diberikan permintaan *user* dalam jumlah banyak dalam waktu 1 menit, kemudian dilihat apakah *website* mengalami penurunan kinerja atau tidak, dengan kemampuan *bandwith* *website* sebesar 247.58 MB, hasil pengujian dapat dilihat pada Tabel 2.

Tabel 2. Hasil user handling.

No	Metode	Jumlah User	Rata-Rata Waktu Respon	Hasil Respon	Total bandwidth	Delay Memuat Halaman
1.	User per test	250	337 ms	250 Success	3.34 MB	2 Detik
2.			352 ms	250 Success	3.34 MB	
3.		500	358 ms	500 Success	6.68 MB	
4.			373 ms	500 Success	6.68 MB	
5.	User per Second	20	338 ms	1200 success	16.03 MB	
6.	user/second	user/second	368 ms	1200 success	16.03 MB	

2.6.3. Pengujian Waktu Pengiriman Data

Pengujian waktu pengiriman data dilakukan untuk melihat berapa lama waktu *delay* yang terjadi dalam proses pengiriman data dari ESP32 kedalam *database* dan *database* kedalam halaman *website*. Dalam pengujian ini ESP32 dihubungkan dengan 2 *internet provider* berbeda. Detail dari *provider* dapat dilihat pada Tabel 3 dan hasil pengujian *delay* pada komunikasi data dilihat pada Tabel 4.

Tabel 3. Data Kecepatan Internet Provider.

No	Internet Provider	Download Speed	Upload Speed	Ukuran Data Sensor Ke Database
1.	PT. Telkom Indonesia	0.21 Mbps	1.23 Mbps	±2.4 – 3 KB
2.	Hutchison 3 Indonesia	1.9 Mbps	16.8 Mbps	

Tabel 4. Delay Pada Komunikasi Data.

No	Delay Pengiriman Data ESP32	Kemampuan Proses Database	Delay Memuat Data Pada Web	Record Waktu Data Tersimpan
1.	12 Menit	1 Detik	2 Detik	08/12/2023. (19:09:33)
2.	12 Menit	1 Detik	2 Detik	08/12/2023. (19:21:33)
3.	12 Menit	1 Detik	2 Detik	08/12/2023. (19:33:36)
4.	12 Menit	1 Detik	2 Detik	08/12/2023. (19:45:37)
5.	12 Menit	1 Detik	2 Detik	08/12/2023. (19:57:38)

Berdasarkan Tabel 3 *ISP* mempunyai *upload speed* sebesar 1.23 Mbps dan 16.8 Mbps. Selanjutnya *database* menerima data berukuran 2.4-3 KB atau jika dikonversi menjadi satuan transfer data adalah 19,2 Kbps – 24 Kbps (KB x 8 bit/byte). Artinya, dengan nilai *upload speed* 1.23 Mbps dan 16.8 Mbps sudah optimal untuk melakukan pengiriman data selama sinyal dan layanan *internet* tidak terputus. Kemudian untuk rata-rata kecepatan *upload speed ISP* di Indonesia dapat dilihat pada Gambar 4.



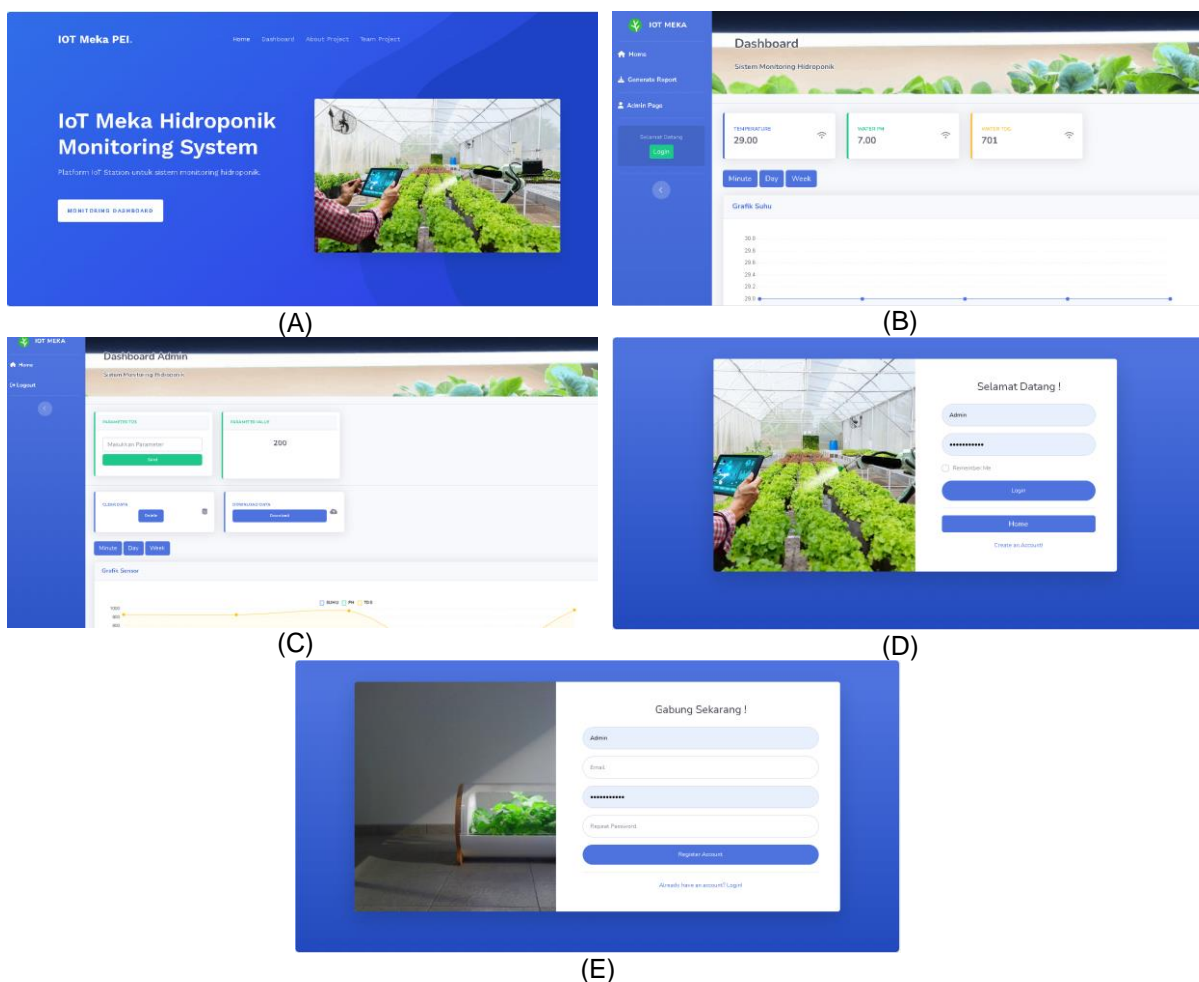
Gambar 4. Kecepatan Upload Speed ISP.

Berdasarkan grafik pada Gambar 4, kecepatan *upload* ISP di Indonesia rata-rata berada di atas 1 Mbps, sehingga semua ISP dapat digunakan untuk sistem *IoT* dalam aspek pengiriman data ke dalam *database*.

3. HASIL DAN PEMBAHASAN

3.1. Hasil Interface Website

Interface website adalah *platform* yang digunakan untuk monitoring nilai dari parameter tanaman, *user* dapat melakukan monitoring dengan mengakses alamat *domain website*, hasil dari perancangan *interface website* dapat dilihat pada Gambar 5.



Gambar 5. Hasil Rancangan Interface Website.

Gambar 5 menampilkan hasil implementasi dari perancangan *website* yang telah dibuat sebelumnya, hasil tersebut akan digunakan didalam *hosting* untuk tampilan pengguna. Adapun penjelasan halaman sebagai berikut pada Gambar 5:

- (A) Halaman *landing page* untuk *domain* *iotmeka.online* memiliki beberapa bagian utama. “Navbar” menyediakan tombol navigasi untuk mengakses halaman *dashboard*, *about project*, *team project* dan *home*. Bagian “About Project” menjelaskan tujuan dan keunggulan *platform* IoT dalam monitoring parameter sensor. “Team Project” menampilkan anggota tim yang terlibat dalam pembuatan sistem, dan “Dashboard” digunakan *user* untuk mengakses halaman monitoring.
- (B) Halaman *dashboard monitoring* digunakan untuk *user* melakukan monitoring parameter sensor pada sistem hidroponik, data divisualisasikan dalam bentuk nilai dan grafik. Pada bagian “Navbar” menyediakan tombol “Login” untuk *user* melakukan *login*, serta tombol “Generated Report” untuk *download* data parameter dalam format *excel* (.xls).
- (C) Halaman *admin dashboard* dapat diakses dengan melakukan *login* menggunakan akun *admin*. Data parameter sensor divisualisasikan dalam satu grafik dan menampilkan nilai parameter *setpoint* TDS. Admin dapat mengubah nilai *setpoint* TDS pada *form area*, dan terdapat tombol fungsi “Delete” untuk menghapus data parameter dari *database*.
- (D) Halaman *login* digunakan *user* untuk melakukan *login* akun kedalam halaman *dashboard* monitoring dan *admin dashboard*.
- (E) Halaman *register* digunakan *user* untuk mendaftarkan akun kedalam *database website*.

3.2. Hasil Pengujian Sistem Website

Uji coba *website* diperlukan untuk mengetahui kinerja dan kapasitas *website* dalam mengelola data dan *user* yang masuk kehalaman *website*, berikut merupakan serangkaian pengujian *website* yang

3.2.1. Hasil Pengujian File PHP

Berdasarkan pengujian *file* PHP pada Tabel 1, status *file* dinyatakan bekerja dengan baik. Pada bagian ini, hasil pengujian akan dijelaskan secara lebih rinci dengan melihat respons dari *file* PHP ketika dijalankan. Hasil pengujian dapat dilihat pada Tabel 5.

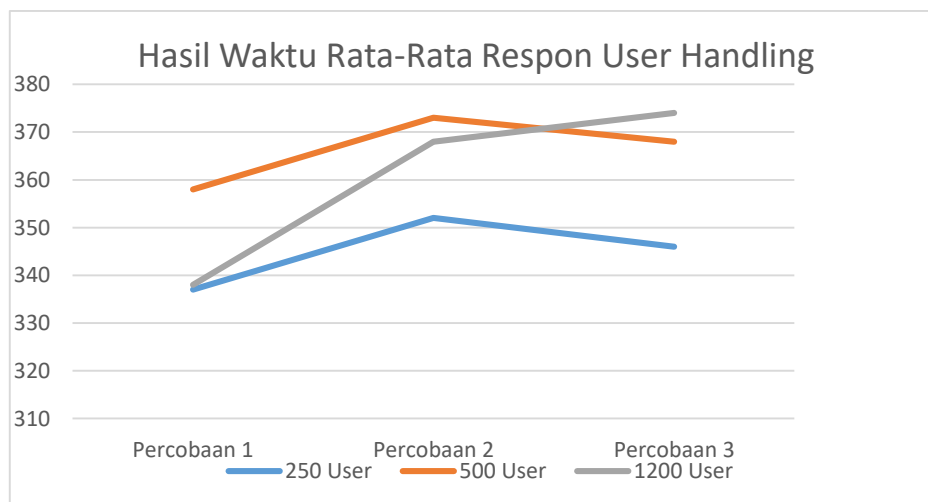
Tabel 5. Hasil respon file PHP.

No	Nama File	Trigger File	Respon File
1.	connection.php	Baris program <i>interval</i> pada ESP 32	Berhasil mengirim parameter sensor kedalam <i>database</i>
2.	GrafikpH.php	Baris program <i>interval Javascript</i> setiap 1 detik pada <i>file</i> HTML	Berhasil memuat grafik parameter pH pada <i>website</i>
3.	GrafikSuhu.php	Baris program <i>interval Javascript</i> setiap 1 detik pada <i>file</i> HTML	Berhasil memuat grafik parameter suhu pada <i>website</i>
4.	GrafiktDs.php	Baris program <i>interval Javascript</i> setiap 1 detik pada <i>file</i> HTML	Berhasil memuat grafik parameter TDS pada <i>website</i>
5.	Grafiksensor.php	Baris program <i>interval Javascript</i> setiap 1 detik pada <i>file</i> HTML	Berhasil memuat grafik parameter seluruh sensor pada <i>website</i>
6.	NilaiPpH.php	Baris program <i>interval Javascript</i> setiap 1 detik pada <i>file</i> HTML	Berhasil memuat nilai parameter pH pada <i>website</i>
7.	NilaiSuhu.php	Baris program <i>interval Javascript</i> setiap 1 detik pada <i>file</i> HTML	Berhasil memuat nilai parameter suhu pada <i>website</i>

8.	Nilaitds.php	Baris program <i>interval Javascript</i> setiap 1 detik pada <i>file HTML</i>	Berhasil memuat nilai parameter TDS pada <i>website</i>
9.	nilaiparameter.php	Baris program <i>interval Javascript</i> setiap 1 detik pada <i>file HTML</i>	Berhasil memuat nilai parameter setpoint pada <i>website</i>
10.	Export_to_excel.php	Fungsi tombol <i>generate report</i> pada <i>website</i>	Data parameter sensor <i>download</i> secara otomatis dalam bentuk <i>excel (xls)</i>
11.	delete.php	Fungsi tombol <i>delete</i> pada <i>admin dashboard</i>	Data parameter sensor didalam <i>database</i> terhapus
12.	sendparameter.php	Fungsi tombol <i>send</i> pada <i>admin dashboard</i>	Mengirim nilai parameter <i>setpoint</i> kedalam <i>database</i>
13.	connlog.php	Fungsi tombol <i>login</i> pada halaman <i>login</i>	Mengambil data <i>user</i> pada <i>form login</i> dan mencocokkan dengan data <i>user</i> pada <i>database</i>
14.	connreg.php	Fungsi tombol <i>register account</i> pada halaman <i>register</i>	Mengambil data <i>user</i> pada <i>form register</i> dan disimpan kedalam <i>database</i>
15.	Kirimparameter.php	Baris program <i>interval</i> pada <i>ESP32</i>	Mengirim data parameter <i>serpoint</i> dari <i>database</i> kepada <i>ESP32</i>

3.2.2. Hasil Pengujian User Handling

Berdasarkan hasil pengujian pada Tabel 2, *website* mampu mengelola permintaan dari *user* dengan sangat baik. Saat diuji dengan 500 *user* dalam waktu 1 menit, *website* mampu memberikan respon kepada semua *user* tanpa masalah. *Delay* dalam memuat data sensor tetap konsisten hanya sekitar 2 detik. Hal ini menunjukkan bahwa performa *website* tetap optimal tanpa mengalami *crash* atau penurunan kinerja. Rata-rata hasil respon dapat dilihat pada grafik Gambar 6.

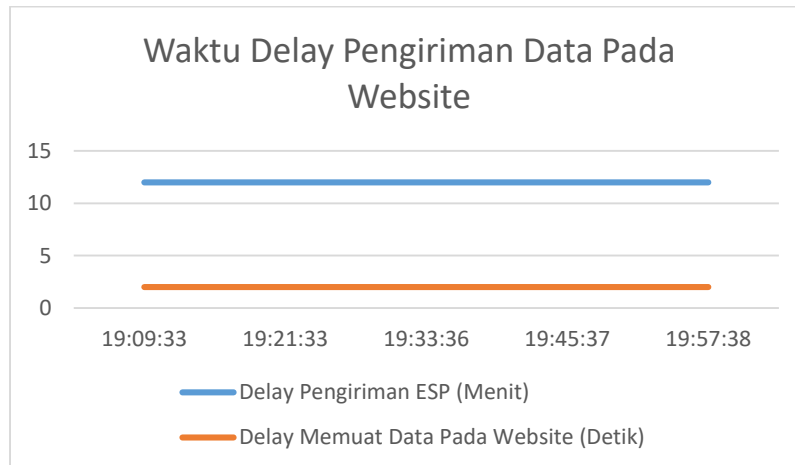


Gambar 6. Grafik rata-tata respon user handling.

3.2.3. Hasil Pengujian Waktu Pengiriman Data

Tabel 4 didapatkan hasil data diterima dan disimpan setiap 12 menit 1 detik sekali, hal ini karena *ESP-32* di-*setting* untuk melakukan pengiriman data sensor setiap 12 menit, hal ini dilakukan agar data sensor pada *database* tidak menumpuk, serta *ESP32* melakukan sampling air setiap waktu interval 10

menit, hal ini yang menjadi alasan agar data yang dikirim merupakan data parameter air yang sudah stabil. Kemudian ditambah oleh kemampuan *database* untuk memproses data *query* yang dilakukan setiap 1 detik dan fungsi *Javascript* untuk menampilkan hasil *query* pada *file* PHP dilakukan setiap interval waktu 1 detik, hal inilah yang menyebabkan proses memuat data pada *dashboard* terdapat delay selama 2 detik. Hasil akhir dapat dilihat pada grafik Gambar 7.



Gambar 7. Grafik hasil pengujian delay pada website.

4. KESIMPULAN

Berdasarkan serangkaian hasil pengujian, dapat disimpulkan bahwa sistem yang dibangun berhasil mengintegrasikan *framework Bootstrap* dan *chart.js* untuk visualisasi data sensor pada sistem hidroponik berbasis *Internet of Things* (IoT) berjalan dengan sangat baik. Pengujian menunjukkan bahwa konsep *website* telah berhasil diimplementasikan menggunakan kedua *framework* tersebut. Waktu respon terhadap permintaan pengguna yang mencapai 354 ms menandakan performa sistem yang optimal dalam menangani *user request*, selain itu konsistensi dalam memuat data parameter sensor konstan pada waktu delay 2 detik menunjukkan stabilitas sistem. Kemudian, kemampuan sistem untuk menangani *request* hingga 500 user dapat berjalan tanpa mengalami kegagalan, hal ini menunjukkan sistem memiliki skalabilitas yang baik. Dengan demikian, berdasarkan serangkaian hasil pengujian, sistem ini dianggap berhasil dalam memenuhi kebutuhan visualisasi data sensor pada sistem hidroponik berbasis IoT.

DAFTAR PUSTAKA

- [1] N. Novianti, S. Santoso Heru Hadi, and A. Aprilia, "Quality Control of Romaine Lettuce Products in Hydroponic Cultivation Systems (Case Study in MSME Vegetable Gardens, Surabaya City, East Java)," *Agrisocionomics J. Sos. Ekon. Pertan.*, vol. 3, no. 2, pp. 131–150, 2019, [Online]. Available: <https://doi.org/10.14710/agrisocionomics.v3i2.5287>
- [2] D. Kurnia, S. Riyadi, D. A. Dewi, A. A. Suprianto, and O. Rahmadani, "IoT Platform for Monitoring Systems Water pH in the Freshwater Fish Cultivation Process," vol. 16, no. 1, pp. 11–16, 2024.
- [3] C. Wibisono Darmawan, S. R. U A Sompie, and F. D. Kambey, "Implementasi Internet of Things pada Monitoring Kecepatan Kendaraan Bermotor," *J. Tek. Elektro dan Komput.*, vol. 9, no. 14, pp. 91–100, 2020.
- [4] M. Syarif, A. Bastian, and I. Mahjud, "Rancang Bangun Monitoring Nutrisi Tanaman Hidroponik Berbasis Internet of Things (IoT)," *Pros. 4th Semin. Nas. Penelit. Pengabd. Kpd. Masy. 2020*, vol. 9, no. 4, pp. 62–67, 2020.
- [5] Miftahul Walid, H. Hoiriyah, and A. Fikri, "Pengembangan Sistem Irigasi Pertanian Berbasis Internet of Things (IoT)," *J. Mnemon.*, vol. 5, no. 1, pp. 31–38, 2022, doi: 10.36040/mnemonic.v5i1.4452.
- [6] T. Hariono and L. F. Fajriyah, "Monitoring Sistem Otomatisasi Hidroponik Berbasis Mobile,"

-
- Exact Pap. Compil.*, vol. 3, no. 2, pp. 347–352, 2021.
- [7] D. Kurnia, R. Setiawan, and Janizal, "Development of IoT Systems for Fire Detection Tools Using ESP 8266 and Telegram Notifications," *Ramatekno*, vol. 3, no. 2, pp. 18–27, 2023, doi: 10.61713/jrt.v3i2.98.
- [8] A. Zakir, "Rancang Bangun Responsive Web Layout dengan Menggunakan Bootstrap Framework," no. 70, pp. 7–10.
- [9] M. F. Santoso, "Teknik Responsive Web Design Bootstrap 4 Serta Penerapannya Dalam Rancang Bangun Layout Web," *J. Pilar Nusa Mandiri*, vol. 15, no. 1, pp. 61–68, 2019, doi: 10.33480/pilar.v15i1.101.
- [10] G. A. Supriatmaja, P. M. Y. P. I, K. Mahendra, K. D. D. Widyaputra, J. Deva, and G. S. Mahendra, "Sistem Informasi Perpustakaan Menggunakan Framework Bootstrap Dengan PHP Native dan Database MySQL Berbasis Web Pada SMP Negeri 2 Dawan," *J. Teknol. Ilmu Komput.*, vol. 1, no. 1, pp. 7–15, 2022, doi: 10.56854/jtik.v1i1.30.
- [11] A. Noviantoro, A. Belinda, and R. Rahmalia, "Rancangan dan Implementasi Aplikasi Sewa Lapangan Badminton Wilayah Depok Berbasis WEB," vol. 1, no. 2, pp. 88–103, 2022.
- [12] Ranto, "Membangun Frontend Website Sistem Infomasi Pemantauan dan Peringatan Dini Bencana Tsunami Berbasis IOT dengan Menggunakan Library React Js," 2023.
- [13] R. A. Murdiyantoro, A. Izzinnahadi, and E. U. Armin, "Sistem Pemantauan Kondisi Air Hidroponik Berbasis Internet of Things Menggunakan NodeMCU ESP8266," *J. Telecommun. Electron. Control Eng.*, vol. 3, no. 2, pp. 54–61, 2021, doi: 10.20895/jtece.v3i2.258.
- [14] A. Junaidi, "Studi Perbandingan Performansi Antar Mongodb Dan Mysql Menggunakan Php Dalam Lingkungan Big Data," *Pros. Annu. Res. Semin. 2016*, vol. 2, no. 1, pp. 460–465, 2016.
- [15] K. S. Haryana, "Pengembangan Perangkat Lunak Dengan Menggunakan Php," *J. Comput. Bisnis*, vol. 2, no. 1, pp. 14–21, 2008, [Online]. Available: <http://jurnal.stmik-mi.ac.id/index.php/jcb/article/view/74>