

Pengujian Stress Testing MQTT dan API pada Sistem Monitoring Suhu Tubuh, Detak Jantung, Saturasi Oksigen, dan Tekanan Darah berbasis Internet of Things

I Nyoman Tri Aditya^{a1}, Kadek Suar Wibawa^{a2}, I Putu Agung Bayupati^{a3}

^aProgram Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana, Bali
e-mail: ¹mangtri93@gmail.com, ²suar_wibawa@unud.ac.id, ³bayupati@unud.ac.id

Abstrak

Pengujian alat monitoring suhu tubuh, detak jantung, saturasi oksigen, dan tekanan darah berbasis internet of things dilakukan untuk memastikan ketahanan dari alat serta perangkat lunak pendukungnya dibutuhkan stress testing. Stress testing yang diujikan pada protokol MQTT antara lain; pengujian waktu koneksi dan konsumsi daya, pengujian publish dan subscribe, dan pengujian alat secara berulang. Stress testing yang diujikan pada API yaitu pengujian dengan beban hingga 100 user dengan ramp up period 10 detik dan loop count 1 yang diuji pada API. Hasil pengujian adalah alat mampu berjalan dengan baik pada kurun waktu hingga 60 menit dan mengkonsumsi sedikit daya, transfer data yang cepat, ringan serta dengan kesesuaian antara data yang di-publish dan di-subscribe cukup akurat dengan error sebesar 0.5%.

Kata kunci: API, Internet of Things, Monitoring, MQTT, Stress Testing,

Abstract

Testing of internet of things-based monitoring tools for body temperature, heart rate, oxygen saturation, and blood pressure is carried out to ensure that the resistance of the tools and their supporting software requires stress testing. The stress testing tested on the MQTT protocol includes; connection time and power consumption testing, publish and subscribe testing, and iterative tool testing. The stress testing tested on the API is testing with a load of up to 100 users with a ramp up period of 10 seconds and a loop count of 1 which is tested on the API. The test results are that the tool is able to run well for a period of up to 60 minutes and consumes little power, data transfer is fast, light and the compatibility between published and subscribed data is quite accurate with an error of 0.5%.

Keywords : API, Internet of Things, Monitoring, MQTT, Stress Testing

1. Pendahuluan

Perkembangan teknologi yang pesat saat ini memicu hadirnya revolusi industri 4.0. Hampir semua bidang industri saat ini, mulai beralih dari hal yang konvensional menuju digital. Salah satu teknologi yang kini populer adalah *internet of things*. Menuju digitalisasi industri pada bidang kesehatan, *internet of things* dapat diterapkan pada alat medis yang biasa digunakan seperti, mengukur suhu tubuh, detak jantung, saturasi oksigen, dan tekanan darah. Penerapan IoT pada alat medis, dapat secara otomatis membuat alat berkomunikasi dengan internet, sehingga data hasil pemeriksaan pasien dapat disimpan ke dalam sistem dan dapat dijadikan riwayat kesehatan.

Alat yang dibuat menggunakan mikrokontroler arduino sebagai *board* utama dan NodeMCU ESP8266 sebagai modul WiFi. Sensor dan komponen yang digunakan untuk melakukan pengukuran suhu tubuh adalah GY-906, pengukuran detak jantung dan saturasi oksigen adalah MAX30100, dan pengukuran tekanan darah adalah tensimeter CK-101. Alat terhubung dengan aplikasi *mobile* pendukung untuk memonitoring pengukuran yang

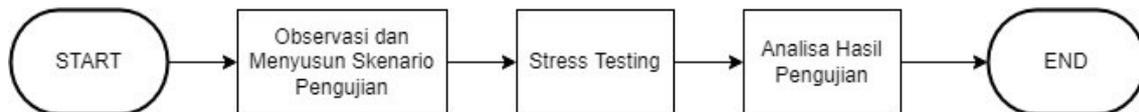
berlangsung pada alat langsung melalui *smartphone* yang dihubungkan melalui protokol MQTT serta menyimpan hasil pengukuran ke sistem sesuai dengan *user* yang diperiksa.

2. Metodologi Penelitian

Metodologi penelitian yang memaparkan bagaimana alur tahapan *stress testing* MQTT dan API pada sistem monitoring suhu tubuh, detak jantung, saturasi oksigen dan tekanan darah berbasis internet of things adalah sebagai berikut

2.1 Alur Penelitian

Alur penelitian merupakan serangkaian kegiatan yang dirancang dan dilakukan untuk mencapai tujuan dan hasil yang diinginkan. Berikut alur penelitian yang digunakan pada penelitian ini.

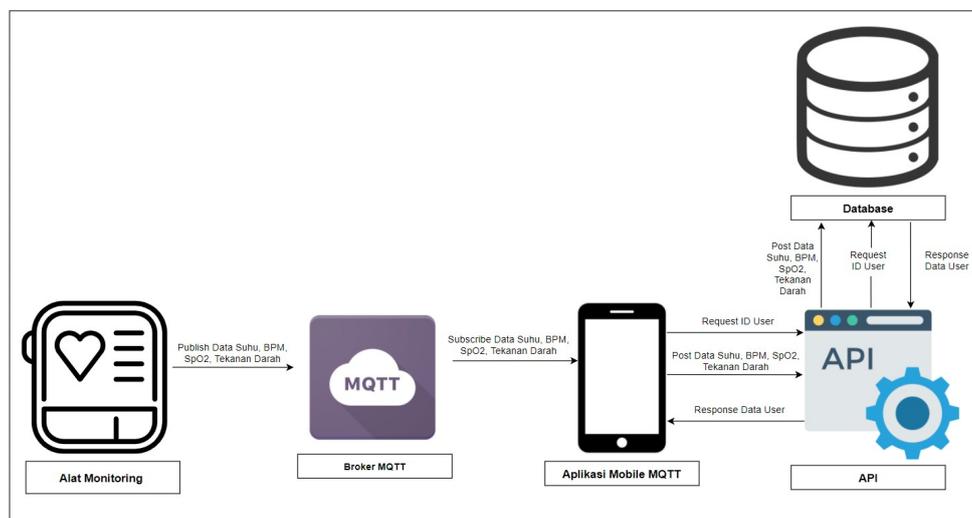


Gambar 1 Alur Penelitian

Gambar 1 adalah alur penelitian *stress testing* MQTT dan API pada sistem monitoring suhu tubuh, detak jantung, saturasi oksigen dan tekanan darah berbasis *internet of things*. Penelitian dimulai dengan melakukan observasi dan menyusun skenario pengujian, kemudian dilanjutkan dengan melakukan *stress testing* sesuai dengan skenario pengujian dan diakhiri dengan menganalisa hasil pengujian.

2.2 Gambaran Umum

Gambaran umum sistem merupakan bagan yang menjelaskan seperti apa bentuk dari suatu sistem, entitas yang terlibat serta alur kerja atau komunikasi dari sistem tersebut. Berikut gambaran umum pada penelitian ini



Gambar 2 Gambaran Umum Sistem

Gambar 2 menjelaskan alur komunikasi data antar *device* yang digunakan pada penelitian ini. Pertama, aplikasi *mobile* MQTT melakukan *request data user* (pasien) ke API dengan mengirimkan ID pasien yang diinput pada aplikasi *mobile* MQTT. Selanjutnya API akan melanjutkan *request* data *user* ke *database*. *Database* memberikan *response data user* sesuai ID pasien yang diberikan ke API dan diteruskan kembali ke aplikasi *mobile* MQTT. Data *user* berhasil didapatkan oleh aplikasi *mobile* MQTT dan dapat terhubung ke alat monitoring. Alat monitoring melakukan *publish* data pengukuran ke broker MQTT. Selanjutnya aplikasi *mobile* MQTT melakukan *subscribe* data pengukuran ke *broker* dan *topic* yang sama. Ketika pengukuran telah selesai, kirim hasil pengukuran ke API untuk disimpan ke dalam *database*.

2.3 Skenario Pengujian

Skenario pengujian merupakan tahapan atau simulasi yang dilakukan untuk menguji fungsionalitas serta ketahanan dari alat dan sistem agar sesuai tujuan dan manfaat serta menemukan kesalahan atau *error* dari sistem. *Broker* dan *topic* yang digunakan pada *publisher* dan *subscriber* adalah sebagai berikut

Tabel 1. Alamat *Broker* MQTT

Nama <i>Broker</i>	HiveMQ
Alamat <i>Broker</i>	http://broker.mqtt-dashboard.com/
TCP Port	1883

Tabel 1 merupakan *broker* yang digunakan pada penelitian ini. HiveMQ merupakan salah satu *broker* MQTT yang dapat digunakan secara gratis. *Broker* ini berjalan pada port TCP 1883. Selanjutnya *topic* yang didefinisikan pada *publisher* dan *subscriber*

Tabel 2 Topic Publish dan Subscribe

N O	<i>Topic Publish</i> (Alat Monitoring)	<i>Topic Subscribe</i> (Aplikasi <i>Mobile</i> MQTT)	Keterangan
1	iot12/temp	iot12/#	Suhu tubuh
2	iot12/bpm		Detak Jantung
3	iot12/spo2		Saturasi Oksigen
4	iot12/sistole		Tekanan Darah Sistole
5	iot12/diastole		Tekanan Darah Diastole

Tabel 2 merupakan *topic – topic* yang digunakan pada penelitian ini. Terdapat 5 *topic* yang digunakan pada *publish* untuk mengirim masing-masing data pengukuran oleh alat. Sedangkan pada *subscriber* cukup menggunakan 1 *topic wildcard multilevel* saja untuk menerima semua data dari *publisher*. Adapaun *device* yang digunakan pada *subscriber* adalah *smartphone* Redmi 9T dengan spesifikasi RAM 4GB, baterai 6000mAh, dan OS Android 10 dan untuk *powering* alat monitoring menggunakan *powerbank* ACMIC dengan kapasitas 10000mAh.

3. Kajian Pustaka

Kajian Pustaka merupakan segala sumber pustaka terkait dengan penelitian yang dilakukan. Pustaka terkait penelitian ini akan dijelaskan sebagai berikut.

3.1 Internet of Things

Internet of Things atau yang sering disebut dengan IoT adalah gagasan bahwa semua objek di dunia nyata dapat saling berkomunikasi sebagai bagian dari sistem terintegrasi yang menggunakan jaringan internet sebagai koneksinya [1]. IoT merupakan teknologi yang digunakan untuk menyatukan semua perangkat melalui media internet tanpa perlu terhubung langsung dengan manusia. Peran manusia dalam *internet of things* adalah untuk melakukan kontrol objek atau perangkat IoT ini dari jarak jauh melalui perangkat elektronik yang dapat terhubung ke internet, seperti *smartphone* dan komputer. Prasyarat untuk perangkat yang disebut *internet of things* adalah perangkat yang terhubung ke mikrokontroler yang diprogram dan dapat terhubung ke *smartphone* pengguna melalui internet [2].

3.2 Protokol MQTT

Message Queuing Telemetry Transport (MQTT) adalah protokol pesan *publish/subscribe* ringan yang digunakan di atas paket protokol TCP/IP. Protokol ini memiliki ukuran paket data yang kecil, *overhead* yang rendah (minimal 2 *gigabyte*), dan konsumsi daya yang rendah. MQTT terbuka, sederhana dan mudah diimplementasikan, serta satu *server* MQTT dapat menangani ribuan *client* jarak jauh. Properti ini membuatnya cocok untuk digunakan dalam aplikasi *Internet of Things*. MQTT memiliki beberapa komponen penting, diantaranya [3]:

- a. *Broker*, diibaratkan sebagai *server* yang memiliki alamat IP untuk menangani data dari *publisher* dan *subscriber*.
- b. *Publisher*, merupakan kemampuan *device* untuk mengirimkan datanya ke *broker*.

- c. *Subscriber*, merupakan kemampuan *device* untuk menerima data yang dikirim oleh *publisher*.
- d. *Topic*, merupakan kunci penghubung antara *publisher* dan *subscriber*.

3.3 Application Programming Interface (API)

Application Programming Interface (API) adalah teknologi antarmuka virtual yang memungkinkan pertukaran informasi antar aplikasi atau sistem. API merupakan protokol yang digunakan untuk mengintegrasikan sebuah sistem dengan sistem lainnya. Contoh, mengintegrasikan *website* dengan aplikasi *mobile*. Jadi, API dapat dikatakan sebagai jembatan penghubung antara sistem [4].

3.4 Stress Testing

Stress Testing digunakan untuk menguji keandalan, stabilitas, serta menentukan batas dari perangkat keras ataupun perangkat lunak. Pengujian ini dapat menentukan gaya pemulihan sistem dan penanganan kesalahan dalam kondisi beban yang sangat tinggi. Uji di luar titik operasi normal untuk mengevaluasi bagaimana sistem berperilaku dalam kondisi ekstrem seperti itu. *Stress Testing* dilakukan untuk memastikan bahwa sistem berjalan efektif dalam situasi kritis [5].

4. Hasil dan Pembahasan

Pengujian stress testing dilakukan dengan mengacu pada skenario pengujian yang disebutkan sebelumnya. Pengujian dilakukan ke alat monitoring, aplikasi *mobile* MQTT, dan API sistem. Hasil dan pembahasan, serta analisa pada penelitian ini, dapat dilihat sebagai berikut.

4.1 Pengujian Waktu Koneksi MQTT dan Konsumsi Daya

Pengujian ini adalah uji coba yang bertujuan guna mengetahui seberapa lama *device* dapat terhubung dengan *broker* MQTT tanpa terputus. Pengujian dilakukan dalam beberapa rentang waktu yaitu, 10, 30, dan 60 menit. Langkah pengujian dilakukan dengan menghidupkan alat monitoring dan aplikasi *mobile* MQTT kemudian melakukan koneksi ke *broker* MQTT yang telah ditentukan dalam waktu hingga 60 menit serta melihat daya yang dikonsumsi pada masing-masing *device*. Hasil pengujian dapat dilihat sebagai berikut.

Tabel 3 Pengujian Waktu Koneksi MQTT

Device	10 menit	30 menit	60 menit	Keterangan
Alat Monitoring	v	v	v	Sukses
Aplikasi <i>Mobile</i> MQTT	v	v	v	Sukses

Tabel 3 merupakan hasil pengujian waktu konektivitas ke *broker* MQTT pada alat monitoring dan aplikasi *mobile*. Alat dan aplikasi berhasil terhubung pada *broker* MQTT dalam waktu 10, 30, dan 60 menit tanpa mengalami masalah dan tanpa terputus dari *broker*. Adapun konsumsi baterai yang terpakai pada masing-masing alat dan *smartphone* adalah sebagai berikut.

Tabel 4 Pengujian Konsumsi Daya pada Alat dan Smartphone

Device	10 menit	30 menit	60 menit
Alat Monitoring	-1%	-2%	-4%
Aplikasi <i>Mobile</i> MQTT	-2%	-4%	-7%

Tabel 4 merupakan hasil pengujian konsumsi daya pada alat monitoring dan aplikasi *mobile*. Alat monitoring dengan daya dari *powerbank* yang memiliki kapasitas 1000mAh menghabiskan daya sebesar 4% selama 1 jam pemakaian yang artinya sekitar 400 mAh perjam sedangkan pada aplikasi *mobile* menghabiskan daya sebesar 7% selama 1 jam pemakaian. Hasil ini tergantung dari jenis *powerbank* dan *smartphone* yang digunakan.

4.2 Pengujian Data Publish dan Subscribe

Pengujian data *publish* dan *subscribe* dilakukan untuk mengetahui jumlah data, kesesuaian data serta kemungkinan data *error* antara yang di-*publish* serta di-*subscribe* dalam kurun waktu 10, 30, dan 60 menit. Langkah pengujian yang dilakukan adalah alat monitoring digunakan untuk mengukur suhu tubuh, detak jantung, saturasi oksigen dan

tekanan darah. Setiap pengukuran memiliki *topic* nya masing-masing. Pengukuran suhu tubuh, detak jantung, dan saturasi oksigen ketika digunakan melakukan *publish* data setiap 1 detik sedangkan pengukuran tekanan darah melakukan *publish* data hanya ketika pengukuran selesai dilakukan. Alat monitoring melakukan *publish* data pada masing-masing *topic* pengukuran yang telah ditentukan dan aplikasi *mobile* melakukan *subscribe* data yang di-*publish* oleh alat monitoring sesuai *topic* masing-masing pengukuran. Hasil pengujian data *publish* dan *subscribe* dalam pengujian selama 10 menit adalah sebagai berikut.

Tabel 5 Pengujian Data Publish dan Subscribe Selama 10 Menit

Objek Pengukuran	Topic	Jumlah data terkirim (<i>publisher</i>)	Jumlah data diterima (<i>subscriber</i>)	Selisih data	Error
Suhu Tubuh	iot12/suhu	299	299	0	0%
Detak Jantung	iot12/bpm	299	299	0	0%
Saturasi Oksigen	iot12/spo2	299	299	0	0%
Tekanan Darah (Sistole)	iot12/sistole	1	1	0	0%
Tekanan Darah (Diastole)	iot12/diastole	1	1	0	0%

Tabel 5 merupakan hasil pengujian data *publish* dan *subscribe* dalam kurun waktu 10 menit. Terlihat, dari 10 menit pengujian, alat monitoring melakukan *publish* data pengukuran sebanyak 299 data masing-masing pada pengukuran suhu, detak jantung, saturasi oksigen dan aplikasi *mobile* berhasil melakukan *subscribe* masing-masing sebanyak 299 data pengukuran suhu, detak jantung, saturasi oksigen. Sedangkan pada pengukuran tekanan darah dilakukan satu kali dan berhasil di-*subscribe* oleh aplikasi *mobile* MQTT. Hasil dari pengujian 10 menit ini, didapatkan selisih data antara data yang di-*publish* dan di-*subscribe* pada semua pengukuran adalah sebesar 0 dengan error 0%.

Tabel 6 Pengujian Data Publish dan Subscribe Selama 30 Menit

Objek Pengukuran	Topic	Jumlah data terkirim (<i>publisher</i>)	Jumlah data diterima (<i>subscriber</i>)	Selisih data	Error
Suhu Tubuh	iot12/suhu	1795	1779	16	0.8%
Detak Jantung	iot12/bpm	1795	1779	16	0.8%
Saturasi Oksigen	iot12/spo2	1795	1779	16	0.8%
Tekanan Darah (Sistole)	iot12/sistole	5	5	0	0%
Tekanan Darah (Diastole)	iot12/diastole	5	5	0	0%

Tabel 6 merupakan hasil pengujian data *publish* dan *subscribe* dalam kurun waktu 30 menit. Terlihat, dari 10 menit pengujian, alat monitoring melakukan *publish* data pengukuran sebanyak 1795 data pada masing-masing pengukuran suhu, detak jantung, saturasi oksigen dan aplikasi *mobile* berhasil melakukan *subscribe* masing-masing sebanyak 1779 data pengukuran suhu, detak jantung, saturasi oksigen. Sedangkan pada pengukuran tekanan darah dilakukan 5 kali dan berhasil di-*subscribe* semua oleh aplikasi *mobile* MQTT. Hasil dari pengujian selama 30 menit ini, didapatkan selisih data antara data yang di *publish* dan di *subscribe* pada pengukuran suhu, detak jantung, saturasi oksigen adalah sebesar 16 data dengan error 0.8% dan selisih sebesar 0 data dengan error 0% pada pengukuran tekanan darah.

Tabel 7 Pengujian Data Publish dan Subscribe Selama 60 Menit

Objek Pengukuran	Topic	Jumlah data terkirim	Jumlah data diterima	Selisih data	Error
------------------	-------	----------------------	----------------------	--------------	-------

		(<i>Publisher</i>)	(<i>Subscriber</i>)		
Suhu Tubuh	iot12/suhu	3572	3552	20	0.5%
Detak Jantung	iot12/bpm	3572	3552	20	0.5%
Saturasi Oksigen	iot12/spo2	3572	3552	20	0.5%
Tekanan Darah (Sistole)	iot12/sistole	10	10	0	0%
Tekanan Darah (Diastole)	iot12/diastole	10	10	0	0%

Tabel 7 merupakan hasil pengujian data *publish* dan *subscribe* dalam kurun waktu 60 menit. Terlihat, dari 10 menit pengujian, alat monitoring melakukan *publish* sebanyak 3572 data pengukuran pada masing-masing pengukuran suhu, detak jantung, saturasi oksigen dan aplikasi *mobile* berhasil melakukan *subscribe* masing-masing sebanyak 3552 data pengukuran suhu, detak jantung, saturasi oksigen. Sedangkan pada pengukuran tekanan darah dilakukan 10 kali dan berhasil di-*subscribe* semua oleh aplikasi *mobile* MQTT. Hasil dari pengujian selama 60 menit ini, didapatkan selisih data antara data yang di *publish* dan di *subscribe* pada pengukuran suhu, detak jantung, saturasi oksigen adalah sebesar 20 data dengan error 0.5% dan selisih sebesar 0 data dengan error 0% pada pengukuran tekanan darah.

4.3 Pengujian Penggunaan Alat dan Aplikasi

Pengujian penggunaan alat dan aplikasi dilakukan hingga 25 kali percobaan. Pengukuran dilakukan secara lengkap, mulai dari mendapatkan data pengukuran oleh alat monitoring, melakukan *publish* data ke aplikasi *mobile* MQTT, hingga menyimpan hasil pengukuran ke sistem secara berulang. Langkah pengujian diawali dengan menginput ID *user* pada aplikasi *mobile* MQTT untuk mendapatkan data *user* yang sedang diperiksa. Selanjutnya dilakukan pengukuran suhu tubuh, detak jantung, saturasi oksigen dan tekanan darah kepada *user*. Pengukuran diawali dengan mengukur tekanan darah, dilanjutkan dengan mengukur detak jantung dan saturasi oksigen, dan terakhir suhu tubuh. Data pengukuran dapat langsung dilihat pada aplikasi *mobile* MQTT. Apabila pengukuran telah selesai, simpan hasil pengukuran melalui aplikasi *mobile* MQTT ke sistem. Hasil pengujian alat dan aplikasi dapat dilihat sebagai berikut.

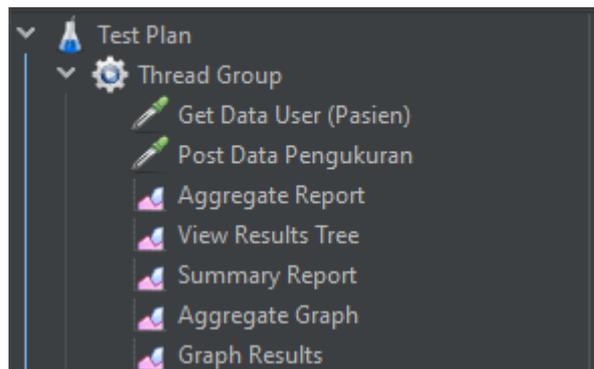
Tabel 8 Pengujian Penggunaan Alat dan Aplikasi

Uji Coba ke-	Rincian Ujicoba					Keterangan
	I	II	III	IV	V	
1-5	P	P	P	P	P	Pengukuran berhasil dilakukan, data pengukuran dari alat terkirim ke aplikasi <i>mobile</i> dan hasil pengukuran dapat disimpan ke sistem
6-10	P	P	P	P	P	Pengukuran berhasil dilakukan, data pengukuran dari alat terkirim ke aplikasi <i>mobile</i> dan hasil pengukuran dapat disimpan ke sistem
11-15	P	P	P	P	P	Pengukuran berhasil dilakukan, data pengukuran dari alat terkirim ke aplikasi <i>mobile</i> dan hasil pengukuran dapat disimpan ke sistem
16-20	P	P	P	X	P	Pengukuran berhasil dilakukan, data pengukuran dari alat terkirim ke aplikasi <i>mobile</i> dan hasil pengukuran dapat disimpan ke sistem, tetapi pada percobaan ke-19 alat pengukur suhu tubuh menghasilkan data <i>error</i> atau gagal mendapatkan nilai suhu tubuh <i>user</i>
21-25	P	X	P	P	P	Pengukuran berhasil dilakukan, data pengukuran dari alat terkirim ke aplikasi <i>mobile</i> dan hasil pengukuran dapat disimpan ke sistem tetapi, pada percobaan ke-22 data tekanan darah tidak terkirim ke alat dan aplikasi,

Tabel 8 merupakan hasil pengujian alat dan aplikasi secara lengkap dengan melakukan pengukuran ke beberapa pengguna sebanyak 25 kali. Terlihat, dari 25 kali pengujian, terdapat 2 kali pengujian yang gagal pada uji coba ke-19 dan ke-22. Uji coba ke-19, pengukur suhu tubuh menghasilkan data *error* atau tidak dapat mengukur suhu tubuh. Sedangkan pada ujicoba ke-22 data pengukuran tekanan darah tidak terkirim ke mikrokontroller sehingga tidak ada data yang di-*publish* ke aplikasi *mobile* MQTT.

4.4 Pengujian API Sistem

Pengujian API sistem menggunakan *tools* JMeter dengan memasukkan alamat API untuk *get* data *user* dan *post* data pengukuran. Pengujian ini adalah uji coba yang bertujuan guna mengetahui kehandalan sistem dalam menangani banyak beban secara bersamaan. Pengujian dilakukan dengan menguji alamat API yang memiliki method GET dan POST. Skenario pengujian pada apache JMeter dapat dilihat sebagai berikut.



Gambar 3 Skenario Pengujian pada Apache Jmeter

Gambar 3 merupakan skenario pengujian API pada tools Jmeter. Pengujian ini dilakukan dengan memberikan beban user sebesar 10, 25, 50, 100 dengan periode *ramp up* 10 detik dan *loop count* 1 pada masing-masing alamat API. Hasil pengujian dengan beban 10 user adalah sebagai berikut.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Get Data User (...)	10	255	169	913	219.30	0.00%	1.1/sec	0.77	0.16	714.0
Post Data Peng...	10	146	110	379	77.86	0.00%	1.2/sec	0.43	0.33	364.0
TOTAL	20	200	110	913	173.47	0.00%	2.2/sec	1.15	0.45	539.0

Gambar 4 Pengujian API dengan beban 10 user

Gambar 4 merupakan hasil pengujian dengan beban sampel sebesar *user* 10 pada report JMeter. Terlihat, pengujian mendapatkan hasil dengan *error* 0% pada kedua API yang diujikan.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Get Data User (...)	25	203	172	461	73.95	0.00%	2.6/sec	1.78	0.36	714
Post Data Peng...	25	116	100	175	13.96	0.00%	2.6/sec	0.92	0.71	364
TOTAL	50	159	100	461	68.72	0.00%	5.0/sec	2.66	1.04	539

Gambar 5 Pengujian API dengan beban 25 user

Gambar 5 merupakan hasil pengujian dengan beban sampel sebesar *user* 25 pada report JMeter. Terlihat, pengujian mendapatkan hasil dengan *error* 0% pada kedua API yang diujikan.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Get Data User (...)	50	286	171	1436	210.90	36.00%	4.7/sec	13.68	0.66	2980.8
Post Data Peng...	50	158	99	691	114.67	42.00%	4.7/sec	14.53	1.29	3155.7
TOTAL	100	222	99	1436	181.45	39.00%	9.2/sec	27.53	1.90	3068.2

Gambar 6 Pengujian API dengan beban 50 user

Gambar 6 merupakan hasil pengujian dengan beban sampel sebesar *user 25* pada report JMeter. Terlihat, pengujian mendapatkan hasil dengan *error 8%* pada kedua API yang diujikan.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Get Data User (...)	100	302	163	5873	621.98	65.00%	9.8/sec	45.87	1.36	4807.0
Post Data Peng...	100	136	98	551	77.97	66.00%	9.8/sec	45.55	2.69	4751.0
TOTAL	200	219	98	5873	450.93	65.50%	19.2/sec	89.58	3.97	4779.0

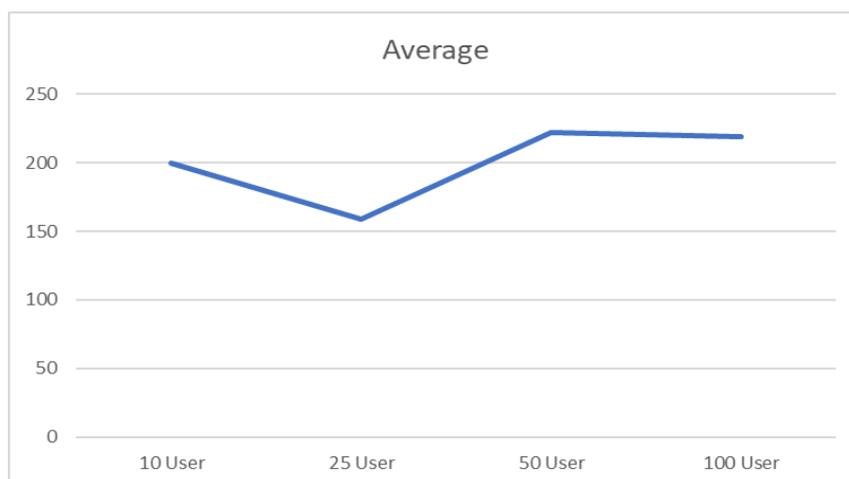
Gambar 7 Pengujian API dengan beban 100 user

Gambar 7 merupakan hasil pengujian dengan beban sampel sebesar *user 25* pada report JMeter. Terlihat, pengujian mendapatkan hasil dengan *error 54.50%* pada kedua API yang diujikan..

Tabel 9 Rangkuman Hasil Pengujian

Parameter	10 User	25 User	50 User	100 User
Average	200	159	222	219
Deviation	173.47	68.72	181.45	450.93
Error	0%	0%	39%	65.5%
Throughput	2.2/sec	5.0/sec	9.2/sec	19.2/sec
Byte Send	0.45	1.04	1.90	3.97
Byte Received	1.15	2.66	27.53	89.58

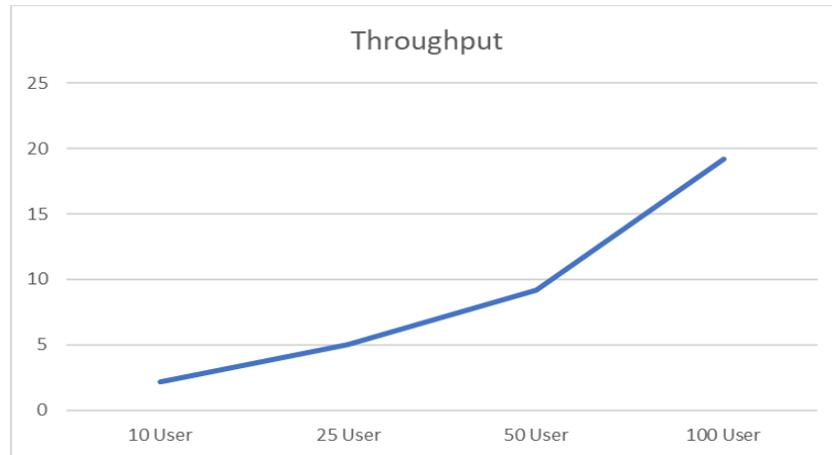
Tabel 9 merupakan rangkuman hasil pengujian stress testing API dengan beban user sebesar 10, 25, 50, 100 pada masing-masing alamat API. Pengujian yang berhasil dilakukan sebelumnya, memunculkan beberapa hasil yang bisa dianalisa. Pengujian *stress testing* API menghasilkan beberapa poin penting seperti *average*, *throughput*, *deviation*, dan *error*. Berikut grafik analisa dari hasil pengujian *average*.



Gambar 8 Grafik Hasil Pengujian Average

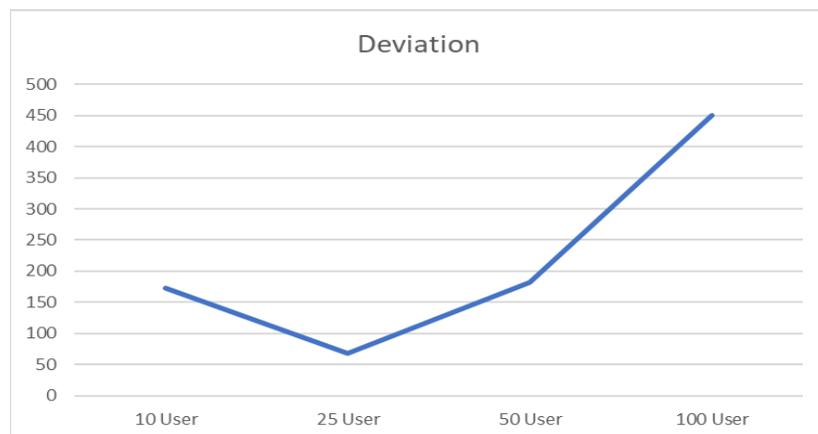
Gambar 8 merupakan hasil pengujian *average* dalam bentuk grafik pada satuan waktu ms. Terlihat dalam pengujian dengan beban 10, 25, 50, 100 user, tidak terdapat perbedaan yang signifikan pada masing-masing kategori beban yang diujikan, yang artinya sistem dapat

berjalan secara konsisten. Selanjutnya, analisa hasil pengujian *throughput* dapat dilihat sebagai berikut.



Gambar 9 Grafik Hasil Pengujian *Throughput*

Gambar 9 merupakan hasil pengujian throughput API dalam bentuk grafik setiap detiknya. Terlihat, *throughput* mengalami peningkatan seiring beban yang bertambah, mulai dari 10, 25, 50, dan 100 user. Kenaikan *throughput* menunjukkan, sistem dapat menangani beban dengan baik seiring kenaikan beban akses yang diterimanya, karena semakin meningkatnya *throughput* sebanding dengan peningkatan *request user*. Selanjutnya, analisa hasil pengujian *deviation* dapat dilihat sebagai berikut.



Gambar 10 Grafik Hasil Pengujian *Deviation*

Gambar 10 merupakan hasil pengujian *deviation* dalam bentuk grafik pada satuan waktu ms. Terlihat pada pengujian beban 25 *user* mengalami penurunan, kemudian pada beban 50 *user* mengalami kenaikan, tetapi masih dalam nilai yang tidak terlalu jauh dengan beban 10 *user*. Sedangkan pada beban 100 *user* kembali mengalami kenaikan yang cukup jauh. Hasil *deviation* pada beban sampai 50 *user* lebih kecil dari hasil *average* yang artinya performa sistem semakin baik sedangkan pada beban 100 *user* mengalami kenaikan lebih dari hasil *average*, yang artinya kinerja sistem berat atau tidak berjalan dengan baik pada saat menangani beban 100 *user*.

5. Kesimpulan

Pengujian dilakukan dalam beberapa skenario pengujian yang menguji coba perangkat keras, perangkat lunak, serta protokol komunikasi yang digunakan. Pengujian perangkat keras yaitu pada alat monitoring menghasilkan alat yang mampu berjalan secara baik, menghasilkan pengukuran yang akurat serta hemat daya. Alat mampu digunakan secara baik hingga 25 kali percobaan secara berturut-turut dengan hanya mengalami 2 kali kegagalan. Penggunaan alat

selama 60 menit hanya menghabiskan daya sekitar sebesar 400 mAh. Pengujian data *publish* dan *subscribe* selama 60 menit menghasilkan banyak data yang di-*publish* pada pengukuran suhu tubuh, detak jantung, dan saturasi oksigen masing-masing sebesar 3572 data dan data yang berhasil di-*subscribe* oleh aplikasi *mobile* MQTT sebanyak 3552 data dengan *error* 0.5%. Selanjutnya, *stress testing* API menggunakan beban user sebesar 10, 25, 50, dan 100 dengan count loop 1 dan 10 detik untuk periode *ramp up* pada masing-masing alamat API yang diuji. Hasil yang didapatkan diantaranya, pada *average* didapatkan hasil yang konstan pada semua kategori sampel, *throughput* mendapatkan nilai yang meningkat berbanding lurus dengan kenaikan jumlah *request* dari sampel, dan *deviation* didapatkan nilai menurun pada pengujian 25 *user*, kemudian mengalami kenaikan pada beban 50 *user*, tetapi nilainya masih setara dengan beban 10 *user*, dan pada beban 100 *user* mengalami kenaikan yang cukup tinggi. Sehingga, sistem dengan beban 100 *user* pada masing-masing alamat API yang diuji, tidak dapat berjalan dengan baik. Secara keseluruhan, sistem monitoring suhu tubuh, detak jantung, saturasi oksigen, dan tekanan darah berbasis *internet of things* berjalan dengan cukup baik.

Referensi

- [1] Efendi, Y. (2018). *Internet of things* (Iot) Sistem Pengendalian Lampu Menggunakan Raspberry Pi Berbasis *Mobile*. *Jurnal Ilmiah Ilmu Komputer*, 4(2), 21–27. <https://doi.org/10.35329/jiik.v4i2.41>
 - [2] Supriyatna, H. A., & Away, Y. (2019). Desain Sistem *Internet of things* (Iot) Untuk Pemantauan Dan Prediksi Gejala Serangan Jantung. *Jurnal Karya Ilmiah Teknik Elektro*, 4(1), 31–39.
 - [3] Saputra, G. Y., Afrizal, A. D., Mahfud, F. K. R., Pribadi, F. A., & Pamungkas, F. J. (2017). Penerapan Protokol MQTT Pada Teknologi Wan (Studi Kasus Sistem Parkir Universitas Brawijaya). *Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer*, 12(2), 69. <https://doi.org/10.30872/jim.v12i2.653>
 - [4] Surahman dan E. B. Setiawan, "Aplikasi *Mobile* Driver Online Berbasis Android Untuk Perusahaan Rental Kendaraan," *J. Ultim. InfoSys*, vol. 8, no. 1, hal. 35–42, 2017, doi: 10.31937/si.v8i1.554.
 - [5] Sontana, I., Rahmatulloh, A., & Rachman, A. N. (2019). Application Programming Interface Google Picker Sebagai Penyimpanan Data Sistem Informasi Arsip Berbasis Cloud. *Jurnal Nasional Teknologi Dan Sistem Informasi*, 5(1), 25–32. <https://doi.org/10.25077/teknosi.v5i1.2019.25-32>
 - [6] <https://sis.binus.ac.id/2019/05/13/stress-testing/>, diakses tanggal 5 Desember 2021
-