IMPLEMENTASI DISPENSER PINTAR PENGISIAN OTOMATIS MENGGUNAKAN BASIS DATA DAN WEB SERVER BERBASIS IOT

IMPLEMENTATION OF AUTOMATIC FILLING SMART DISPENSER USING DATABASE AND WEB SERVER WITH IOTBASED

Aditya Karyadi Kusuma[1], Favian Dewanta[2], Istikmal[3]
[1,2,3] Universitas Telkom, Bandung
adityakaryadi@student.telkomuniversity.ac.id[1], favian@telkomuniversity.ac.id[2], istikmal@telkomuniversity.ac.id[3]

Abstrak

Kebutuhan air putih pada orang dewasa disarankan yaitu sekitar delapan gelas atau total 2 liter setiap harinya. Pengambilan air pada dispenser umumnya masih manual tanpa adanya fitur-fitur yang dapat memudahkan dalam pengambilan maupun pemantauan penggunaan air minum yang dikonsumsi setiap harinya. Penelitian kali ini, penulis membuat sebuah dispenser pintar yang terhubung dengan *web server* dan *database*, yang dapat memantau dan merekam aktivitas dispenser. Kemudian, pengguna dapat melihat informasi tersebut menggunakan *browser* mereka dengan mengakses *web server*. Sistem dispenser pintar yang penulis rancang berbasis *Internet Of Things* (IoT), sehingga aktivitas pengambilan air minum dapat dilihat dan dipantau secara *realtime* melalui jaringan internet. Dari hasil pengujian yang telah dilakukan, diketahui sistem dapat bekerja dengan baik. Hasil pengujian fungsionalitas, seluruh fitur yang terdapat di *website* Dispenserku dapat diakses dan digunakan dengan baik. Selain itu pengujian *Quality Of Service* juga dilakukan untuk mengetahui kualitas dari layanan yang telah dibuat. Hasil pengujian *delay* dari *client-server* mendapatkan *delay* rata-ratanya sebesar 0.138 detik dan *server-client* mendapatkan *delay* rata-ratanya sebesar 0.198 detik. Hasil pengujian *jitter* dari *client-server* mendapatkan *jitter* rata-ratanya sebesar 0.00511 ms dan *server-client* mendapatkan *jitter* rata-ratanya sebesar 0.00356 ms. Pengujian performansi yang telah dilakukan dapat disimpulkan bahwa server masih dapat berjalan dengan baik.

Kata kunci : Dispenser Pintar, *Web Server*, *Website*, Air Minum, MySQL.

Abstract

The recommended water requirement for adults is about eight glasses or a total of 2 litres per-day. In this study, the author makes a smart dispenser that is connected to a web-server and database, which can monitor and record dispenser activity. Then, users can query information using their browser by accessing the web-server. The smart dispenser system that the author designed is based on the Internet of Things (IoT), so that drinking water extraction activities can be seen and monitored in real-time via the internet network. From the results of the tests that have been carried out, it is known that the system can work well. The functionality test results show that all the features on the Dispenserku website can be accessed and used properly. In addition, Quality of Service testing is also carried out. The delay test results from client-server show an average delay of 0.138 seconds and server-client shows an average delay of 0.198 seconds. The results of the client-server jitter test show an average jitter of 0.00511 ms and the server-client shows an average jitter of 0.00356 ms. In the load testing that has been done, it can be concluded that the server can still run well.

Keywords*: Smart Dispenser, Web Server, Website, Drinking Water, MySQL.*

1. Pendahuluan

Indonesia telah mengalami perkembangan teknologi yang tentunya sudah semakin pesat dengan seiring berjalannya perkembangan teknologi di dunia yang juga semakin canggih salah satu contohnnya adalah IoT atau *Internet of Things*. IoT mengolah berbagai data yang telah dicari dan dikumpulkan yang kemudian nantinya data tersebut diolah menjadi data lainnya yang lebih bermanfaat. Jadi perkembangan antara revolusi industri 4.0 dan *Internet of Things* di Indonesia saling berkaitan dan berhubungan[1]. Salah satu perkembangan pada IoT adalah *Smart Home. Smart home* adalah salah satu dari sistem pengendali rumah yang memberikan kenyamanan kepada pemilik rumah untuk mengendalikan peralatan elektronik menggunakan *android*. Konsep dari *smart home* adalah sebuah sistem yang ditujukan untuk rumah agar kita dapat tinggal dengan nyaman. Konsep ini dapat diterapkan dengan mengatur peralatan elektronik pada rumah kita[2]

Dispenser pintar ini adalah sebuah sistem berbasis IoT yang dibangun dengan tujuan memberikan kemudahan kepada masyarakat yang sering lupa kapan dan berapa banyak seseorang telah melakukan pengambilan air minum pada hari tersebut. Dispenser pintar ini dibangun dengan menggunakan mikrokontroler, sensor-sensor seperti *Water Flow Sensor, Fingerprint Sensor* dan *Ultrasonic Sensor*. Dari sensor yang telah disebutkan, pengguna dispenser pintar ini mendapatkan data-data seperti siapa dan berapa jumlah air yang dikonsumsi setiap harinya. Data yang telah didapatkan tersebut, kemudian disimpan ke dalam *database* berbasis *web server* yang selanjutnya dapat dikendalikan oleh pengelola menggunakan *website* yang sebelumnya telah dirancang. Pengguna dapat melihat data tersebut di dalam *website* dengan tujuan agar terhindar dari dehidrasi.

Diharapkan alat ini dapat membantu untuk mengingatkan bahwa pentingnya memenuhi kebutuhan asupan air minum, dan memberikan kemudahan kepada masyarakat yang terkadang sering lupa kapan waktu pengambilan air minum dan berapa banyak jumlah pengambilan yang telah dilakukan pada hari tersebut. Alat ini juga dapat memudahkan pengguna dalam pengambilan air minum di rumah dari berbagai fitur-fitur yang dimiliki oleh alat ini. Pengguna dapat menggunakan *website* untuk membantu memantau jumlah ideal air yang dibutuhkan dalam sehari-hari. *Website* tersebut terhubung dengan alat ini dan menyimpan datanya di *database* yang telah disediakan sehingga pengguna tidak perlu khawatir tentang berapa jumlah air minum yang dibutuhkan dalam sehari-hari.

2. Dasar Teori

2.1 Air

Air merupakan sumber kehidupan bagi semua makhluk hidup di dunia, mulai dari mikroorganisme hingga manusia. Air dapat diperoleh dari berbagai sumber di bumi oleh makhluk hidup seperti air hujan, air permukaan, air tanah, dan air laut. Orang membutuhkan air minum per hari sekitar 2 liter atau 8 gelas setiap hari untuk menjaga kesehatan tubuhnya[3].

2.2 Dispenser

Dispenser merupakan sebuah alat yang digunakan untuk mengalirkan air minum dari galon ke dalam gelas atau wadah. Awal mulanya fungsi dispenser diciptakan seperti itu, namun seiring dengan perkembangan zaman fungsi dispenser untuk saat ini menjadi lebih beragam di antaranya untuk memanaskan atau mendinginkan air minum sesuai dengan kebutuhan[4].

2.3 *Internet of Things*

Internet of Things (IoT) merupakan suatu jaringan yang memiliki identitas untuk mengenalinya serta memiliki alamat IP yang dapat menghubungkan antar perangkat yang saling terkoneksi ke internet secara terus-menerus, sehingga antar perangkat dapat saling berkomunikasi dan bertukar informasi[5].

2.4 *Database*

Basis data atau *database* adalah suatu kumpulan informasi yang dapat diklasifikasikan dan disimpan di

dalam komputer sehingga dapat dicari dan diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut[6].

2.5 *Web Server*

Web server adalah sebuah perangkat lunak yang menjadi penyambung untuk jaringan dunia pertama (www) yang didirikan sekitar tahun 1980-an. *Web server* digunakan untuk melakukan proses komunikasi satu arah antara *client* dengan server, klien dapat menggunakan *web browser* seperti chrome, firefox dan *browser* lainnya, kemudian *web server* akan merespons permintaan tersebut dengan hasil proses berupa data yang ingin ditampilkan ke dalam *browser*[7].

2.6 *Website*

Website atau situs web merupakan kumpulan halaman yang berisikan kumpulan dokumen yang dapat memuat teks, gambar, suara, animasi maupun video di dalam situs web menggunakan protokol HTTP dan untuk bisa menggunakan situs web diperlukan perangkat lunak yang disebut dengan *browser*[8].

2.7 *Restful Webservice*

Restful webservice merupakan salah satu contoh pengaplikasian di antara pengaplikasian lainnya dari API (*Application Programming Interface*), protokol komunikasi yang dapat bekerja pada *restful webservice* biasanya menggunakan HTTP *(Hypertext Transfer Protocol*), format yang dikeluarkan oleh *restful webservice* adalah berupa JSON, yang bertujuan untuk memudahkan penerima dalam proses membaca dan memahami konten[9].

2.8 *Quality Of Service*

Quality of service atau kualitas layanan adalah kemampuan dari suatu layanan yang mengelola lalu lintas daya dan menjamin suatu performansi jaringan yang merupakan sebuah parameter untuk mengukur kualitas dari layanan tersebut. Dalam pengukuran kali ini yang menjadi parameter untuk dianalisis adalah *delay* dan *jitter*[10].

2.8.1 *Throughput*

Throughput merupakan kecepatan pengiriman data yang diukur dalam bit per detik (bps). *Throughput* adalah parameter dari QoS yang menunjukkan dari kecepatan rata-rata *bandwidth* yang aktual, diukur dengan satuan waktu dalam kondisi jaringan untuk pengiriman paket dengan ukuran tertentu [10].

Tabel 2.1 Kategori *Throughput*

Kategori *Throughput*	*Throughput*	Indeks
Sangat Bagus	>2,1 Mbps	4
Bagus	1200 kbps – 2,1 Mbps	3
Sedang	700 – 1200 kbps	2
Jelek	338 – 700 kbps	1
Sangat Jelek	0 – 338 kbps	0

Tabel 2.1 di atas merupakan kategori *throughput* yang dapat dihitung nilai tersebut dengan menggunakan persamaan sebagai berikut:

$$Throughput = \frac{Paket\ Data\ Yang\ Diterima}{Lama\ Waktu\ Pengamatan}$$

2.8.2 *Delay*

Delay adalah lamanya waktu yang dibutuhkan data atau paket untuk menempuh jarak dari asal ke tujuan pada sistem *end to end service*. *Delay* dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama[10].

Tabel 2.2 Kategori *Delay*[10]

Kategori *Delay*	Besar *Delay* (ms)	Indeks
Sangat Bagus	< 150 ms	4
Bagus	150 ms s/d 300 ms	3
Sedang	300 ms s/d 450 ms	2
Jelek	> 450 ms	1

Tabel 2.2 di atas merupakan kategori delay yang dapat dihitung nilai tersebut dengan menggunakan persamaan sebagai berikut:

$$Delay = \frac{Panjang\ paket}{Link\ bandwith}$$

2.8.3 *Jitter*

Jitter yang lazimnya juga disebut dengan variasi *delay* merupakan variasi kedatangan paket yang diakibatkan oleh variasi dalam panjang antrean, dalam waktu pengolahan data, dan juga dalam waktu penghimpunan ulang paket-paket di akhir perjalanan *jitter*. *Jitter* masih berhubungan dengan *delay*, yang menunjukkan banyaknya variasi *delay* pada transmisi data di jaringan. Berikut adalah nilai standardisasi yang menunjukkan bahwa dapat dikatakan baik atau tidaknya[11].

Tabel 2.3 Kategori *Jitter* [11]

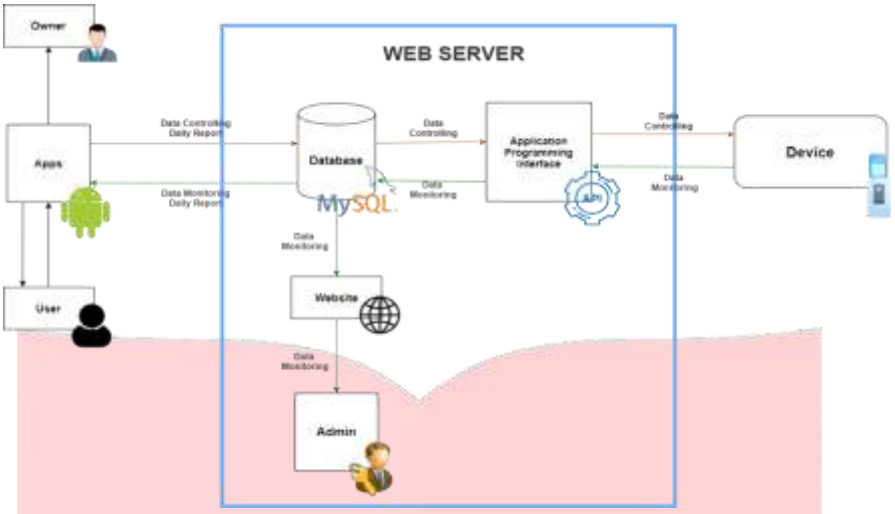
Kategori *Jitter*	*Jitter* (ms)	Indeks
Sangat Bagus	0 ms	4
Bagus	0 ms s/d 75 ms	3
Sedang	75 s/d 125 ms	2
Jelek	125 ms s/d 225 ms	1

Tabel 2.3 di atas merupakan kategori *jitter* yang dapat dihitung nilai tersebut dengan menggunakan persamaan sebagai berikut:

$$Jitter = \frac{\text{Total Variasi } delay}{\text{Total paket yang diterima}}$$

3 Model Sistem dan Perancangan

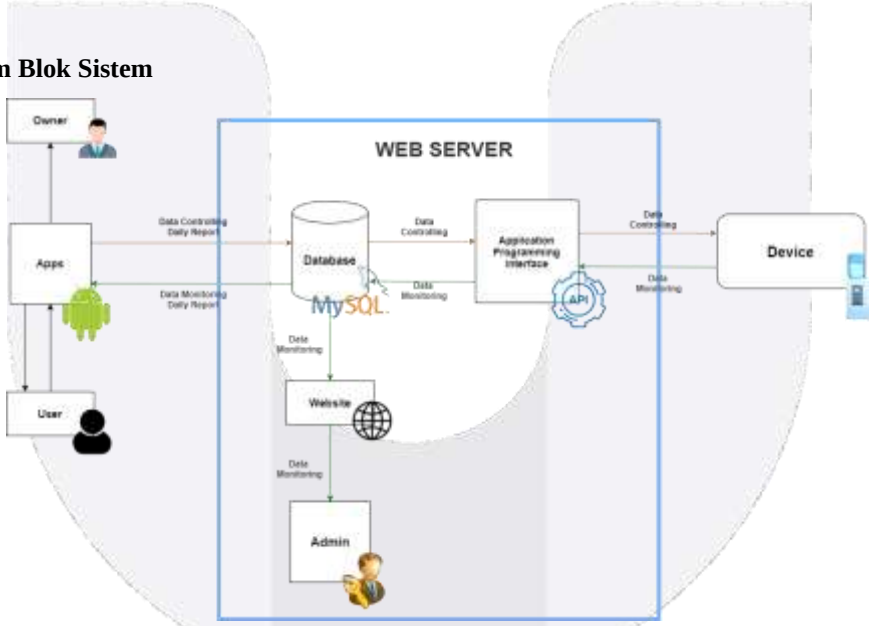
3.1 Desain Sistem



Gambar 3.1 Desain sistem

Desain sistem ini dibagi menjadi dua tahapan yaitu, tahap pertama adalah data dari sistem menggunakan protokol HTTP diteruskan menggunakan modul ESP8266 yang sudah terhubung ke internet yang nantinya dapat diakses oleh platform. Tahapan kedua adalah data yang masuk dikirim ke *web server*. Data yang diambil dari *web server* ditampilkan melalui *website* yang dibuat sehingga pengelola dapat mengaksesnya menggunakan jaringan internet.

3.1.1 Diagram Blok Sistem

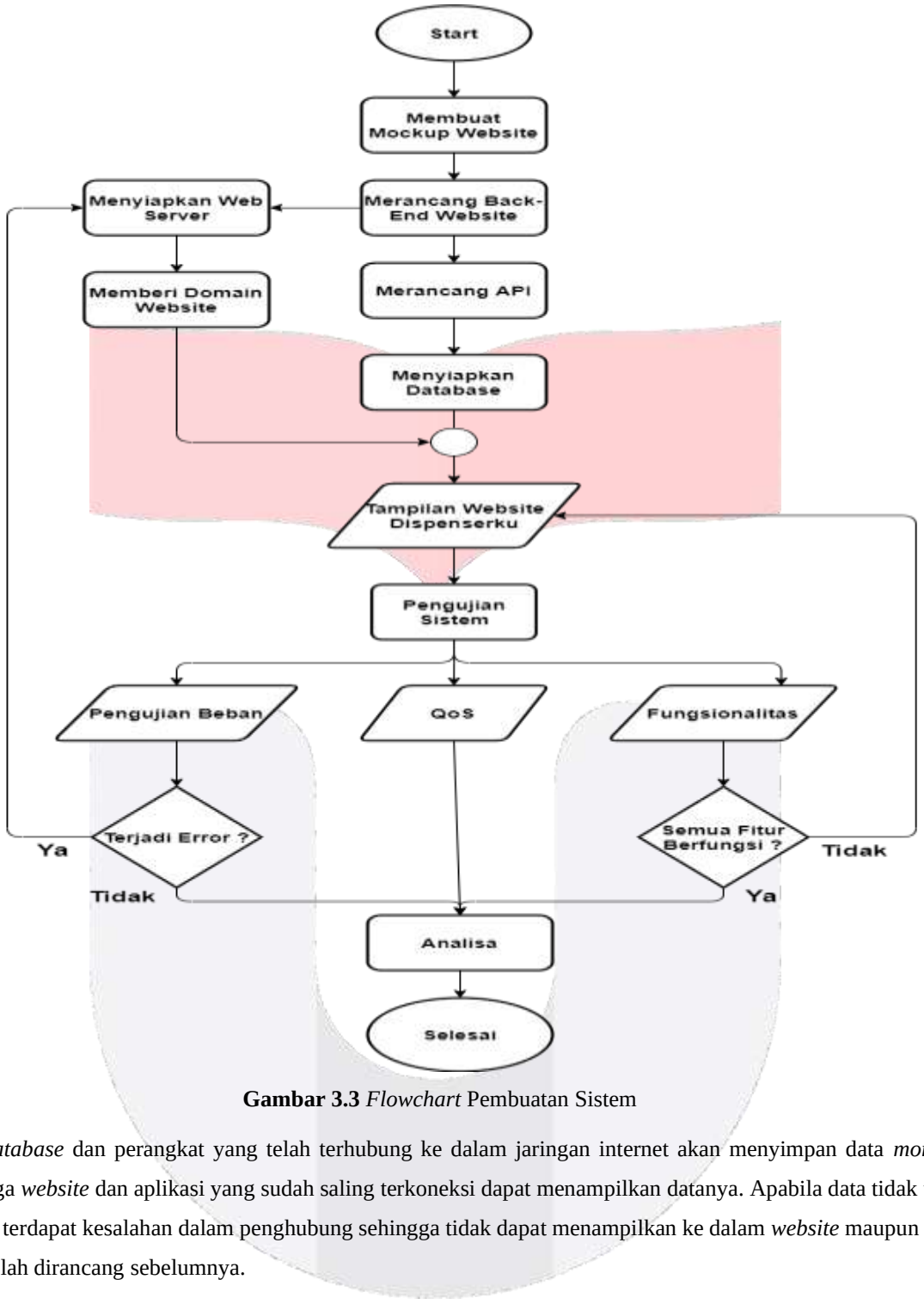


Gambar 3.2 Diagram Blok Sistem

Data sensor perangkat merupakan data *monitoring* yang akan dikirimkan ke API yang terhubung dengan jaringan internet. Data yang diterima dari API yang akan diteruskan dan dibaca oleh *database*, akan disimpan dalam bentuk SQL. Sebelum dikirim ke *database*, data akan dikelola oleh API agar data *monitoring* tidak tergabung sehingga pemantauan pengambilan air minum dapat dilakukan dengan mudah.

Selanjutnya data *monitoring* ditampilkan ke dalam *website* yang telah dirancang sebelumnya, sehingga mengetahui kapan dan berapa jumlah data pengambilan yang telah dilakukan. Data tersebut ditampilkan dalam bentuk tabel sehingga memudahkan pengguna untuk melihat hasil yang diperoleh. Hak akses untuk *website* ini dapat digunakan oleh pengelola dan pengguna.

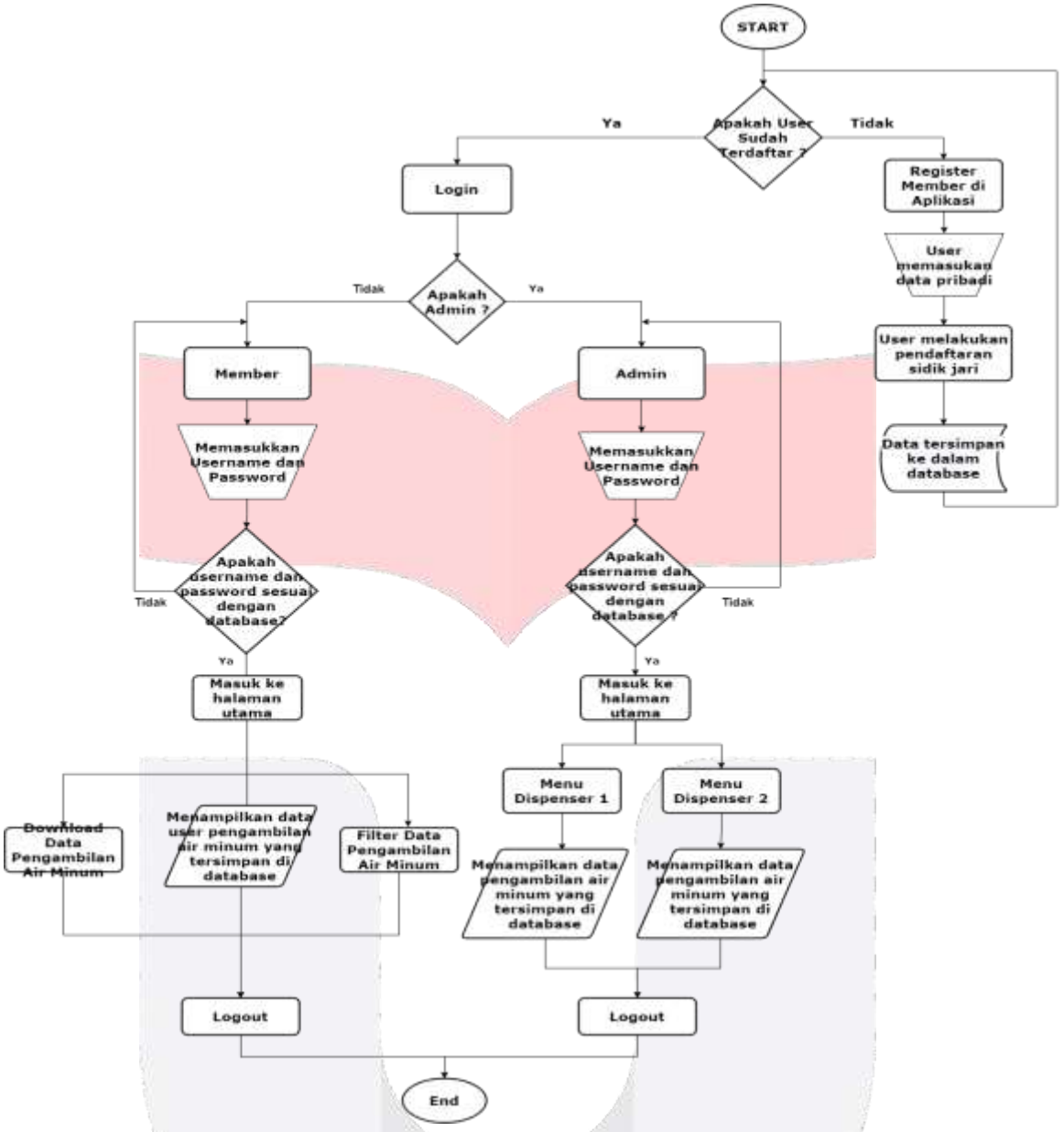
3.1.2 Diagram Alir Pembuatan Sistem



Gambar 3.3 *Flowchart* Pembuatan Sistem

Database dan perangkat yang telah terhubung ke dalam jaringan internet akan menyimpan data *monitoring* sehingga *website* dan aplikasi yang sudah saling terkoneksi dapat menampilkan datanya. Apabila data tidak terbaca, artinya terdapat kesalahan dalam penghubung sehingga tidak dapat menampilkan ke dalam *website* maupun aplikasi yang telah dirancang sebelumnya.

3.1.3 Diagram Alir Keseluruhan Sistem



Gambar 3.4 *Flowchart* Keseluruhan Sistem

Gambar 3.4 di atas menjelaskan tentang pengolahan data pada sistem pemantauan pengambilan air minum yang bekerja sehingga dapat ditampilkan ke dalam *website*. Pengguna masuk ke dalam halaman *website* yang telah dibuat. Jika pengguna belum memiliki akun, pengguna melakukan pendaftaran terlebih dahulu pada aplikasi dan mendaftarkan sidik jari pada alat. Setelah berhasil melakukan *login*, *website* akan menampilkan data pengambilan air minum yang telah tersimpan di dalam *database* sesuai dengan data pengguna yang telah diisi pada *form login* sebelumnya.

3.2 Desain Web

Website yang dirancang ini berbasis html yang berarti dapat diakses di semua *web browser* dan diberi domain dan *hosting* khusus supaya bisa diakses kapan dan di mana saja. Melalui *website* ini pengelola dapat menyimpan data-data dari *device* dan memantau agar menciptakan kondisi untuk kebutuhan air minum manusia setiap harinya.

3.3 *Use Case Diagram*

Di dalam web, pengguna dapat melihat data pengambilan air minum dengan lebih detail serta lebih terfokus agar pengguna dapat memiliki pola minum yang ideal setiap harinya agar terhindar dari dehidrasi.

3.4 *Database*

Database digunakan untuk menampilkan data pengambilan air minum dan menyimpan data untuk keperluan pengujian. Data yang akan disimpan untuk pengujian ditampilkan dalam bentuk histori dan data yang telah disimpan di histori akan diolah kemudian disimpan dalam bentuk *user*.

4 Analisis dan Hasil

4.1 Pengujian Fungsionalitas

4.1.1 Pengujian Halaman *Login*

Tabel 4.1 Fungsionalitas *Login*

Pengujian	Test step	Keterangan	Hasil
Administrator masuk ke halaman *login*	Administrator memasukan *user*name dan pasword	Administrator masuk ke menu utama	Valid
	Administrator mengosongkan kolom *user*name dan password	Muncul notif "*User*name atau password anda salah!"	
	Melakukan SQL *Injection*	Muncul notif "*User*name atau password anda salah!"	

Berdasarkan hasil dari Tabel 4.1 di atas dan dari *use case diagram* yang telah dibuat sebelumnya, dapat disimpulkan bahwa administrator dapat masuk ke menu utama dan fitur kolom *username* dan *password* berfungsi dengan baik.

4.1.2 Pengujian Halaman Utama

Tabel 4.2 Fungsionalitas Halaman Utama

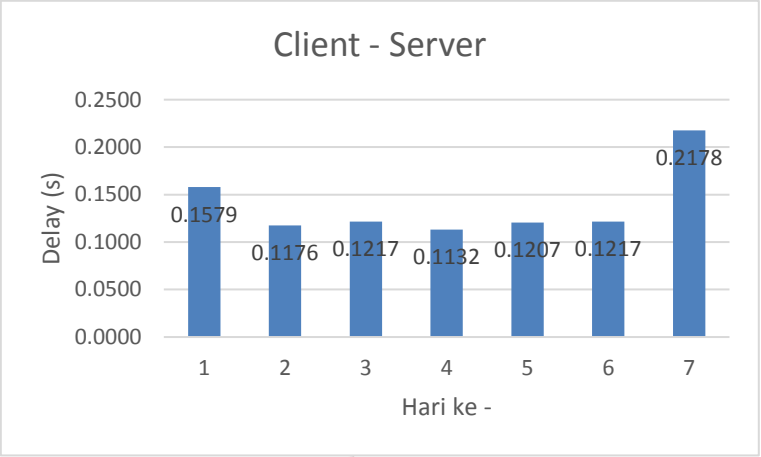
Pengujian	**Test step**	**Keterangan**	**Hasil**
Halaman menampilkan halaman *dashboard*	Mengakses halaman utama	Berhasil menampilkan halaman *dashboard*	Valid
Halaman utama menampikan dan menekan *icon logout*	Mengakses halaman utama	Halaman utama berhasil menampilkan *icon logout*	Valid
	Administrator menekan *icon logout* untuk keluar	Berhasil keluar dari halaman utama dan kembali ke halaman *login*	
Halaman utama menampilkan menu *dashboard* dan administrator menekan menu dispenser 1	Mengakses halaman utama	Halaman utama berhasil menampilkan menu *dashboard*	Valid
	Administrator menekan menu *dashboard*	Berhasil mengakses halaman dispenser 1	Valid
	Administrator menekan menu detail pengambilan air	Berhasil mengakses dan menampilkan data pengguna dispenser 1	Valid
Halaman utama menampilkan menu *dashboard* dan administrator menekan menu dispenser 2	Mengakses halaman utama	Halaman utama berhasil menampilkan menu *dashboard*	Valid
	Administrator menekan menu *dashboard*	Berhasil mengakses halaman dispenser 2	Valid
	Administrator menekan menu detail pengambilan air	Berhasil mengakses dan menampilkan data pengguna dispenser 1	Valid
Halaman utama menampilkan *header* "Smart Dispenser"	Mengakses menu utama	Berhasil menampilkan *header* "Smart Dispenser"	Valid

Tabel 4.2 di atas dapat disimpulkan bahwa seluruh fitur yang terdapat pada menu dan tampilan pada menu utama dapat berjalan dengan baik. Administrator dapat mengakses menu selanjutnya atau keluar dari menu utama.

4.2 Pengujian QoS

Pengujian dengan mengukur performa yang dilakukan dengan menguji QoS dari *client-server* dan *server-client*. Parameter QoS yang diuji adalah *delay* dan *jitter*.

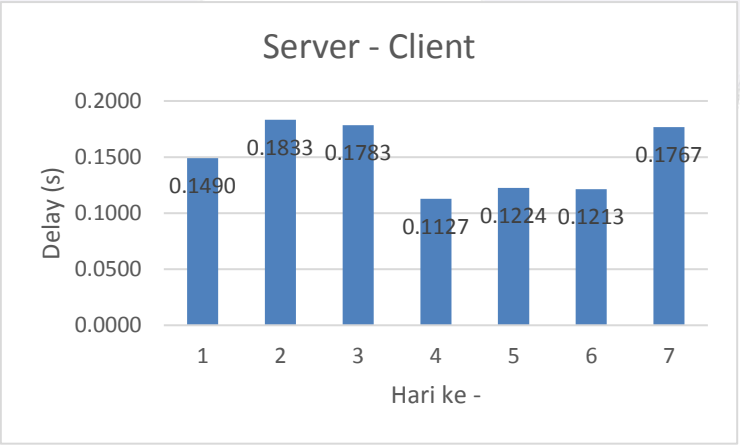
4.2.1 *Pengujian Delay Client - Server*



Gambar 4.1 *Delay Client-Server*

Gambar 4.1 di atas merupakan pengambilan data pengujian *delay* dengan cara diambil selama 7 hari dimulai pada hari Senin hingga Minggu di mana terdapat 10 *sample* setiap harinya. Pengujian mulai dilakukan pada pukul 09.00 hingga pukul 18.00 di mana setiap jamnya dilakukan pengujian selama 5 menit dan kemudian dari 10 *sample* yang didapatkan setiap harinya akan dihitung *delay* rata-ratanya. *Delay* terkecil terjadi pada hari keempat dengan nilai yang diperoleh sebesar 0.1132 detik. Lalu *delay* terbesar terjadi pada hari terakhir sebesar 0.2178 detik dan mendapatkan *delay* rata-ratanya sebesar 0.1387 detik.

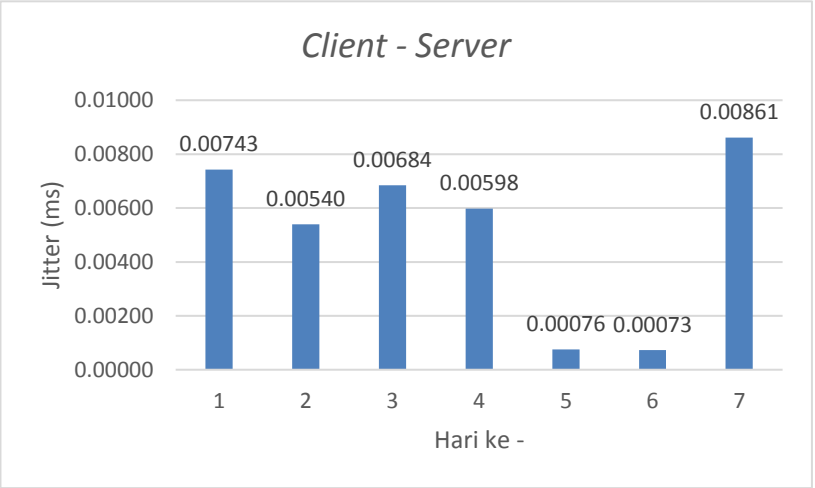
4.2.2 *Pengujian Delay Server-Client*



Gambar 4.2 *Delay Server-Client*

Gambar 4.2 di atas merupakan pengambilan data pengujian *delay* dengan cara diambil selama 7 hari dimulai pada hari Senin hingga Minggu di mana terdapat 10 *sample* setiap harinya. Pengujian mulai dilakukan pada pukul 09.00 hingga pukul 18.00 di mana setiap jamnya dilakukan pengujian selama 5 menit dan kemudian dari 10 *sample* yang didapatkan setiap harinya akan dihitung *delay* rata-ratanya. *Delay* terkecil terjadi pada hari keempat dengan nilai yang diperoleh sebesar 0.1127 detik. Lalu *delay* terbesar terjadi pada hari kedua sebesar 0.1833 detik dan mendapatkan *delay* rata-ratanya sebesar 0.1491 detik.

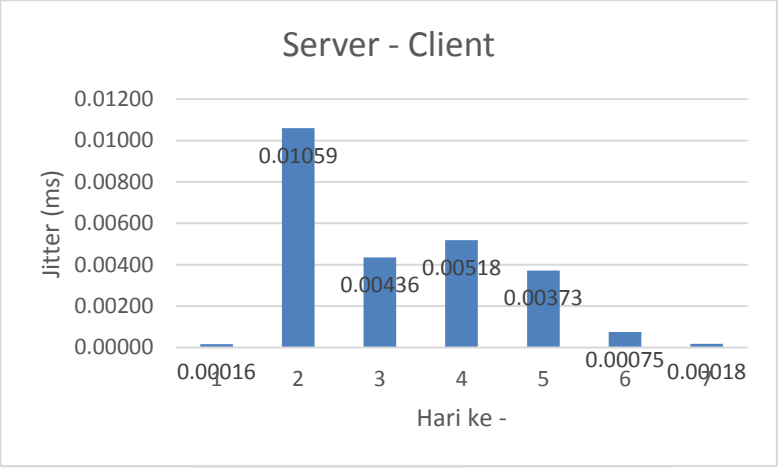
4.2.3 *Pengujian Jitter Client – Server*



Gambar 4.3 *Jitter Client – Server*

Gambar 4.3 di atas merupakan pengambilan data pengujian *jitter* dengan cara diambil selama 7 hari dimulai pada hari Senin hingga Minggu di mana terdapat 10 *sample* setiap harinya. Pengujian mulai dilakukan pada pukul 09.00 hingga pukul 18.00 di mana setiap jamnya dilakukan pengujian selama 5 menit dan kemudian dari 10 *sample* yang didapatkan setiap harinya akan dihitung *jitter* rata-ratanya. Didapatkan hasil terbesar rata-rata setiap harinya pada hari terakhir sebesar 0,00861 ms, hasil terkecil rata-rata setiap harinya pada hari keenam sebesar 0,00073 ms, dan hasil rata-rata seluruhnya adalah sebesar 0,00511 ms.

4.2.4 *Pengujian* Jitter Server – Client



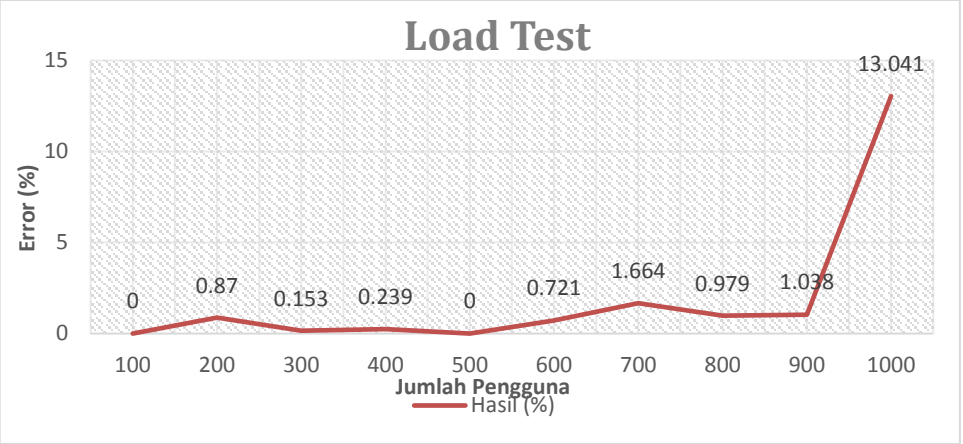
Gambar 4.4 *Jitter Server-Client*

Gambar 4.4 di atas merupakan pengambilan data pengujian *jitter* dengan cara diambil selama 7 hari dimulai pada hari Senin hingga Minggu di mana terdapat 10 *sample* setiap harinya. Pengujian mulai dilakukan pada pukul 09.00 hingga pukul 18.00 di mana setiap jamnya dilakukan pengujian selama 5 menit dan kemudian dari 10 *sample* yang didapatkan setiap harinya akan dihitung *jitter* rata-ratanya. Didapatkan hasil terbesar rata-rata setiap harinya pada hari kedua sebesar 0,01059 ms, hasil terkecil rata-rata setiap harinya pada hari pertama sebesar 0,00016 ms, dan hasil rata-rata seluruhnya adalah sebesar 0,00356 ms.

4.3 Pengujian Performansi Web

Pembahasan kali ini menjelaskan bagaimana mengukur tingkat kualitas dari suatu aplikasi web dengan menggunakan aplikasi *open source* Jmeter Apache dengan berbasis Java yang dirancang untuk memuat perilaku-perilaku fungsionalitas tes dan mengukur kinerja. Berikut adalah beberapa pengujian yang telah penulis lakukan.

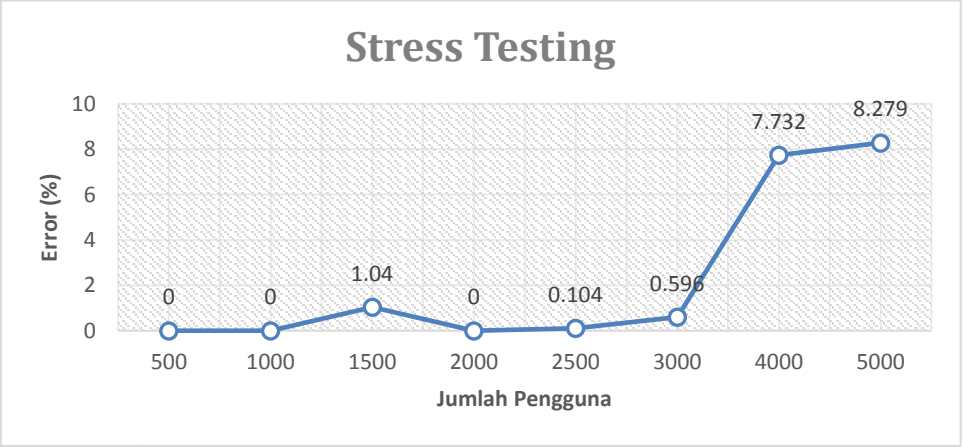
4.3.1 *Load Testing* (Pengujian Beban)



Gambar 4.5 *Load Testing*

Gambar 4.5 di atas merupakan data pengujian yang telah didapatkan. Diketahui bahwa nilai persentase *error* terkecil adalah 0% dengan jumlah pengguna 100 dan 500, sedangkan nilai persentase *error* terbesar adalah 13,041% pada saat 1000 *user* mengakses web secara bersamaan dengan *range* waktu sekitar 5 menit dan memiliki nilai rata-rata persentase *error* sebesar 1,8705%. Hasil yang didapatkan pada saat pengujian beban yang telah dilakukan adalah server dapat menangani pengakses aplikasi web tersebut yang mengakses secara bersama-sama sejumlah yang ditargetkan dengan mendapatkan hasil *error* atau kesalahan dengan beban menumpuk pada bagian halaman *login* dan *logout* tetapi server masih dapat berjalan dengan baik.

4.3.2 *Stress Testing*



Gambar 4.6 *Stress Testing*

Penulis mendapatkan hasil dengan tidak ada kesalahan pada pengguna sebanyak 500, 1000, dan 2000. Kemudian pada pengujian ini mendapatkan nilai persentase *error* terbesar adalah 8,279% pada saat 5000 pengguna melakukan permintaan terhadap server dengan mendapatkan seluruh nilai persentase *error* rata-rata sebesar 1,8705% untuk pengujian beban server. Dari hasil yang didapatkan server masih dapat bekerja dengan baik dengan banyaknya pengguna yang melakukan permintaan terhadap server.

5 Penutup

5.1 Kesimpulan

Keseluruhan sistem penelitian ini mengenai pemantauan pengambilan air minum berbasis Internet of Things (IoT) ini dapat berfungsi dengan optimal dan baik. Berdasarkan perancangan sistem yang penulis buat serta hasil dari pengujian dan analisis, penulis mengambil kesimpulan sebagai berikut:

1. Sensor dapat mengirim informasi dengan akurat dan sesuai fungsinya, *web server* sebagai penghubung antara nodeMCU dengan *database* dapat mengirim data dengan maksimal dan baik.
2. Sistem pada *web server* untuk melakukan proses *monitoring* terhadap dispenser yang penulis sudah implementasikan dapat berjalan dengan baik sehingga *website* sudah bisa menampilkan data-data yang disimpan di dalam *database.* Pengguna dapat dengan mudah untuk melakukan proses pemantauan terhadap pengambilan air minum.
3. Pembuatan *database* berdasarkan pada *database management system* (DBMS) sudah berhasil terbentuk dengan bentuk format *structured query language* (SQL).
4. Data diolah berdasarkan pada kondisi di tiap dispenser 1 dan dispenser 2.
5. Hasil pengujian fungsionalitas dari website yang sudah penulis buat dapat berjalan dengan baik dan seluruh fitur di dalamnya dapat berjalan 100%.
6. Hasil pengujian Quality of Service (QoS) *client–server* mendapatkan *delay* rata-rata sebesar 0.147 detik. Mendapatkan nilai *jitter* rata-rata sebesar 0.00511 ms. Hasil pengujian QoS *server–client* mendapatkan *delay* rata-rata yang dihasilkan sebesar 0.206 detik. Mendapatkan keseluruhan nilai *jitter* rata-rata sebesar 0.00356 ms. Data yang diperoleh menurut standardisasi TIPHON termasuk ke dalam kategori sangat baik dengan nilai indeks 4.
7. Hasil pengujian performansi web yang telah dilakukan pada *website* untuk menguji kemampuan kinerja suatu server dengan melakukan *stress testing* dan *load testing* diperoleh data tidak terjadi kesalahan pada server hingga pada saat pengujian. Hasil ketahanan dan penanganan kesalahan pada sistem dalam kondisi beban penggunaan yang sangat berat didapatkan server masih dapat bekerja dengan baik dengan banyaknya pengguna yang melakukan permintaan terhadap *server*. Dan pada pengujian *response time* didapatkan hasil pengujian bahwa sistem yang bekerja pada *website* ini berjalan dengan sangat baik.

5.2 Saran

Dari hasil penelitian tugas akhir yang telah didapatkan dan untuk meningkatkan performa *web server* dalam pemantauan pengambilan air minum, dengan ini penulis memberikan saran untuk pengembangan sistem ataupun penelitian selanjutnya yaitu:

1. Menyediakan *web server* dengan spesifikasi yang lebih tinggi dan baik lagi agar kemampuan kinerja suatu server dapat menampung jumlah pengguna yang lebih banyak sehingga tidak terjadi kesalahan yang dapat menyebabkan ketidaknyamanan pengguna.
2. Membuat tampilan *website* lebih menarik lagi dan menambahkan fitur sehingga pengguna dapat merasakan manfaat yang lebih baik lagi.

REFERENSI

[1] D. M. Hernandez, G. Peralta, L. Manero, R. Gomez, J. Bilbao, and C. Zubia, "Energy and coverage study of LPWAN schemes for Industry 4.0," *Proc. 2017 IEEE Int. Work. Electron. Control. Meas. Signals their Appl. to Mechatronics, ECMSM 2017*, 2017, doi: 10.1109/ECMSM.2017.7945893.

[2] M. Muslihudin, W. Renvilia, Taufiq, A. Andoyo, and F. Susanto, "Implementasi Aplikasi Rumah Pintar Berbasis Android Dengan Arduino Microcontroller," *J. Keteknikan dan Sains*, vol. 1, no. 1, pp. 23–31, 2018.

[3] T. Susana, "Air Sebagai Sumber Kehidupan," *Oseana*, vol. 28, no. 3, pp. 17–25, 2003, [Online]. Available: www.oseanografi.lipi.go.id.

[4] I. Oktariawan, M. Martinus, and S. Sugiyanto, "Pembuatan Sistem Otomasi Dispenser Menggunakan Mikrokontroler Arduino Mega 2560," *J. Ilm. Tek. Mesin FEMA*, vol. 1, no. 2, pp. 18–24, 2013.

[5] Ernita Dewi Meutia, "Internet of things – Keamanan dan Privasi," *Semin. Nas. dan Expo Tek. Elektro*, vol. 1, no. 1, pp. 85–89, 2015.

[6] A. Andaru, "Pengertian database secara umum," *OSF Prepr.*, vol. 1, pp. 1–6, 2018.

[7] E. Nurmiati, "Analisis dan Perancangan Web Server Pada Handphone," *Stud. Inform. J. Sist. Inf.*, vol. 5, no. 2, pp. 1–17, 2012.

[8] P. S. Hasugian, "Perancangan Website Sebagai Media Promosi Dan Informasi," *J. Inform. Pelita Nusant.*, vol. 3, no. 1, pp. 82–86, 2018.

[9] R. Rizal and A. Rahmatulloh, "Restful Web Service Untuk Integrasi Sistem Akademik Dan Perpustakaan Universitas Perjuangan," *J. Ilm. Inform.*, vol. 7, no. 01, pp. 54–59, 2019.

[10] R. Hanifia, "Penerapan Quality of Service (Qos) Differentiated Service Pada Jaringan Multi-Protocol Label Switching (Mpls)," *J. Manaj. Inform.*, vol. 9, no. 2, pp. 1–7, 2019.

[11] D. L. Kurdi and B. S. Panca, "Pengujian Performa Komunikasi VoIP Menggunakan Static dan Dynamic Routing Protocol," *J. Strateg.*, vol. 2, no. 1, pp. 111–119, 2020.