

# Sistem Pemantauan Tempat Sampah menggunakan Pemodelan *Edge Computing*

SITI ROKOIYE, URAY RISTIAN, KASLIONO

Rekayasa Sistem Komputer, Universitas Tanjungpura, Indonesia  
Email: [siti\\_rokoiye@student.untan.ac.id](mailto:siti_rokoiye@student.untan.ac.id)

*Received* 17 Oktober 2023 | *Revised* 13 November 2023 | *Accepted* 29 November 2023

## ABSTRAK

*IoT adalah teknologi yang memanfaatkan konektivitas internet untuk menghubungkan perangkat dan bertukar data. Contoh penerapannya adalah pada sistem pemantauan tempat sampah yang melakukan pemantauan kapasitas sampah dari jarak jauh melalui koneksi internet. IoT memiliki tantangan seperti keterlambatan respons data dan ketergantungan pada internet. Oleh karena itu, untuk mengatasi masalah tersebut, dalam penelitian ini diterapkan pemodelan edge computing dengan memanfaatkan edge server. Penggunaan edge server pada sistem dapat memproses dan mengirimkan data dari berbagai tempat sampah. Ketergantungan pada koneksi internet saat melakukan pemrosesan data dari perangkat IoT dapat dikurangi karena edge server melakukan pemrosesan data secara lokal. Pemrosesan dan pengiriman data menggunakan protokol HTTP. Hasil pengujian QoS yang dilakukan dalam penerapan model edge computing pada sistem pemantauan tempat sampah diperoleh nilai rata-rata throughput 1338 bps dan rata-rata delay adalah 115 ms.*

**Kata kunci:** *Internet Of Things, Pemantauan Tempat Sampah, Edge Computing*

## ABSTRACT

*IoT is a technology that utilizes internet connectivity to connect devices and exchange data. An example of its application is in a trash bin monitoring system that remotely monitors trash capacity through an internet connection. IoT has challenges such as data response delays and dependence on the internet. Therefore, to overcome these problems, this research applies edge computing modeling by utilizing edge servers. The use of edge servers in the system can process and transmit data from various bins. Dependence on the internet connection when processing data from IoT devices can be reduced because the edge server performs data processing locally. Processing and sending data using the HTTP protocol. The results of QoS testing carried out in the application of the edge computing model in the bin monitoring system obtained an average throughput value of 1338 bps and the average delay is 115 ms.*

**Keywords:** *Internet of Things, Trash Bin, Edge Computing*

## 1. PENDAHULUAN

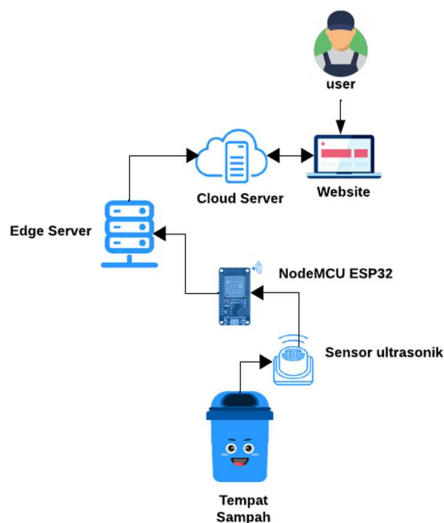
Dalam upaya menjaga kebersihan lingkungan dan mengurangi dampak buruk sampah, pemerintah telah menetapkan aturan yang mengharuskan pemilihan sampah pada tempatnya. UU No. 18 Tahun 2008 Pasal 29 Ayat (1) memberlakukan sanksi pidana bagi pelanggaran tersebut, baik berupa kurungan maupun denda. Meskipun pemerintah menunjukkan komitmen dengan menerapkan aturan tersebut, kenyataannya pengelolaan sampah secara manual oleh petugas kebersihan tanpa dukungan teknologi informasi yang memadai masih menyebabkan proses penanganan sampah menjadi lambat dan tidak efisien. Oleh karena itu, diperlukan solusi untuk mengatasi masalah ini. Salah satu solusi yang dapat digunakan adalah teknologi *Internet of Things* (IoT). Teknologi *Internet of Things* (IoT) memanfaatkan konektivitas internet yang selalu terhubung untuk mengendalikan dan menghubungkan suatu perangkat terhubung dengan perangkat lainnya. Salah satu pemanfaatan IoT yaitu pada sistem pemantauan tempat sampah, memudahkan pemantauan ketinggian sampah dari jarak jauh. Data ketinggian akan selalu *update* secara *real-time* melalui koneksi Wi-Fi dan internet **(Suryaningrat, dkk, 2021)**. Namun, sistem pemantauan yang berbasis IoT sebagian besar menggunakan sumber daya *cloud* untuk menjalankan monitoring dan komputasi sederhana **(Sahifa, dkk, 2020)**. *Edge computing* adalah model komputasi baru yang mengalokasikan sumber daya komputasi dan penyimpanan di tepi jaringan lebih dekat ke sensor atau sumber data **(Cao, dkk, 2020)**. *Edge computing* merupakan sistem komputasi yang dioptimalkan untuk memproses dan menyimpan aliran data dekat dengan perangkat IoT secara lokal, yang ditempatkan dekat dengan sumber data atau *node* **(Izquierdo, dkk, 2019)**. *Edge computing* melakukan pemrosesan data dan pengambilan keputusan dilakukan secara lokal di perangkat *edge* atau *edge server* **(Dalbina, 2019)**.

Berdasarkan latar belakang yang diuraikan masalah kendala dalam proses pengelolaan sampah secara manual mengakibatkan lambatnya penanganan sampah. Teknologi *Internet of Things* (IoT) menawarkan solusi pemantauan sampah secara *real-time* **(Khozin, dkk, 2022)**, tetapi menghadapi tantangan seperti keterlambatan respons data dan ketergantungan pada koneksi internet **(Murdiyantoro, dkk, 2021)**, untuk mengatasi hal tersebut maka dilakukan penelitian dengan judul "Sistem Pemantauan Tempat Sampah menggunakan Pemodelan *Edge Computing*". Dengan adanya sistem ini, data dapat terus diproses, dan disimpan di *edge* hingga kendala teratasi, dan hasilnya dapat disinkronkan kembali dengan *cloud server*. Sehingga sistem tetap dapat beroperasi di *edge server* bahkan saat terjadi gangguan di *cloud server*.

## 2. METODE

### 2.1 Deskripsi Sistem

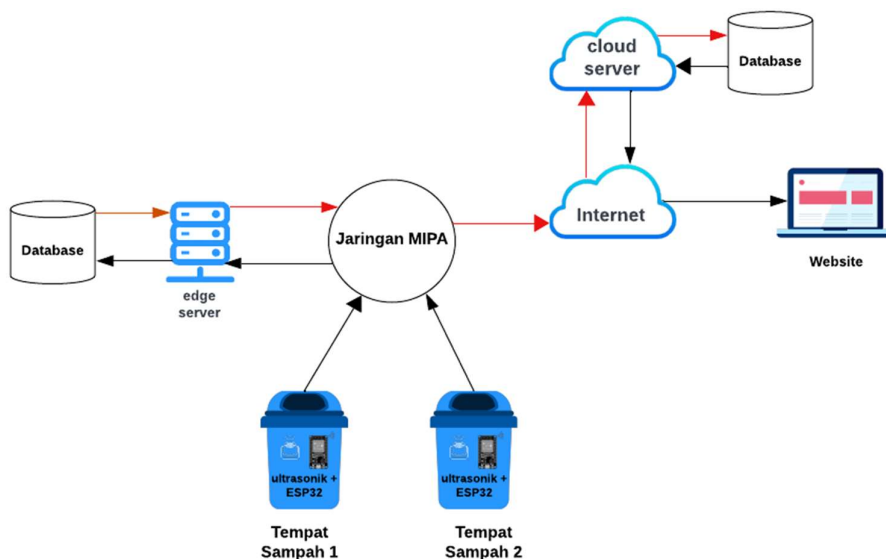
Pada penelitian ini sistem yang akan dibangun adalah sebuah sistem yang dapat memantau kapasitas sampah pada tempat sampah dengan menerapkan model *edge computing* yaitu pemrosesan awal atau pengumpulan data dilakukan di lokasi yang lebih dekat dengan sumber data **(Dalbina, 2019)**. Sistem ini memanfaatkan perangkat keras NodeMCU ESP32 dan sensor ultrasonik untuk mengukur kapasitas tempat sampah dengan implementasi model *edge computing* yang berarti pengolahan data akan dilakukan terlebih dahulu dari tempat sampah ke *edge server* sebelum data dikirim ke server pusat (*cloud server*). Sistem terhubung *cloud server* melalui jaringan internet, Informasi mengenai kapasitas sampah dapat diakses dan dikelola dari jarak jauh, dengan demikian pengguna dapat memantau kapasitas sampah pada setiap tempat sampah secara *real-time* **(Suryaningrat, dkk, 2021)**. Gambar 1 merupakan gambaran sistem secara umum.



**Gambar 1. Sistem Secara Umum**

**2.2 Model Arsitektur**

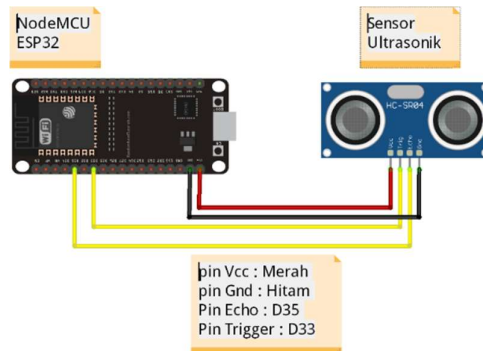
Gambar 2 merupakan model arsitektur jaringan. Proses dimulai dengan *node* yang bertugas mengumpulkan data dari tempat sampah. Data ini mencakup informasi tentang kapasitas sampah. Selanjutnya, data dari *node* akan dikirimkan ke *edge server* menggunakan HTTP *Request* dengan bantuan API (Ehsan, dkk, 2022). *Edge server* menerima data dan menyimpannya di dalam *database*. Setelah tersimpan, data akan dikirimkan ke *cloud server* melalui HTTP *Request*. Jika koneksi internet stabil atau tidak ada gangguan, *edge server* akan melakukan HTTP *Request* untuk menyimpan data di dalam *database cloud server*. Data tersebut kemudian dapat diakses melalui *website*. Pada sistem ini memanfaatkan model *edge computing* untuk memproses data secara lokal dengan memanfaatkan *edge server* sebagai pusat pengelolaan dan pengiriman data dari berbagai tempat sampah. Penggunaan *edge server* pada sistem dapat memproses data dan mengurangi ketergantungan pada koneksi internet saat melakukan pemrosesan data dari perangkat IoT (Cao, dkk, 2020).



**Gambar 2. Arsitektur Jaringan**

### 2.3 Perancangan Perangkat Keras

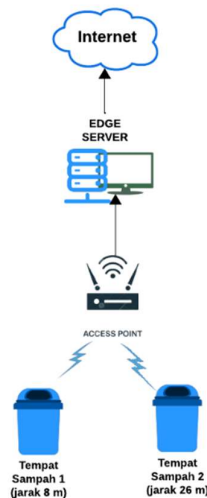
Gambar 3 merupakan sistem perangkat keras yang dirancang untuk mengukur dan mengetahui kapasitas sampah pada tempat sampah. Adapun komponen yang digunakan untuk membangun sistem adalah NodeMCU ESP32 dan sensor ultrasonik. Pada penelitian ini NodeMCU ESP32 bertugas sebagai Mikrokontroler (**Hadyanto & Muhammad, 2022**) yang berfungsi untuk mengontrol operasi dari sensor ultrasonik. Data jarak yang diterima oleh sensor ultrasonik digunakan oleh ESP32 untuk menghitung kapasitas sampah pada tempat sampah tersebut. ESP32 juga bertanggung jawab untuk mengirimkan data hasil pengukuran tersebut ke *edge server* melalui koneksi jaringan. Perancangan sistem pemantauan tempat sampah (**Suryaningrat, dkk, 2021**).



**Gambar 3. Sistem Pemantauan Tempat Sampah**

### 2.4 Perancangan Model Sistem *Edge Computing*

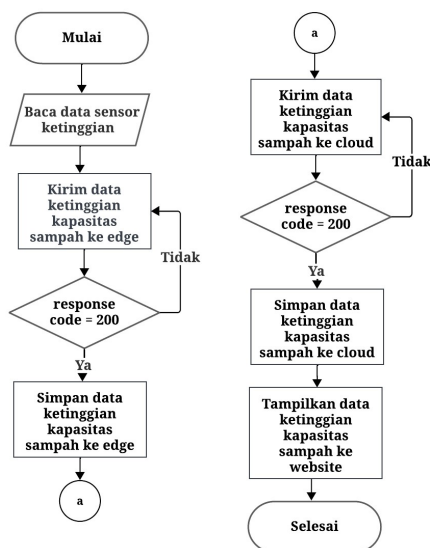
Pada penelitian ini, dirancang sebuah sistem yang memanfaatkan konsep *edge computing* yaitu pengolahan data dilakukan di dekat sumber data (**Waranugraha & Suryanegara, 2020**). Model ini didukung oleh sebuah perangkat keras yang disebut sebagai *edge server* (**Cao, dkk, 2020**). *Edge server* ini difungsikan untuk memproses data yang diterima dari kedua tempat sampah tersebut. Tempat sampah 1 diletakkan dengan jarak 8 meter dari *edge server* dan tempat sampah 2 dengan jarak 26 meter dari *edge server*. Perancangan model sistem *edge computing* dapat dilihat pada Gambar 4.



**Gambar 4. Perancangan Model Sistem *Edge Computing***

## 2.5 Perancangan Perangkat Lunak

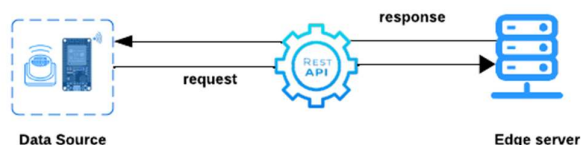
Perancangan perangkat lunak pada implementasi pemodelan *edge computing* pada sistem pemantauan tempat sampah melibatkan pengiriman data dari sensor ke *edge server*. Data yang diterima akan disimpan terlebih dahulu di *edge server* sebelum diteruskan ke *cloud server*. Konsepnya adalah memanfaatkan *edge server* sebagai titik pertama penerimaan data, yang kemudian akan memproses, menyimpan, dan mengirimkannya ke *cloud server* jika kondisi jaringan dan sistem berjalan normal. Jika ada kendala dalam pengiriman data ke *cloud server*, data akan tetap disimpan di *edge server* dengan status 'pending' dan akan diusahakan untuk dikirim ulang ke saat masalah tersebut teratasi. Data yang telah terkumpul di *cloud server* akan ditampilkan dalam bentuk antarmuka *website*. Diagram alir perangkat lunak dapat dilihat pada Gambar 5.



Gambar 5. Diagram Alir Perangkat Lunak

## 2.6 Perancangan Desain Komunikasi Antara *Node* dan *Edge Server*

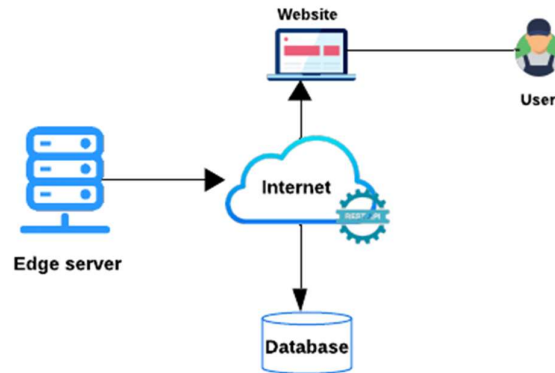
Sistem perancangan desain komunikasi antara *node* dan *edge server* pada sistem pemantauan tempat sampah menggunakan protokol komunikasi HTTP (**Abdurahman, dkk, 2020**). *Node* akan mengirimkan permintaan HTTP (*HTTP request*) kepada *edge server* dengan menyertakan informasi data kapasitas sampah seperti *id\_sensor*, *kapasitas*, dan *status*. Selanjutnya, *edge server* akan menerima permintaan HTTP dari *node* dan mengolah data yang diterima. Data kapasitas sampah akan disimpan dalam basis data di *edge server*. Proses desain komunikasi antara *node* dan *edge server* menggunakan HTTP melalui API sebagai penghubung dalam memastikan data kapasitas sampah diterima dengan tepat (**Ehsan, dkk, 2022**). Sistem ini juga mampu mengatasi masalah koneksi atau kesalahan di *computing* atau *cloud server* pusat dengan menyimpan data di *edge* dengan status pending. Data akan tetap tersimpan di *edge server* dan akan dikirim ke *cloud server* pusat ketika koneksi antara *edge server* dan *cloud server* pusat kembali normal dan data yang sebelumnya berstatus *pending* akan berubah statusnya menjadi sukses.



Gambar 6. Perancangan Desain Komunikasi antara *Node* dan *Edge Server*

## 2.7 Perancangan Desain Komunikasi Antara *Edge Server* dan *Cloud Server*

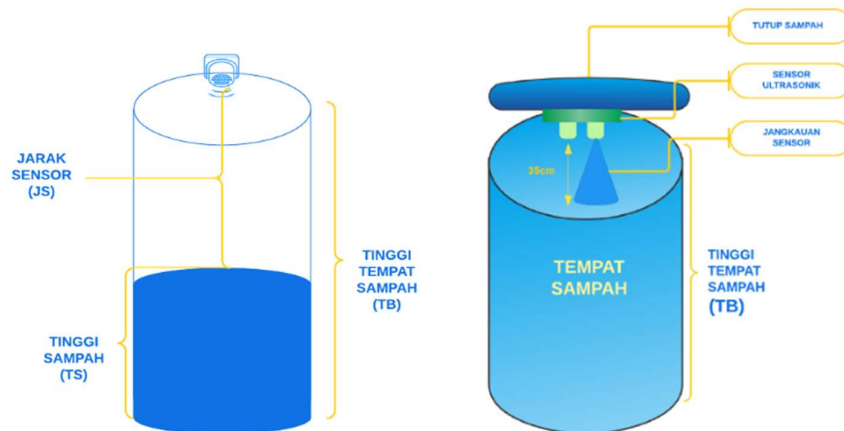
Gambar 7 merupakan perancangan desain komunikasi antara *edge server* dan *cloud server* menggunakan protokol HTTP. Jika tidak ada gangguan koneksi, data akan langsung dikirim ke *cloud server*. Namun, jika ada gangguan koneksi, data akan tetap tersimpan di *edge server* dalam status *pending*. Setelah kondisi jaringan normal kembali, data tertunda akan dikirimkan ke *cloud server* dan disimpan dalam *database*. Pengguna dapat melihat data melalui aplikasi *website* untuk memantau kondisi tempat sampah secara *real-time*.



**Gambar 7. Perancangan Desain Komunikasi antara *Edge Server* dan *Cloud Server***

## 2.8 Perancangan Perhitungan Kapasitas Sampah

Gambar 8 merupakan gambar perhitungan kapasitas sampah. Menggunakan sensor ultrasonik untuk mengukur tinggi tumpukan sampah dan kemudian menghitung persentase kapasitas sampah.



**Gambar 8. Perancangan Perhitungan Kapasitas Sampah**

Perhitungan persentase kapasitas sampah dalam tempat sampah diperoleh dari perhitungan dengan menggunakan Persamaan 1 dan Persamaan 2.

$$TS = TB - \text{jaraksensor} \quad (1)$$

$$\text{CAPACITY} = (TS) / TB \times 100 \quad (2)$$

Keterangan:

TS = Tinggi Sampah.

TB = Tinggi Bak (tinggi tempat sampah).

## 2.9 Perancangan Pengujian QoS Performa *Edge Computing*

Pengujian QoS dilakukan untuk mengetahui nilai *delay* dan *throughput* data dengan penerapan *edge computing* pada sistem pemantauan tempat sampah. Pada pengujian QoS dilakukan terhadap 2 parameter, diantaranya *throughput* dan *delay* (ETSI, 1999) (Rasudin, 2014). Pengujian dilakukan dengan menggunakan aplikasi wireshark digunakan untuk mengetahui setiap paket data yang dikirim antara sensor, *edge server* dan *cloud server* (Nugraha, dkk, 2022).

1. Skenario Pengujian ke – 1  
Pengujian dilakukan dengan meletakkan tempat sampah 1 dengan jarak 8 meter dari *edge server*.
2. Skenario Pengujian ke – 2  
Pengujian ini dilakukan dengan meletakkan tempat sampah 2 dengan jarak 26 meter dari *edge server*.
3. Skenario Pengujian ke – 3  
Pengujian ini dilakukan dengan mengirimkan data dari *edge server* ke *cloud server*.

## 3. HASIL DAN PEMBAHASAN

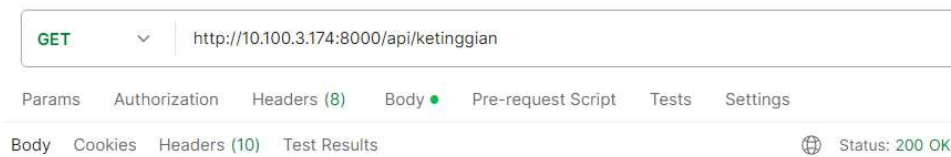
### 3.1 Hasil Implementasi

Pada implementasi perangkat keras komponen utama dalam sistem pemantauan kapasitas sampah adalah NodeMCU ESP32 dan sensor ultrasonik. Sensor ultrasonik yang terhubung dengan NodeMCU ESP32 berperan untuk mengukur kapasitas sampah di dalam tempat sampah dengan meletakkan sensor di atas tutup tempat sampah. Gambar 9 merupakan gambar sistem pemantauan tempat sampah (Suryaningrat, dkk, 2021).



**Gambar 9. Sistem Pemantauan Tempat Sampah**

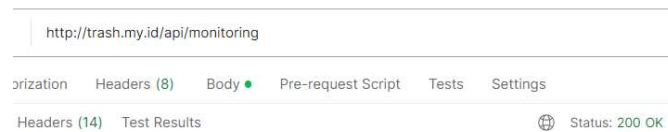
Dalam implementasi perangkat lunak, dilakukan pengembangan tampilan antarmuka berbasis *website*. Proses ini juga melibatkan sistem komunikasi antara *node*, *edge server* dan *cloud server*. Dalam implementasi sistem komunikasi antara *node*, *edge server* dan *cloud server*, dilakukan pengujian dengan melakukan *request API* pada *endpoint* yang telah disediakan. Pengujian kelayakan API yang telah dibuat. Untuk memverifikasi bahwa antarmuka pemrograman aplikasi (API) yang menghubungkan antara sensor ke *edge server* dan *cloud server* beroperasi sesuai yang diharapkan. Proses pengujian ini melibatkan pengiriman ataupun permintaan ke API yang telah dibangun. Adapun cara menguji API dengan menggunakan Postman.



**Gambar 10. Hasil Pengiriman Data ke Edge Server**

Gambar 10 merupakan hasil API menunjukkan respons kode 200, yang menandakan bahwa API berhasil menjalankan tindakan sesuai permintaan.

Implementasi sistem komunikasi antara *edge server* dan *cloud server* merupakan mekanisme yang digunakan untuk memastikan terhubungnya *edge server* dengan *database* di *cloud server*. Membuat API yang berfungsi untuk merespons permintaan yang berasal dari *edge server* yang mengirimkan data ke dalam *database cloud server*. Dilakukan pengujian dengan melakukan *request* API pada *endpoint* yang telah disediakan. Pengujian kelayakan API yang telah dibuat. Untuk memverifikasi bahwa *edge server* dan *cloud server* beroperasi sesuai yang diharapkan. Adapun langkah-langkah penerapannya melibatkan tahap validasi data masukan sebelum informasi tersebut dimasukkan ke dalam basis data. Proses pengujian ini melibatkan pengiriman ataupun permintaan ke API yang telah dibangun. Adapun cara menguji API dengan menggunakan Postman.



**Gambar 11. Hasil Pengiriman Data ke Cloud Server**

Gambar 11 API menunjukkan respons kode 200, yang menandakan bahwa API berhasil menjalankan tindakan sesuai permintaan.

### 3.2 Hasil Pengujian

Tabel 1 merupakan hasil pengujian pendeteksian kapasitas pada tempat sampah 1 secara manual dan di *website* bertujuan untuk memastikan keakuratan dan keselarasan data antara kondisi tempat sampah secara langsung dengan informasi yang ditampilkan di *website*. Proses ini penting untuk memastikan bahwa data yang diberikan kepada pengguna melalui *website* mencerminkan kondisi aktual dari tempat sampah yang sedang dipantau. Pada tiap pengujian dilakukan sebanyak lima kali pengujian. Pengujian kapasitas tempat sampah 1 memperoleh nilai *error* rata-rata 3,2 % (Billin, dkk, 2021). Dari data Tabel 1, dapat diambil kesimpulan bahwa tingkat akurasi sensor 1 sebesar 96,8 %.

**Tabel 1. Pengujian Sistem Pembacaan Kapasitas Sampah 1**

Pengujian ke -	Tempat Sampah 1				Selisih
	Manual		Website		
	TS (cm)	Kapasitas (%)	TS (cm)	Kapasitas (%)	
1	6	16	5	13	3
2	8	21	9	24	3
3	13	35	14	38	3
4	15	40	16	43	3
5	17	45	18	48	3
6	20	54	21	57	3
7	24	65	25	68	3



8	27	73	28	76	3
9	31	84	33	89	5
10	33	89	34	92	3
Rata-rata (%)					3,2

Tabel 2 merupakan hasil pengujian pendeteksian kapasitas pada tempat sampah 2 secara manual dan di *website*. Dari data Tabel 2, dapat diambil kesimpulan bahwa tingkat akurasi sensor 2 sebesar 97,9 %.

**Tabel 2. Pengujian Sistem Pembacaan Kapasitas Sampah 2**

Pengujian ke-	Tempat Sampah 2				Selisih
	Manual		<i>Website</i>		
	TS (cm)	Kapasitas (%)	TS (cm)	Kapasitas (%)	
1	0	0	0	0	0
2	9	25	10	27	2
3	11	30	12	33	3
4	18	50	19	52	2
5	29	81	31	84	3
6	17	47	17	47	0
7	21	58	22	61	3
8	25	69	26	72	3
9	22	61	23	63	2
10	26	72	27	75	3
Rata-rata (%)					2,1

Gambar 12 merupakan hasil pengujian pengiriman data dari tempat sampah ke *edge server* ketika ada kendala atau gangguan di *cloud server* data akan tetap tersimpan di *edge server* dengan status *pending* dan akan di kirim ulang ketika gangguan di *cloud server* sudah teratasi kemudian data yang sudah terkirim di *edge server* akan berubah statusnya menjadi *success*.

```

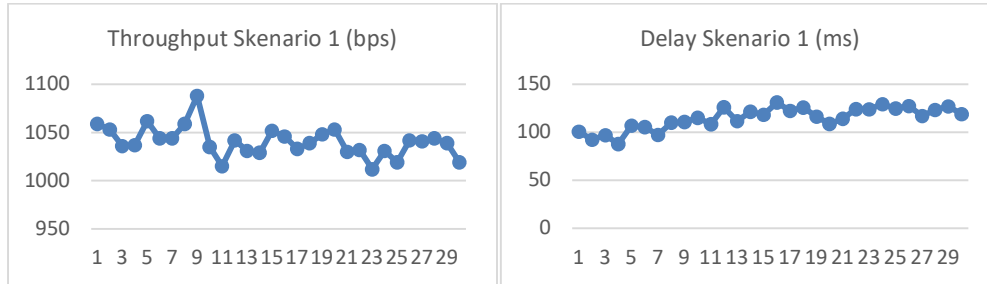
Monitor x
o send message to 'DOIT ESP32 DEVKIT V1' on 'COM3'

-> IP Sensor: 10.100.3.26
-> Levelling...
-> #Tinggi Sampah (CM) = 16
-> #KAPASITAS % = 45
-> [HTTP] Memulai...
-> Data : id_sensor=2&kapasitas=45
-> [HTTP] Melakukan POST ke server...
-> HTTPCODE : 200[HTTP] Kode respons POST: 200
-> {"code":200,"message":"Upload Edge Success, but cloud server has trouble","data":{"id_sensor":1,"kapasitas":45,"status":"pending"},
-> IP Sensor: 10.100.3.26
-> Levelling...
    
```

**Gambar 12. Pengiriman Data Ketika *Cloud* Terkendala**

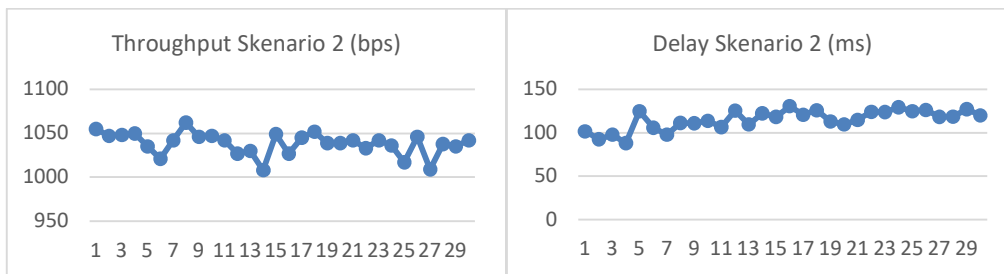
Gambar 13 merupakan hasil *throughput* dan *delay* dengan skenario 1. *Throughput* yaitu kecepatan (*rate*) transfer data efektif, yang diukur dalam bps (*bit per second*). *Throughput* merupakan jumlah total paket yang sukses yang diamati selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut (**ETSI, 1999**). *Delay* atau *latency* adalah waktu yang

dibutuhkan data untuk menempuh jarak dari asal ke tujuan (**ETSI, 1999**). *Delay* dapat dipengaruhi oleh jarak, media fisik, atau waktu proses yang lama. Hasil pengujian *throughput* yang dilakukan diperoleh nilai rata-rata *throughput* = 1040 bps dan hasil pengujian *delay* yang dilakukan diperoleh nilai rata-rata *delay* = 114 ms.



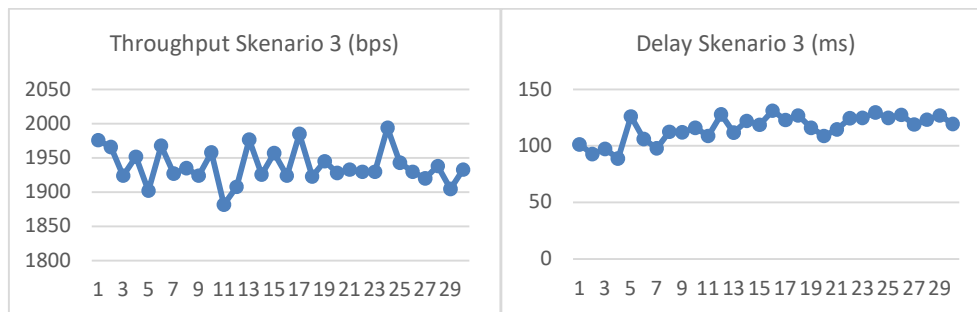
**Gambar 13. *Throughput* dan *Delay* Skenario 1**

Gambar 14 merupakan hasil *throughput* dan *delay* dengan skenario 2. Hasil pengujian *throughput* yang dilakukan diperoleh nilai rata-rata *throughput* = 1038 bps. dan hasil pengujian *delay* yang dilakukan diperoleh nilai rata-rata *delay* = 115 ms.



**Gambar 14. *Throughput* dan *Delay* Skenario 2**

Gambar 15 Merupakan hasil *throughput* dan *delay* dengan Skenario 3. Hasil pengujian *throughput* yang dilakukan diperoleh nilai rata-rata *throughput* = 1938 bps. Hasil pengujian *delay* yang dilakukan diperoleh nilai rata-rata *delay* = 116 ms.



**Gambar 15. *Throughput* dan *Delay* Skenario 3**

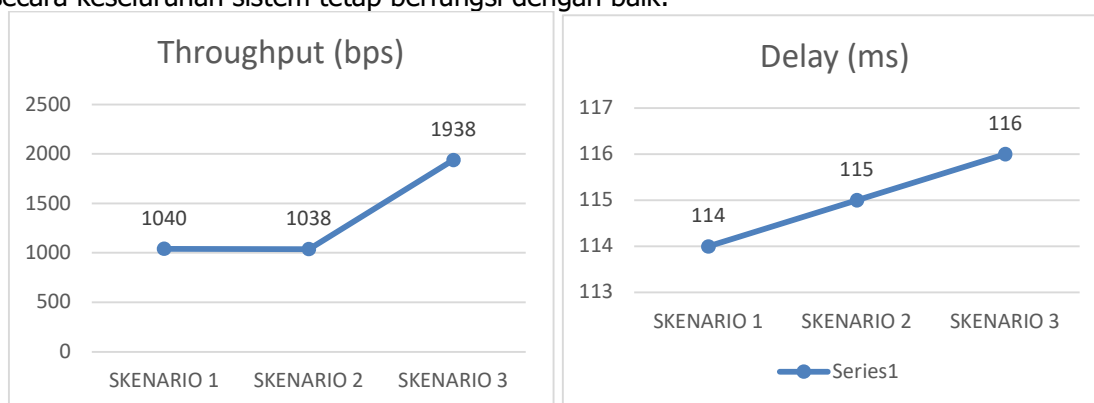
Gambar 16 menunjukkan perbandingan *throughput* dan *delay* antara pengujian skenario 1, skenario 2 dan skenario 3. Hasil pengujian kualitas layanan (QoS) pada model sistem *edge computing* menunjukkan bahwa sistem mampu mengirim data dari sensor ke *edge server* dengan baik pada jarak yang berbeda. Pada Tempat Sampah 1 yang berjarak 8 meter dari *edge server* diperoleh nilai *throughput* rata-rata sebesar 1040 bps dan nilai rata-rata *delay* 114

ms. Sementara itu, pada Tempat Sampah 2 yang berjarak 26 meter dari *edge server* diperoleh nilai *throughput* rata-rata sebesar 1038 bps dan nilai rata-rata *delay* 115 ms.

Perbedaan nilai *throughput* dan *delay* antara kedua jarak tersebut dapat dijelaskan oleh beberapa faktor. Salah satunya adalah jarak antara tempat sampah dan *edge server*. Tempat sampah 1 memiliki keunggulan dalam hal jarak yang lebih dekat dengan infrastruktur jaringan, sehingga *throughput* sedikit lebih tinggi dan *delay* lebih rendah dibandingkan dengan Tempat sampah 2 yang berjarak lebih jauh.

Pengiriman data dari *edge server* ke *cloud server*, terjadi peningkatan *throughput* rata-rata menjadi 1938 bps, sedangkan peningkatan *delay* rata-rata adalah 116 ms. Peningkatan *throughput* dan *delay* yang terjadi karena jumlah *bytes* dan total paket yang diterima di *edge server* dari 2 tempat sampah yang kemudian di kirim ke *cloud server*, pengiriman data dari *edge server* ke *cloud server* juga dapat dipengaruhi oleh faktor-faktor seperti keterbatasan di sisi *cloud server* dan kondisi jaringan, yang juga mempengaruhi *throughput* dan *delay* dalam pengiriman data ke *cloud server*.

Data hasil pengujian tersebut memberikan gambaran konsistensi performa sistem dalam mengirim data. Meskipun terdapat sedikit variasi dalam nilai *throughput* dan *delay*, namun secara keseluruhan sistem tetap berfungsi dengan baik.



**Gambar 16. Perbandingan *Throughput* dan *Delay***

Penggunaan *edge server* berfungsi sebagai pusat pemrosesan data yang terdekat dengan sumber data, yaitu tempat sampah. *Edge server* merupakan bagian penting dalam sistem pemantauan tempat sampah, beroperasi secara lokal untuk memproses dan mengirimkan data dari 2 tempat sampah. Hal ini mengoptimalkan pengiriman data ke *cloud server*. *Edge server* melakukan pemrosesan awal data di tingkat lokal sebelum mengirimkannya ke *cloud server*, mengurangi ketergantungan pada jaringan internet yang mungkin tidak selalu stabil. Dengan demikian, pengelolaan dan pengiriman data menjadi lebih efisien.

#### 4. KESIMPULAN

Pada penelitian ini telah berhasil dibangun sistem pemantauan tempat sampah dengan menerapkan model *edge computing* yaitu pemrosesan awal dilakukan dekat dengan sumber data memanfaatkan *edge server* yang berada secara fisik lebih dekat dengan tempat sampah. *Edge server* beroperasi secara lokal, memproses, dan mengirimkan data dari berbagai tempat sampah. Penggunaan *edge server* pada sistem dapat memproses dan mengirimkan data dari berbagai tempat sampah. Ketergantungan pada koneksi internet saat melakukan pemrosesan

data dari perangkat IoT dapat dikurangi karena *edge server* melakukan pemrosesan data secara lokal. Hasil dari pengujian QoS dengan implementasi pemodelan *edge computing* pada sistem pemantauan tempat sampah dengan jarak 8 meter dan 26 meter dari *edge server* memperoleh nilai *throughput* rata-rata 1338 bps dan nilai *delay* rata-rata 115 ms dengan kategori sangat baik. Mengoptimalkan dalam proses dan pengiriman data ke *cloud server*.

## DAFTAR RUJUKAN

- Abdurahman, A., Rendy Munadi, I. M., & Indra Irawan, A. S. (2020). Perancangan Dan Implementasi Hidroponik Berbasis Internet Of Things (IOT) Menggunakan Protokol HTTP. *E-Proceeding Of Engineering*, 7, 3862–3868.
- Billin, V., Triyanto, D., & Ristian, U. (2021). Sistem Pemantauan Tempat Penampungan Sampah Secara Realtime Dengan Memanfaatkan Location Tracking Menggunakan Antarmuka Website. *Jurnal Komputer Dan Aplikasi*, 09, 364–374.
- Cao, K., Liu, Y., Meng, G., & Sun, Q. (2020). An Overview On *Edge Computing* Research. In *IEEE Access* (Vol. 8, Pp. 85714–85728). Institute Of Electrical And Electronics Engineers Inc. <https://doi.org/10.1109/ACCESS.2020.2991734>
- Dalbina, D. (2019). *An Overview Of Edge Computing*. [www.ijert.org](http://www.ijert.org)
- Ehsan, A., Abuhaliqa, M. A. M. E., Catal, C., & Mishra, D. (2022). Restful API Testing Methodologies: Rationale, Challenges, And Solution Directions. In *Applied Sciences (Switzerland)* (Vol. 12, Issue 9). MDPI. <https://doi.org/10.3390/app12094369>
- ETSI. (1999). *Telecommunications And Internet Protocol Harmonization Over Networks (TIPHON); General Aspects Of Quality Of Service (Qos). 1*. <http://www.etsi.org>
- Izquierdo, Z., Miguel A., Santa, J., Martínez, J. A., Martínez, V., & Skarmeta, A. F. (2019). Smart Farming Iot Platform Based On *Edge And Cloud* Computing. *Biosystems Engineering*, 177, 4–17. <https://doi.org/10.1016/j.biosystemseng.2018.10.014>
- Khozin, A., Winardi, S., Arifin, M. N., & Nugroho, A. (2022). Tempat Sampah Pintar Berbasis Internet Of Things Pada SMKN 1 Dlanggu Kabupaten Mojokerto. In *Jurnal Pendidikan Teknologi Informasi (JUKANTI)* (Issue 5).
- Murdiyantoro, R. A., Izzinnahadi, A., & Armin, E. U. (2021). Sistem Pemantauan Kondisi Air Hidroponik Berbasis Internet Of Things Menggunakan Nodemcu ESP8266. *Journal Of Telecommunication, Electronics, And Control Engineering (JTECE)*, 3(2), 54–61. <https://doi.org/10.20895/jtece.v3i2.258>
- Nugraha, G., Made, I. A., & Putu, I. A. (2022). Analisis Tren Lalu-Lintas Data Jaringan Menggunakan Teknologi Big Data (Studi Kasus: Universitas Mahadewa Indonesia). *Jurnal Ilmiah Teknologi Dan Komputer*, 3(2).

- Rasudin. (2014). Quality Of Service (QOS) Pada Jaringan Internet Dengan Metode Hierarchy Token Bucket. *TECHSI*, 4, 210–223.
- Sahifa, A., Setiawan, R., & Yazid, M. (2020). Pengiriman Data Berbasis Internet Of Things Untuk Monitoring Sistem Hemodialisis Secara Jarak Jauh. *Jurnal Teknik Its*, 2. <https://ejournal.its.ac.id>
- Suryaningrat, A., Kurnianto, D., & Syifa, F. T. (2021). Pemanfaatan Google Firebase Pada Sistem Tempat Sampah Pintar Berbasis Internet Of Things. *Dinamika Rekayasa*, 17, 1–9. <http://dinarek.unsoed.ac.id>
- Hadyanto, T., & Muhammad, F. A. (2022). Sistem Monitoring Suhu Dan Kelembaban Pada Kandang Anak Ayam Broiler Berbasis Internet Of Things. *JTST*, Vol. 03, No.02.
- Waranugraha, N., & Suryanegara, M. (2020). The Development Of Iot-Smart Basket : Performance Comparison Between *Edge Computing* And *Cloud Computing* System. *International Conference On Computer And Informatics Engineering (IC2IE)*, 3, 410–414.