

ANALISIS KOMPARASI *FOG COMPUTING* - *CLOUD COMPUTING* DALAM IMPLEMENTASI PENGOLAHAN DATA CUACA BERBASIS IOT

COMPARISON ANALYSIS OF FOG COMPUTING AND CLOUD COMPUTING IN CONCERN DATA PROCESSING OF WEATHER CLIMATE BASED ON IOT

Asep Mulyana¹, Tita Haryanti², Risyaf Fawwaz Pradipta³

^{1,2,3}D3 Teknologi Telekomunikasi, Fakultas Ilmu Terapan, Universitas Telkom

¹asepmulyana@telkomuniversity.ac.id, ²tharyanti@telkomuniversity.ac.id,

³risvaffp@student.telkomuniversity.ac.id

Abstrak

Cloud computing merupakan penyediaan sumber daya komputasi yang diakses secara *remote* oleh pengguna. Untuk aplikasi umum, cara ini dinilai sangat efisien dibanding cara konvensional (*on premises*). Namun untuk aplikasi IoT dengan sumber trafik berupa *sensor node* dengan jumlah yang sangat besar tersebar di seluruh dunia bukanlah model *remote user* akan membangkitkan trafik individu dengan laju kedatangan yang sangat tinggi berpotensi terjadinya *overload* pada *cloud*. Untuk mengatasi hal ini para ahli mengembangkan model komputasi antara sumber trafik dengan *cloud* yang dikenal sebagai *fog computing*, bertujuan untuk “meringankan” beban *cloud*. Sampai sejauh pengaruh “meringankan” beban pada *cloud* tersebut; dalam penelitian ini dilakukan eksperimen membangun sistem sensor cuaca berbasis IoT dengan elemen sensor terdiri dari sensor temperatur, kelembaban, tekanan udara, dan curah hujan. Eksperimen dilakukan dalam dua skenario yaitu tanpa *fog* dan dengan *fog*. Sebagai komponen *fog* digunakan *rasberry pi*. Dari hasil pengukuran *delay* rata-rata untuk skenario tanpa *fog* 117.4 mili detik dan dengan *fog* 58.8 mili detik. Ini berarti dengan *fog* diperoleh kinerja dua kali lebih baik (lebih cepat) dibanding tanpa *fog*. Sebagai verifikasi dilakukan pula analisis trafik (teoritik) menggunakan model antrian M/M/1. Diperoleh hasil perbaikan kinerja *delay* pada skenario dengan *fog* dua kali lebih cepat dibanding tanpa *fog* (pada beban 40-60%).

Kata kunci: peramalan cuaca, sensor node cuaca, IoT, *cloud computing*, *fog computing*.

Abstract

Cloud computing is the provision of computing resources that are accessed remotely by users. For general applications, this method is considered very efficient compared to conventional methods (on premises). However, for IoT applications in which very large number of sensor nodes are considered as individual traffic with a very high arrival rate, potentially overloading the cloud. To overcome this, the experts developed a computing model between traffic sources and the cloud known as fog computing, aiming to "lighten" the cloud load. To investigate how far the benefit of fog computing, in this study, experiments were conducted to build an IoT-based weather sensor system with sensor elements consisting of sensors of temperature, humidity, air pressure, and rainfall. The experiment was carried out in two scenarios, without fog and with fog. As a fog component used raspberry pi. From the measurement results, the average delay for scenarios without fog is 117.4 milliseconds and with fog 58.8 milliseconds. This means that with fog twice better (faster) than without fog. As a verification, traffic analysis (theoretical) was also carried out using the M/M/1 queuing model. The results of the delay performance with fog are twice faster than without fog (at 40-60% load).

Keywords: demand forecast, sensor node, IoT, cloud computing, fog computing

1. PENDAHULUAN

Internet of Things (IoT) adalah salah satu pilar teknologi yang mendukung industri 4.0. Banyak sekali bentuk implementasinya antara lain yang paling umum adalah *smart city* yang di dalamnya terdapat *smart transportation*, *smart health*, *smart building*, *weather forecast* [1] dan lain-lain yang dalam pemanfaatannya pada umumnya berbentuk otomasi atau sistem pintar dengan data masukan dari perangkat-perangkat sensor yang tekoneksikan satu sama lain melalui internet sebagai konektivitas antar perangkat IoT [2]. Di sisi lain tren perkembangan internet saat ini mengarah ke *cloud computing* dimana penyediaan sumber daya komputasi tidak lagi harus disediakan sendiri melainkan oleh penyedia layanan yang dapat diakses secara *remote* dengan skema pembayaran secara berlangganan (*subscription*) atau berdasarkan pemakaian (*pay as you go*). Sehingga cara ini dari segi ekonomi banyak menguntungkan [3] sehingga saat ini telah banyak perusahaan/instansi yang telah beralih dari model *on premises* ke *cloud computing*. Untuk aplikasi internet secara umum *cloud computing* memang menguntungkan bagi kedua pihak (*provider* dan *customer*)

Tidak demikian halnya bagi IoT dimana IoT tidak dapat dipandang atau dimodelkan sebagai *client* yang mengakses sumber daya dari jauh (secara *remote*), namun dalam IoT setiap *device* sebagai sumber trafik secara individu berupa perangkat sensor dengan jumlah yang sangat besar. Menurut *International Data Corporation*, sekitar 41,6 miliar perangkat IoT akan terhubung ke Internet, dan 79,4 zettabytes data akan menjadi total data yang dihasilkan pada tahun 2025 [4]. Kondisi ini berpotensi menyebabkan terjadinya beban lebih (*over load*) pada mesin *cloud computing*. Berdasarkan hal ini maka para ahli mengembangkan model komputasi tambahan yang diletakkan antara sumber trafik IoT dengan *Cloud computing* yang kemudian dikenal sebagai *fog computing* yang bertindak sebagai komputasi pembantu (*pre processing*) untuk “meringankan” beban *cloud computing*. Dalam penelitian ini yang menjadi konsen adalah sampai seberapa besar atau signifikan efek “meringankan beban” *fog computing* tersebut terhadap beban *cloud computing*.

Dalam penelitian sebelumnya Maulana, Rizky [5] dkk dalam tesisnya berjudul “*Analisis Kenyamanan Cuaca Berdasarkan Data Cloud Menggunakan Metode Naive Bayes Classifier*” melakukan analisis akurasi prediksi cuaca secara teoritik analitik menggunakan metoda *Naive Bayes Classifier* dengan mengambil data hasil pengukuran dari tiga lokasi yaitu: Bireuen, Banda Aceh, Langsa dan Gayo dengan hasil akurasi 89,47%. Dalam hal ini penelitian tersebut tidak melakukan realisasi pengukuran menggunakan perangkat keras dan aplikasi.

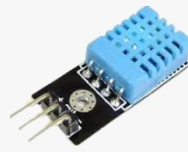
Berikutnya Auliati Nisa [6], dalam penelitiannya berjudul “*Pemanfaatan Teknologi IoT Untuk Monitoring Konsentrasi CO Dan CO₂ Dalam Upaya Mendeteksi Kebakaran Hutan*” melakukan monitoring data dengan membuat perangkat sensor untuk mengukur konsentrasi CO₂ sebagai prediksi terjadi kebakaran lahan/hutan. Namun disini tidak menggunakan *cloud computing* melainkan teknologi LoRa melalui *Gateway* ke pusat server pengolah data.

Dalam penelitian ketiga Nurul Hidayah Zulni [7] dkk dalam penelitiannya berjudul: “*Perancangan Aplikasi Monitoring Dan Prediksi Cuaca Pada Jemuran Otomatis*” melakukan pengukuran tiga parameter yaitu suhu, kelembaban, dan cahaya berbasis IoT untuk memprediksi terjadinya hujan. Namun disini tidak menggunakan teknologi *cloud computing* maupun *fog computing*.

2. DASAR TEORI

2.1. Sensor suhu dan kelembaban

Sensor DHT22 [8] adalah sebuah sensor yang dapat melakukan pengukuran suhu (*temperature*) ruangan dan kelembaban udara (*humidity*), yang didalamnya terdapat sebuah resistor dengan jenis *Negative Temperature Coefficient* (NTC).



Gambar 1 Sensor temperatur dan kelembaban DHT22 [8].

2.2. Sensor tekanan udara

Sensor BMP180 [9] adalah sebuah sensor barometrik (*digital barometric pressure sensor*) yang digunakan pada perangkat untuk mengukur tekanan udara atmosfer serta suhu udara sekitar. Tekanan ini sangat bervariasi sesuai keadaan cuaca dan ketinggian lokasi.



Gambar 2 Sensor tekanan udara (Barometer) [9].

2.3. Sensor curah hujan

Sensor curah hujan (*raindrop sensor*) [10] adalah sebuah sensor yang digunakan untuk mengukur curah hujan. Sensor ini terdiri dari dua modul, *rain board* yang digunakan untuk mendeteksi tetesan air dan modul kontrol yang membandingkan nilai analog dan mengubahnya menjadi nilai digital. Sensor ini biasanya digunakan pada sektor otomotif untuk mengontrol wiper kaca mobil secara otomatis saat terjadi hujan [10].



Gambar 3 Sensor curah hujan [10].

2.4. ESP 32

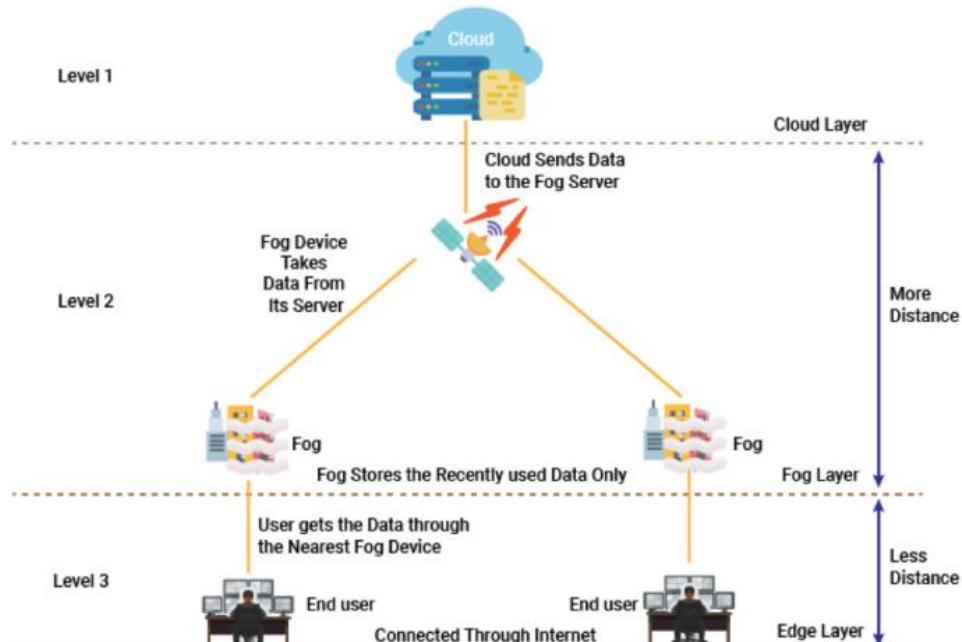
ESP32 [11] merupakan *module development board* berbasis WiFi dan Bluetooth yang dikembangkan sebagai penerus ESP8266 yang dapat diprogram menggunakan *software IDE* Arduino seperti halnya dengan NodeMCU. Salah satu kelebihan dari ESP32 ini dibandingkan dengan *module development board* berbasis ESP8266 lainnya yaitu adanya *module shield* untuk mendukung *hardware plug and play*.



Gambar 4 ESP 32 [11].

2.5. Fog Computing

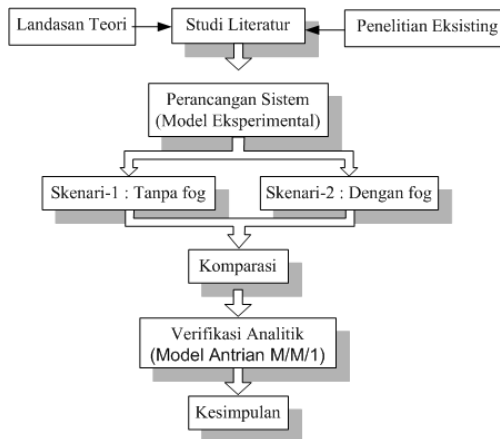
Fog computing [12] adalah struktur komputasi terdesentralisasi yang membawa pemrosesan, penyimpanan, dan kontrol kecerdasan ke perangkat data yang dekat.



Gambar 5 Arsitektur *Fog Computing*.

2.6 Metodologi

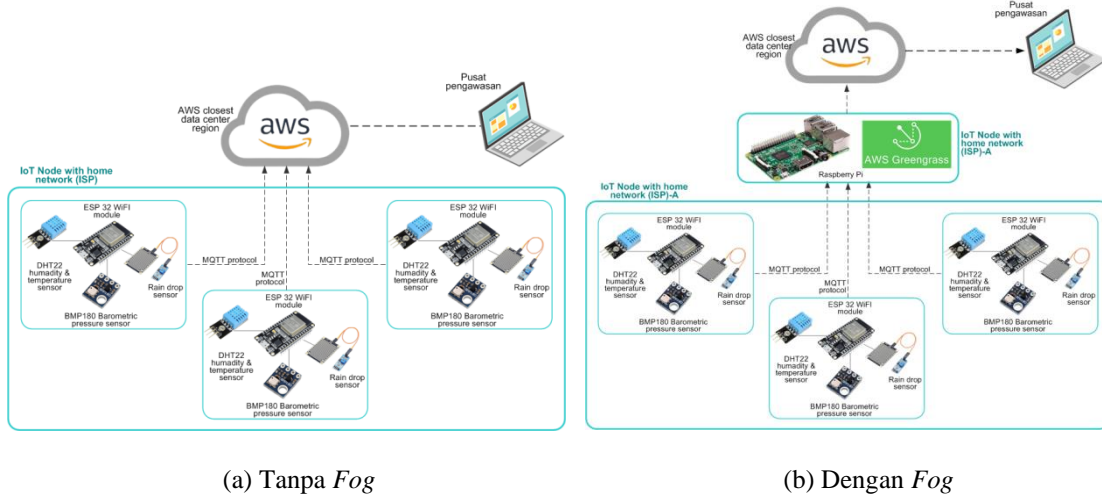
Metodologi yang digunakan dalam penelitian ini digambarkan pada Gambar 6.



Gambar 6 Metodologi Penelitian.

2.7 Perancangan Model Eksperimental

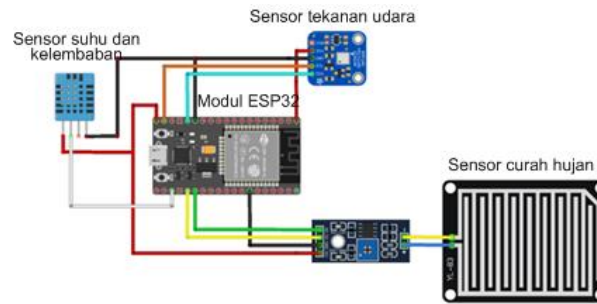
Dalam perancangan model sistem dibuat dua model eksperimen yaitu: tanpa *fog* dan dengan *fog*.



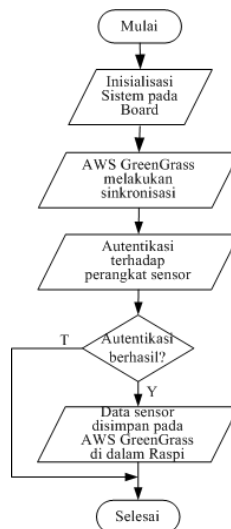
Gambar 7 Model Eksperimen.

2.8 Realisasi

Dalam realisasi koneksi antara sensor dan ESP 32 diperlihatkan pada Gambar 8.



Gambar 8 Realisasi perangkat keras sistem sensor cuaca.



Gambar 9 Diagram Alir pada Raspberry Pi.

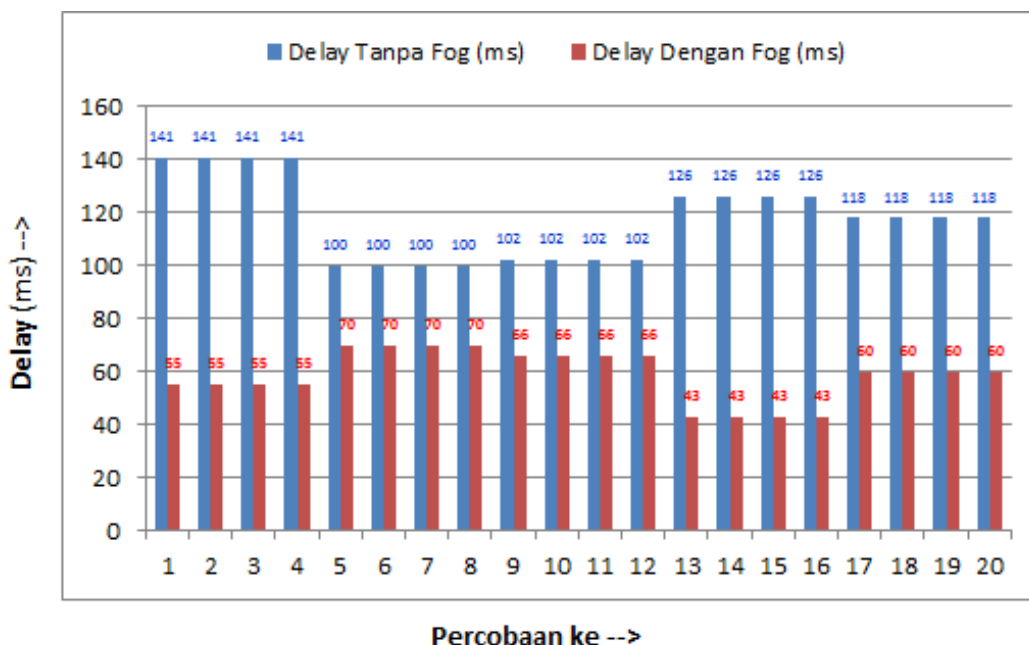
3. HASIL DAN PEMBAHASAN

3.1 Hasil pengujian

Pengujian yang dilakukan adalah pengukuran *delay* untuk dua skenario yaitu tanpa *fog* dan dengan *fog*. Adapun hasil yang diperoleh seperti terlihat pada Tabel 1.

Tabel 1 Hasil Pengukuran *Delay* Untuk Kedua Skenario.

Percobaan ke	Pengaturan urutan penjadwalan	Detik ke	Delay Tanpa Fog (ms)	Delay Dengan Fog (ms)
1	Temperatur	1	141	55
	Kelembaban	2	141	55
	Tekanan Udara	3	141	55
	Curah hujan	4	141	55
2	Temperatur	2	100	70
	Kelembaban	4	100	70
	Tekanan Udara	6	100	70
	Curah hujan	8	100	70
3	Temperatur	3	102	66
	Kelembaban	6	102	66
	Tekanan Udara	9	102	66
	Curah hujan	12	102	66
4	Temperatur	4	126	43
	Kelembaban	8	126	43
	Tekanan Udara	12	126	43
	Curah hujan	16	126	43
5	Temperatur	5	118	60
	Kelembaban	10	118	60
	Tekanan Udara	15	118	60
	Curah hujan	20	118	60



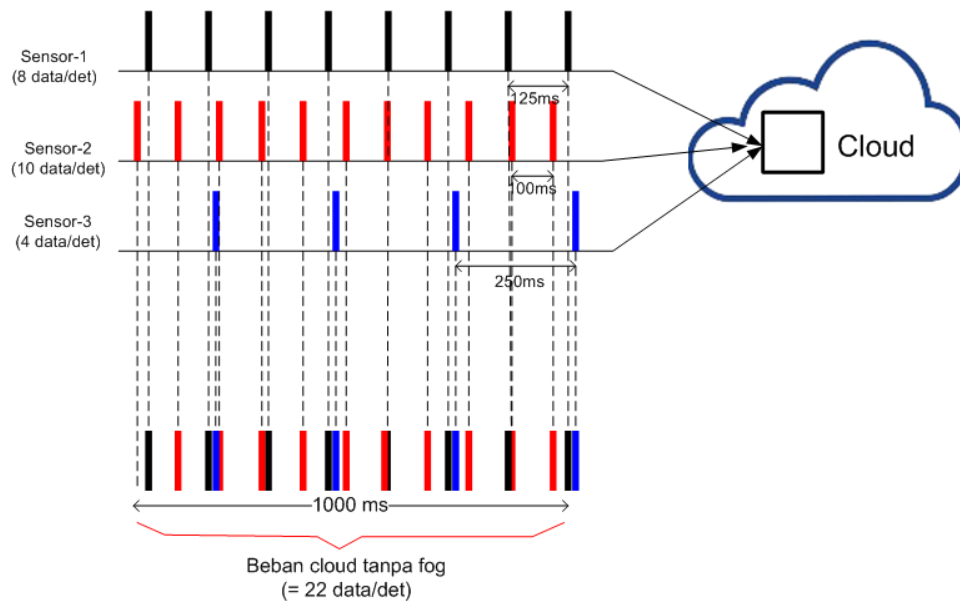
Gambar 10 Perbandingan Delay (*Latency*) Tanpa *Fog* dan Dengan *Fog*.

Dari hasil pengukuran *delay/latency* diperoleh rata-rata 117,4 ms untuk tanpa *fog* dan 58,8 ms untuk yang menggunakan *fog* atau 1,997 kali (dibulatkan 2 kali) lipat *delay* antara tanpa *fog*

dibanding dengan menggunakan *fog*. Atau dengan perkataan lain dengan menggunakan *fog* dua kali lebih cepat dibanding tanpa *fog*.

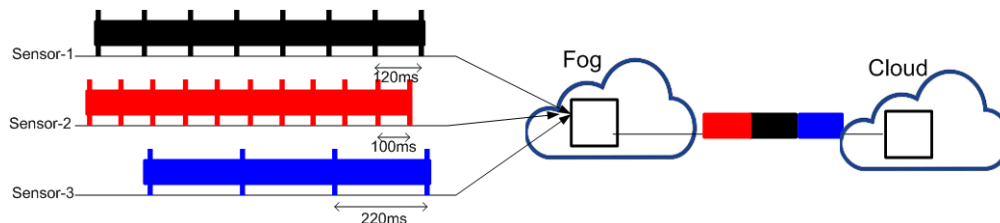
3.2 Analisis Model Trafik (Mekanisme Sistem Antrian)

Dalam analisis pemodelan trafik berikut dilakukan analisis secara umum. Untuk model trafik tanpa *fog* (Gambar 11) dimisalkan terdapat 3 sumber trafik yang berasal dari 3 sensor dimana tiap sensor dijadwalkan pengiriman datanya secara periodik masing-masing sensor setiap 125 ms (atau = 8 data/det), 100 ms (atau = 10 data/det), dan 250 ms (atau = 4 data/det). Maka dari ketiga sensor ini ketika datang pada *cloud* merupakan beban dengan laju kedatangan = $(8+10+4)$ data/det = 22 data/detik seperti diilustrasikan pada Gambar 11.



Gambar 11 Pemodelan Sistem Antrian *Cloud Computing* Tanpa *Fog*.

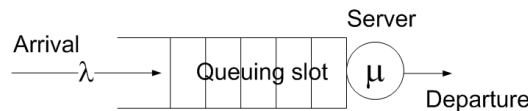
Dengan menggunakan *fog* (Gambar 12) sebagai komputer agregator, proses pengiriman data dapat dilakukan *rescheduling* [13] [14] tanpa mengurangi data rate sehingga masing-masing sensor oleh *fog* akan dikirim 3 set data perdetik dimana masing-masing set data berisi 8 data, 10 data, dan 4 data (Gambar 12). Kejadian ini pada *cloud* akan “nampak” sebagai data dengan laju kedatangan: 3 data perdetik sehingga mengurangi laju kedatangan data atau *request* dari 22 data/det menjadi 3 data/det sebagai hasil dari *pre processing* pada *fog* dalam rangka “meringankan” beban *cloud*.



Gambar 12 Pemodelan Sistem Antrian *Cloud Computing* Dengan *Fog*.

3.3 Analisis Matematis Model Antrian M/M/1

Dari kedua skenario tersebut (tanpa *fog* dan dengan *fog*) akan dianalisis perbandingan kinerjanya berdasarkan parameter *delay* (*latency*) dalam server *cloud*. Menurut ilmu trafik, sebuah server dapat dipandang sebagai sistem antrian seperti layanan pada *bank* dimana dianalogikan prosesor sebagai pelayan (*teller*), dan *buffering memory* (RAM) sebagai kursi tunggu dengan model antrian seperti pada Gambar 13.



Gambar 13 Pemodelan Sistem Antrian.

Jika laju kedatangan data (analogi dengan laju kedatangan nasabah) sebesar λ (data/det), laju pelayanan (analogi dengan laju pelayanan petugas teller) sebesar μ (data/det), dengan mengambil asumsi sumber trafik berasal dari populasi tak hingga dengan pola kedatangan (waktu antar kedatangan) secara acak berdistribusi *Poisson* (eksponensial negatif) dan jumlah slot antrian (ruang tunggu) dianggap tak terbatas, jumlah pelayan (prosesor server) 1 buah, maka ini disebut model antrian M/M/1. Untuk model antrian ini rata-rata *delay* d (lama waktu dalam sistem yang terdiri dari lama waktu tunggu dalam antrian ditambah lama waktu pelayanan dalam server) adalah:

$$d = \frac{1/\mu}{1 - \lambda/\mu}, \quad (1)$$

maka:

- a. Untuk skenario tanpa *fog* ($\lambda_1=22$)
 Dengan menggunakan persamaan (1), diperoleh nilai *delay* tanpa *fog* (d_1) untuk berbagai prosentase beban (λ_1/μ) dari 10% s.d. 90% seperti yang ditunjukkan pada Tabel 2.

Tabel 2 Harga *delay* pada skenario tanpa *fog* (d_1) untuk berbagai prosentase beban *cloud* (λ_1/μ).

% Beban (λ_1/μ)	μ	d_1 (det)
10%	220	0.005051
20%	110	0.011364
30%	73	0.019481
40%	55	0.030303
50%	44	0.045455
60%	37	0.068182
70%	31	0.106061
80%	28	0.181818
90%	24	0.409091

- b. Untuk skenario dengan *fog* ($\lambda_2=3$)
 Dengan cara yang sama (menggunakan persamaan (1)) dan dengan menggunakan kapasitas server *cloud* (μ) yang sama diperoleh harga d_2 untuk prosentase beban yang sesuai dengan (λ_2/μ) seperti yang ditunjukkan pada Tabel 3.

Tabel 3 Harga *delay* pada skenario dengan *fog* (d_2) untuk berbagai prosentase beban *cloud* (λ_2/μ).

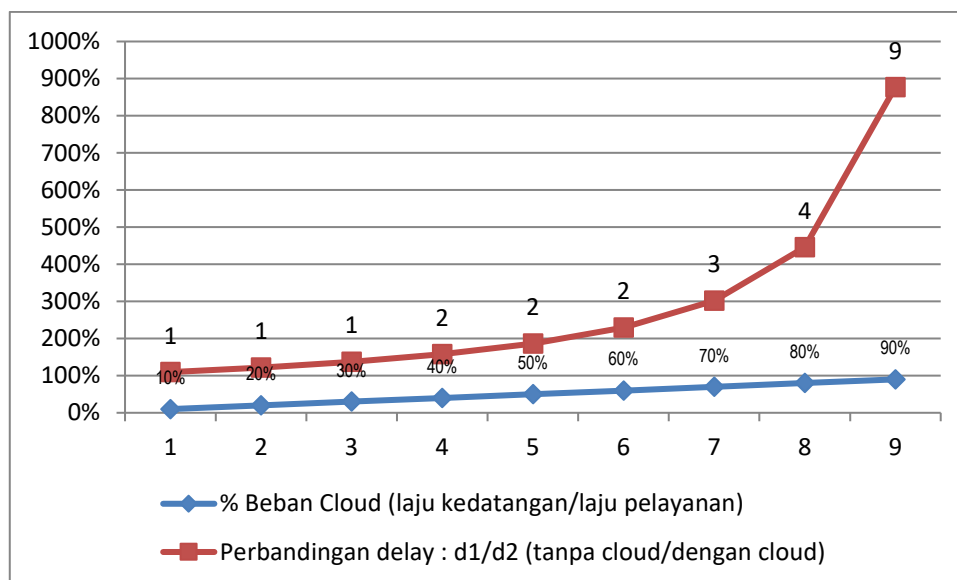
% Beban (λ_2/μ)	μ	$d_2(\text{det})$
1%	220	0.0046083
3%	110	0.0093458
4%	73	0.0142180
5%	55	0.0192308
7%	44	0.0243902
8%	37	0.0297030
10%	31	0.0351759
11%	28	0.0408163
12%	24	0.0466321

Untuk mengetahui perbandingan kinerja tanpa *fog* dibanding dengan *fog* adalah dengan membandingkan harga *delay* dari kedua skenario tersebut seperti pada Tabel 4.

Tabel 4 Perbandingan kinerja tanpa *fog*/dengan *fog* (d_1/d_2).

% Beban (λ_2/μ)	% Beban (λ_2/μ)	$d_1(\text{det})$	$d_2(\text{det})$	d_1/d_2
10%	1%	0.005051	0.004608	1
20%	3%	0.011364	0.009346	1
30%	4%	0.019481	0.014218	1
40%	5%	0.030303	0.019231	2
50%	7%	0.045455	0.02439	2
60%	8%	0.068182	0.029703	2
70%	10%	0.106061	0.035176	3
80%	11%	0.181818	0.040816	4
90%	12%	0.409091	0.046632	9

Maka secara grafis perbandingan kinerja tanpa *fog* dan dengan *fog* untuk berbagai prosentase beban tanpa *fog* dapat digambarkan seperti pada Gambar 14.



Gambar 14 Perbandingan kinerja *delay* tanpa *fog* (d_1) dibanding dengan *fog* (d_2) untuk prosentase pembebanan (λ/μ) yang berbeda-beda mulai dari 10% hingga 90%.

Berdasarkan hasil analisis teoritik (*traffic modeling*) seperti yang ditunjukkan pada Gambar 14, pengaruh penggunaan *fog computing* terhadap *delay* bergantung pada prosentase beban *cloud* yang terbagi ke dalam 4 kategori sebagai berikut:

1. Untuk beban *cloud* di bawah 40%: perbaikan hasil perbaikan perbandingan *delay* tanpa *fog* dengan menggunakan *fog* (d_1/d_2) bernilai 1, artinya tidak ada pengaruh perbaikan dari *fog computing*.
2. Untuk beban 40-60%, *fog computing* memberikan perbaikan *delay* = 2, artinya 2 kali lebih baik (2 kali lebih cepat) dibanding tanpa *fog*.
3. Untuk beban *cloud* 70 dan 80% *fog computing* memberikan perbaikan *delay* masing-masing = 3 dan 4 kali lebih baik (3 kali dan 4 kali lebih cepat) dibanding tanpa *fog*.
4. Perbaikan yang cukup besar yaitu 9 kali lipat terjadi saat beban 90%.

4. KESIMPULAN

Dari hasil pengujian/pengukuran *real*, analisis data hasil pengukuran, serta dari hasil analisis teoritik menggunakan model matematis teori antrian model M/M/1, dapatlah ditarik kesimpulan sebagai berikut:

1. Dari hasil pengukuran menunjukkan dengan menggunakan *fog computing* terjadi peningkatan kinerja *cloud* berdasarkan parameter *delay* sebesar 2 kali lebih baik (lebih cepat) dibanding tanpa *cloud*.
2. Hasil analisis teoritik (*traffic modeling*) pengaruh penggunaan *fog computing* bergantung 2 kali lebih baik (lebih cepat terjadi saat beban *cloud* pada 40-60%).

DAFTAR PUSTAKA

- [1]. Syed AS, Sierra-Sosa D, Kumar A, Elmaghraby A. *IoT in Smart Cities: A Survey of Technologies, Practices and Challenges*. Smart Cities. 2021; 4(2):429-475. <https://doi.org/10.3390/smartcities4020024> (diakses 10 Desember 2021)
- [2]. LeverageLLC, *Introduction o the Internet of Things*, 2018, first edition. <https://www.leverage.com/iot-ebook/how-iot-systems-work> (diakses 1 Oktober 2021)

- [3]. Tandy, Siswono, *Cloud Computing dan Dampaknya Terhadap Bisnis*, ComTech., vol. 4, no. 2, pp. 687-695, 2013.
- [4]. International Data Corporation: *Worldwide Global DataSphere IoT Device and Data Forecast, 2019–2023*. Available online:
- [5]. Maulana, Rizky, 2020, *Analisis Kenyamanan Cuaca Berdasarkan Data Cloud Menggunakan Metode Naive Bayes Classifier*, diunduh dari <http://repositori.usu.ac.id> tanggal 9-2-2021
- [6]. Nurul Hidayah Zulni, Randy Erfa Aaputra, Roswan Latuconsina, 2020, *Perancangan Aplikasi Monitoring Dan Prediksi Cuaca Pada Jemuran Otomatis*, e-Proceeding of Engineering: Vol.7, No.2 Agustus 2020
- [7]. Pccontrol, *Pengetahuan Dasar dan Pemrograman Raspberry Pi* (<https://pccontrol.wordpress.com/2014/06/17/pengetahuan-dasar-dan-pemrograman-raspberry-pi/> diakses tanggal 8 Agustus 2020).
- [8]. Wendi Harjupa, 2017, *Pentingnya Pemanfaatan Teknologi Untuk Pemantauan Cuaca*, diunduh dari <https://indonesiantoday.id/2017/04/30/pentingnya-pemanfaatan-teknologi-untuk-pemantauan-cuaca/> (diakses tanggal 10 Agustus 2021)
- [9]. *How Fog Computing Can Solve the IoT Challenges* <https://www.embeddedcomputing.com/technology/iot/wireless-sensor-networks/how-fog-computing-can-solve-the-iot-challenges> (diakses 18 Agustus 2021)
- [10]. T. Liu, *Digital-Output relative humidity & temperature sensor/module DHT22*, New York Aosong Electron., vol. 22, pp. 1–10, 2015, <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf> (diakses 10 Agustus 2021)
- [11]. HK Shan Hai Group Limited, *Snow & Raindrops Detection Sensor Module*, https://www.terraelectronica.ru/pdf/show?pdf_file=%252Fz%252FDatasheet%252FS%252FSnow%2B%2526%2BRaindrops%2BDetection.pdf (diakses 12 Agustus 2021)
- [12]. Bosch, *Datasheet BMP180 Digital pressure sensor*, <https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf> (diakses 15 Agustus 2022)
- [13]. Sathish, Mani & Meenakshisundaram, Iyapparaja. (2020). *Improving quality-of-service in fog computing through efficient resource allocation*. Computational Intelligence. 36. 10.1111/coin.12285.
- [14]. Mondragón-Ruiz, G., Tenorio-Trigoso, A., Castillo-Cara, M. *et al. An experimental study of fog and cloud computing in CEP-based Real-Time IoT applications. J Cloud Comp* 10, 32 (2021). <https://doi.org/10.1186/s13677-021-00245-7> (diakses 10 Desember 2021)