

# Implementasi Elastic Stack Pada Sistem Pendeteksi Tingkat Stres Menggunakan Sensor GSR dan DS18B20 Berbasis Raspberry Pi

Fatihatul Puti Sabrina<sup>1\*</sup>, Budi Bayu Murti<sup>2</sup>

<sup>1</sup>Program Studi Teknologi Rekayasa Internet, Universitas Gadjah Mada, Sleman, DI Yogyakarta

<sup>2</sup>Program Studi Teknik Telekomunikasi, Universitas Gadjah Mada, Sleman, DI Yogyakarta

Email: <sup>1\*</sup>fatihatunputi@mail.ugm.ac.id, <sup>2</sup>budibm@ugm.ac.id

(Naskah masuk: 25 Okt 2021, direvisi: 21 Feb 2022, diterima: 02 Mar 2022)

## Abstrak

Kesehatan tubuh mencakup kesehatan fisik dan mental. Salah satu faktor penentu kesehatan mental adalah stres. Selama ini, telah tersedia alat pendeteksi stres dengan memanfaatkan indikator fisiologis akibat reaksi yang muncul dari *sympathetic nervous system*, namun alat cenderung mahal dan masih bekerja secara terpisah. Dalam penelitian ini, dibuat dua buah prototipe pendeteksi tingkat stres menggunakan *Galvanic Skin Response*, DS18B20, dan *Raspberry Pi*. Skenario sistem dirancang untuk dua orang pasien dari dua rumah sakit berbeda yang ditangani oleh satu orang tenaga medis. Untuk memastikan reliabilitas jaringan dalam transmisi data, mempertimbangkan pengolahan *database* dan visualisasi pengguna, implementasi *Elastic Stack* dilakukan pada sistem. Data dikirimkan dari *Raspberry Pi* sebagai *client* menggunakan *Beat* dan ditampung ke dalam *Logstash* sebelum dimasukkan ke dalam *database* (*Elasticsearch*). Hasil pengolahan data divisualisasikan menggunakan *Kibana dashboard*. Dalam penelitian, kalibrasi sensor GSR menunjukkan *percentage difference* sebesar 0,79% dan sensor DS18B20 sebesar 0,095%. Rata-rata *delay* dalam proses transmisi data berlangsung sekitar 3-4 detik. Hal ini terjadi karena *Filebeat* akan menyesuaikan kecepatan pengiriman data agar tidak membebani *server*. Mekanisme *harvester* dan *prospector* pada *Filebeat* juga memastikan semua data terkirim dan tersimpan dalam *registry file*, sehingga sistem akan melakukan pengiriman kembali sekalipun *server down*. Secara keseluruhan, hasil pengujian *QoS* menggunakan standar TIPHON menunjukkan bahwa transmisi data dari *Beat* menuju *Logstash* berkategori memuaskan.

**Kata Kunci:** Stres, *Elastic Stack*, GSR, DS18B20, *Raspberry Pi*, *Quality of Services (QoS)*

## *Implementations of Elastic Stack on Raspberry Pi Based Stress Level Monitoring System Using GSR and DS18B20 Sensor*

### *Abstract*

*The human body health includes physical and mental health. One of the factors that determine our mental health is stress. By this time, stress detection tool is available by utilizing physiological indicators due to stress reactions arising from the sympathetic nervous system, but the tool tends to be expensive and still works separately. In this study, two prototypes of stress level detection were created using Galvanic Skin Response, DS18B20, and Raspberry Pi. The system scenario is designed for two patients from two different hospitals handled by one medical personnel. The system employs elastic stack implementation to provide network dependability in data transfer while taking database processing and user visualizations into account. Data is sent from the Raspberry Pi as a client using Beat and accommodated into Logstash before entering the database (Elasticsearch). The results of the processed data are visualized using Kibana dashboard. In this study, the GSR sensor calibration showed a percentage difference of 0.79% and the DS18B20 sensor 0.095%. The average delay in the data transmission process lasts about 3-4 seconds. It is happening because Filebeat will adjust the speed of data transmission so as not to burden the server. The harvester and prospector mechanism in Filebeat also ensures that all data is sent and stored in the registry file, so the system will send it back even if the server is down. Overall, the QoS test results using the TIPHON standard show that the data transmission from Beat to Logstash is categorized as satisfactory*

**Keywords:** *Stress, Elastic Stack, GSR, DS18B20, Raspberry Pi, Quality of Services (QoS)*

**I. PENDAHULUAN**

Pembahasan mengenai kesehatan mental yang masih tabu di Indonesia menyebabkan seorang individu cenderung merepresentasikan apa yang mereka rasakan dari sesuatu yang dapat diamati secara riil, seperti perubahan fisik. Kebiasaan tersebut mengakibatkan banyak orang sulit menyadari bahwa mereka terkena stres. Dewasa ini, alat untuk *screening* kondisi stres secara fisiologis sudah beredar di pasaran, tetapi alat masih bekerja secara terpisah dan mengeluarkan biaya yang mahal untuk digunakan. Dari sisi *user*, keterbatasan waktu juga proses administrasi yang panjang menyebabkan seseorang malas untuk pergi ke tenaga ahli. Selain itu, terdapat ketimpangan dari ketersediaan tenaga ahli dalam bidang kesehatan jiwa di Indonesia. Sangat umum ditemukan jika satu psikiater bekerja dalam dua hingga tiga rumah sakit yang berbeda. Integrasi sistem dengan *IoT*, menjadi solusi untuk mengatasi kesenjangan yang terjadi.

Perubahan fisiologis dari *sympathetic nervous system* yang aktif ketika seseorang stres, menimbulkan reaksi psikologis pada kulit dan perubahan suhu tubuh dapat diamati dengan sensor. Pengukuran kondisi stres dapat dilakukan dari parameter suhu tubuh, degup jantung, GSR, dan tekanan darah. Penggunaan DS18B20 dan GSR dalam mengukur tingkat stres pernah dilakukan dengan menggunakan metode eksperimental dan observasi. Integrasi kedua sensor menggunakan mikrokontroler *Arduino* dan ESP8266 menghasilkan hasil pembacaan yang baik [1]. Metode penggabungan sensor untuk mendeteksi reaksi fisiologis dari stres perlu dilakukan mengingat saat ini pengecekan kondisi stres masih jarang dan sistem bekerja secara terpisah. *Screening* kondisi stres yang paling banyak ditemui adalah pengisian kuesioner psikologi [2].

Selain reaksi dari perubahan suhu tubuh, salah satu parameter yang kerap digunakan dalam pengukuran tingkat stres adalah perubahan resistansi pada kulit. GSR sensor bekerja dengan dipasangkan pada jari telunjuk dan jari tengah, dan akan mengukur konduktansi dari kulit manusia. Uji coba GSR sensor dalam pemasangan dalam kondisi duduk, berjalan dan berdiri, menunjukkan hasil bahwa posisi tubuh seseorang mempengaruhi hasil *output* dari pembacaan sensor. Kondisi duduk dalam pengukuran mendapatkan pembacaan yang lebih stabil dari kondisi lainnya [3]. Selain itu, informasi seperti resistansi kulit juga berguna dalam dunia medis sebagai informasi data *biodemik*. Hasil pembacaan ini dapat digunakan sebagai rekam data yang dapat diakses oleh rumah sakit yang membutuhkan. Skema penelitian ini pernah dilakukan dengan mendeteksi *output* sensor GSR dan ECG dalam bentuk gelombang amplitudo, kemudian menyimpan *output* dalam bentuk PNG, lalu disimpan [4]. Bila dalam data *biodemik* hasil pengukuran akan langsung diperoleh, dalam pengukuran tingkat stres, data *biodemik* perlu diklasifikasi menjadi beberapa tingkatan yaitu rileks, tenang, cemas, dan stres. Dimana kondisi stres terjadi ketika suhu tubuh berada pada nilai kurang dari 33 dan konduktansi kulit memiliki nilai lebih dari 6 [5].

Sistem yang dikembangkan tidak bisa hanya terfokus pada perangkat keras. Pengelolaan, penyimpanan, hingga

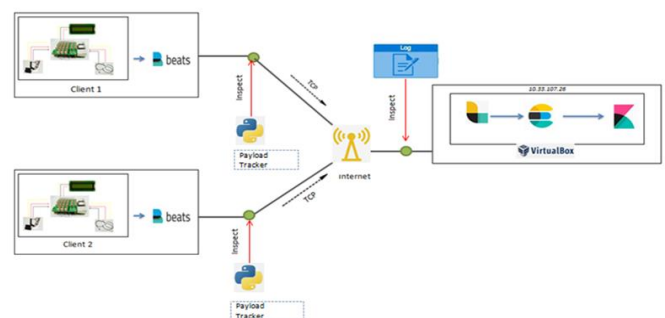
visualisasi data *user* diperlukan. Integrasi dari sistem kesehatan yang dilengkapi dengan *monitoring* dapat membantu pasien, dokter, perawat, juga staf medis dalam menyimpan rekam data pasien dalam bentuk *Electronic Health Record* (EHR) yang dapat di-*maintained* dari setiap rumah sakit [6].

Pengolahan dan penyimpanan data yang baik akan menunjang sistem yang dibuat. Jaringan yang bermasalah dapat menyebabkan sistem tidak *reliable*. Padahal setiap data yang dikirim perlu dipastikan sampai dengan baik di sisi *end user*. Pengelolaan *database* struktural juga akan membaca data secara berurut dari awal hingga akhir, sehingga memerlukan waktu yang cukup lama bila digunakan dalam skala besar. Untuk memenuhi kebutuhan pasien dan psikiater, visualisasi dan rekam data pasien diperlukan sebagai bahan pertimbangan atau catatan riwayat kondisi pasien. Dalam memenuhi kebutuhan dari rancangan yang akan dibuat, implementasi *ELK framework* mencakup *Logstash*, *Elasticsearch*, *Kibana*, dan *Beat* dilakukan. *Filebeat* yang *diset-up* dalam *Raspberry Pi* akan diintegrasikan dengan DS18B20 dan *Galvanic Skin Response* (GSR). *Elastic Stack* dapat digunakan dalam berbagai data kolektor dan memiliki *database* dengan performa yang baik, *Elastic Stack* juga dapat digunakan sebagai GUI dalam visualisasi data oleh *developer* tanpa perlu melakukan *custom* yang rumit.[7]

Berdasarkan problematika yang terjadi, juga merujuk dari tinjauan pustaka yang dilakukan pada penelitian sebelumnya, implementasi *Elastic Stack* dilakukan pada sistem pendeteksi tingkat stres dengan menggunakan sensor DS18B20 dan *Galvanic skin response*. Untuk memastikan kualitas transmisi data antara pengguna menuju *server*, dilakukan pengujian *quality of services* menggunakan parameter *delay*, *packet loss* dan *packet delivery*.

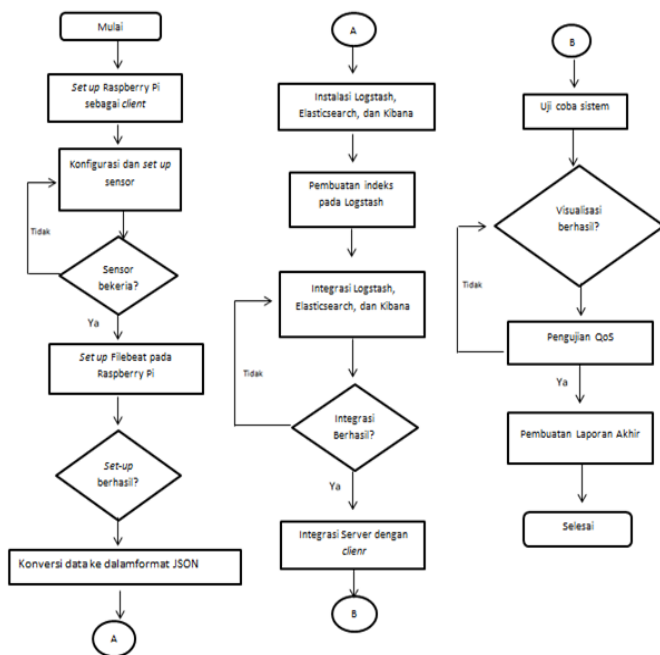
**II. METODE PENELITIAN**

Rancangan sistem pendeteksi tingkat stres yang dibangun menggunakan *Elastic Stack* memiliki empat komponen utama, yaitu prototipe sistem pendeteksi tingkat stres, integrasi *client server*, *dashboard monitoring*, dan pengembangan *framework*. Implementasi *Elastic Stack* pada *user* hanya mencakup instalasi *Beat* sebagai data *forwarder* pada setiap prototipe. Sedangkan *Logstash*, *Elasticsearch*, dan *Kibana* dibangun dalam sebuah mesin *virtual Ubuntu server*. Gambar 1 menunjukkan implementasi sistem:



Gambar 1. Implementasi Sistem

Dua buah prototipe diasumsikan sebagai rumah sakit konseling 1 dan rumah sakit konseling 2 yang ditulis sebagai *client 1* dan *client 2*. Alat ukur stres membaca data dari parameter suhu tubuh dari sensor DS18B20 dan konduktansi kulit dari sensor GSR. Hasil pembacaan akan dikirim dengan *Filebeat* – salah satu jenis *Beat* – menggunakan *Python* sebagai *payload tracker*. Proses ACK berlangsung dalam komunikasi *Beats* dan *Logstash* selama proses transmisi data. Hal ini terjadi karena sistem berjalan dengan protokol TCP. Data dari rumah sakit dalam bentuk *log* di-*parsing* oleh *Logstash server* dan disimpan dengan nama indeks ‘data kesehatan’ di *Elasticsearch*. Indeks ini akan diambil oleh *Kibana* sebagai bahan visualisasi untuk tenaga medis dan pasien. Alur kerja sistem pendeteksi tingkat stres dijelaskan pada Gambar 2 berikut:



Gambar 2. Alur Sistem

Dari Gambar 2, dapat dilihat bahwa sistem dimulai dari perancangan perangkat keras. *Galvanic skin response* yang diintegrasikan dengan sensor suhu DS18B20 di-*set up* dalam *Raspberry Pi*, setiap pengukuran yang berhasil akan disimpan dalam format JSON. *Filebeat* – yang bertindak sebagai data *forwarder* mengirimkan setiap penambahan data baru menuju *Logstash server*. Pada *Logstash*, pembuatan indeks dengan nama “data kesehatan” memberikan inisialisasi pada *Elasticsearch* bahwa hasil pembacaan dari masing-masing *client* disimpan dalam indeks bernama data kesehatan. Indeks ini akan berguna dalam pembuatan visualisasi untuk *user*. Ketika *Filebeat client 1* dan *Filebeat client 2* sudah terhubung dengan *server*, maka setiap hasil pengukuran data pasien akan disimpan dalam *database* dan dapat digunakan lebih lanjut untuk kepentingan visualisasi. Pengujian *QoS* dengan parameter *delay*, *packet loss*, dan *packet delivery* dilakukan dalam proses transmisi data dari *Filebeat* menuju *Logstash*.

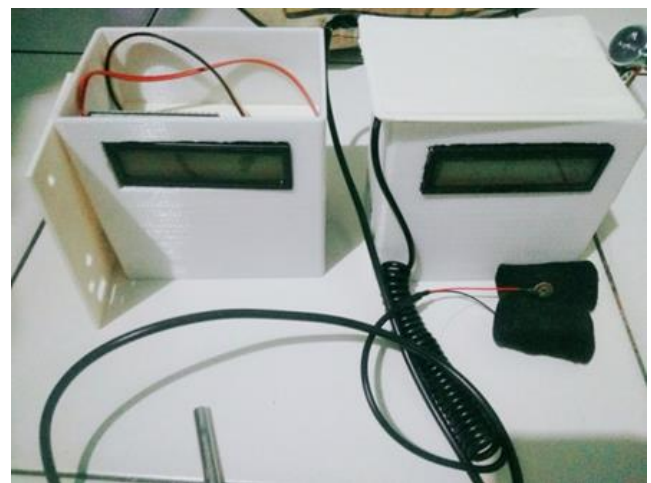
**A. Prototipe Sistem Pendeteksi Stres**

Komponen penting yang pertama kali dibuat adalah prototipe sistem pendeteksi tingkat stres. *Galvanic skin response* dan suhu DS18B20 melakukan pengukuran kondisi masing-masing pasien. Sebelum digunakan masing-masing sensor dikalibrasi dengan metode eror sistematis untuk memastikan bahwa hasil pembacaan baik. Kalibrasi GSR dilakukan dengan membandingkan nilai *voltage* dengan nilai pembacaan *avometer*, sementara DS18B20 dilakukan dengan termometer digital. Untuk mengetahui tingkat stres seseorang, hasil pembacaan akan disesuaikan dengan penggolongan tingkat stres. Dalam penelitian ini, parameter tingkat stres merujuk dari penelitian Suwarto Edi (2012). Tabel 1 merupakan parameter tingkat stres yang digunakan dalam penelitian.

Tabel 1. Parameter Tingkat Stres (Suwarto Edi)

Kondisi	GSR (Siemens)	T (°C)
Rileks	<2	35-37
Tenang	2-4	34-36
Cemas	4-6	33-35
Stres	<6	<33

Masing-masing sensor akan membaca sebanyak dua puluh data dan mengambil nilai rata-rata dalam setiap pengukuran. Hasil pengukuran akan dimasukkan ke dalam kategori tingkat stres. Bila pembacaan program Python sesuai dengan parameter yang dibuat, maka *output* akan disimpan dalam format JSON *file*. Hal ini dilakukan karena *Beat* akan mengirim data dalam format JSON. Dalam pengukuran, sistem menggunakan LCD 16x4 sebagai *interface* dari sisi *hardware*, sehingga pasien mengetahui bila sistem tengah berjalan. Gambar 3 merupakan rancangan dua prototipe pendeteksi tingkat stres:

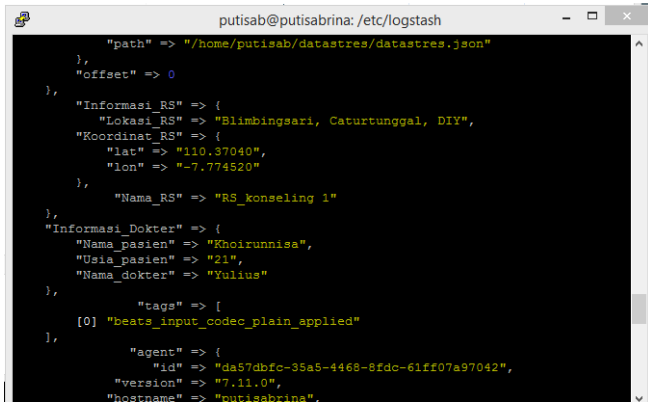


Gambar 3. Rancangan Prototipe Keseluruhan

**B. Integrasi Client Server**

*Beat* yang dikonfigurasi pada prototipe perlu diintegrasikan dengan *Logstash* sehingga dapat mengirim data pada alamat *server*. Dalam konfigurasi *filebeat.yml*, dilakukan penambahan *IP address* dan *port server* (secara default *Logstash* berjalan

pada alamat 5044). Untuk menentukan *file* mana yang akan dikirim oleh *Beat*, diberikan alamat direktori dari data pengukuran tingkat stres yang telah disimpan dalam format JSON. Konfigurasi ini akan membuat *Beat* berjalan dan menyimpan registrasi *file* sehingga setiap data akan dikirim oleh *Filebeat* dan sampai ke *Logstash server*. Gambar 4 menunjukkan integrasi *Filebeat* dan *Logstash*:

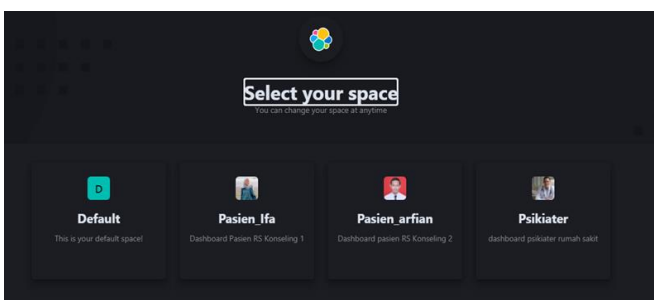


Gambar 4. Integrasi *Filebeat* dan *Logstash*

Tampilan pada sisi kiri gambar menunjukkan hasil *parsing Logstash* dari data yang dikirim *client*, sedangkan sebelah kanan merupakan pengiriman yang dilakukan *client*. Dari sisi *server*, *log* memberikan informasi dari *client* mana data diterima, *log* juga memuat waktu masuknya setiap data yang dikirim *client*. Dalam proses transmisi data dari masing-masing *client* menuju *server* dilakukan pengujian *quality of services* dengan mengukur jeda waktu dari pengiriman hingga data diterima *server*, pengecekan jumlah paket yang hilang, dan paket yang diterima.

**C. Dashboard Monitoring**

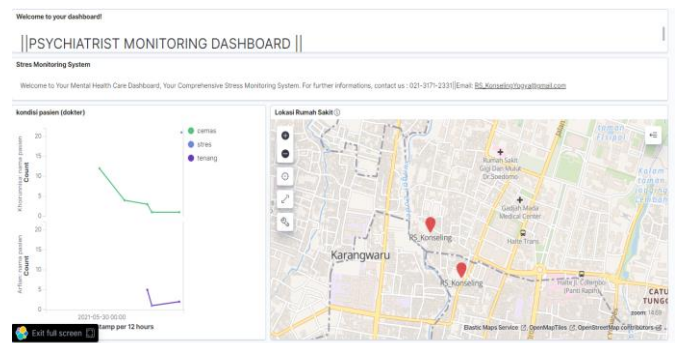
*User interface* menjadi salah satu komponen yang perlu diperhatikan. Pembuatan *dashboard* dengan *Kibana* perlu disesuaikan dengan kebutuhan pengguna. Dalam penelitian, *dashboard* dibagi menjadi tiga *space*. *Dashboard* untuk psikiater, untuk pasien 1, dan untuk pasien 2. Pengaturan *console security* pada *Kibana* dilakukan untuk menambahkan *username* dan *password* untuk masing-masing *user*. Untuk memastikan bahwa tiap *user* memiliki hak akses sesuai proporsinya, konfigurasi *feature visibility Kibana* diterapkan. Pembagian *space Kibana* tertera pada Gambar 5 *Dashboard Monitoring Kibana*:



Gambar 5. *Dashboard Monitoring Kibana*

**D. Pengembangan Framework**

Implementasi dari *Elastic Stack* dilakukan pada empat pilar utama ELK yaitu *Beat* yang di-*setup* pada prototipe pasien, *Logstash* sebagai *pipeline server* yang menerima semua data dari rumah sakit yang masuk, *Elasticsearch* sebagai *database* rumah sakit, *Kibana* sebagai *dashboard monitoring*. Dalam *use case* yang dibuat, prototipe akan digunakan oleh dua orang pasien yang berbeda dan rumah sakit yang berbeda, namun ditangani oleh satu orang pasien. Hal ini sengaja dibuat dari keadaan yang sering ditemukan di masyarakat dimana satu orang tenaga medis sering menangani lebih dari satu orang pasien dalam rumah sakit yang berbeda. Untuk mempermudah kinerja dari tenaga medis, visualisasi akan menampilkan informasi dari pasien dalam bentuk *maps* (memberikan data nama pasien, nama rumah sakit asal, dan alamat rumah sakit asal), visualisasi tabel, dan bar. Gambar 6 menunjukkan contoh visualisasi koordinat rumah sakit:



Gambar 6. Visualisasi Koordinat Rumah Sakit

**III. HASIL DAN PEMBAHASAN**

**A. Pengambilan Data Pasien**

Untuk memastikan akurasi pembacaan sistem, kalibrasi masing-masing sensor perlu dipastikan mendapatkan hasil yang baik. Hasil perbandingan sensor DS18B20 dan termometer digital memperoleh nilai *percentage difference* sebesar 0,79%. Sedangkan *Galvanic skin response* yang *output voltage*-nya dibandingkan dengan *Avometer* digital, memperoleh hasil *percentage difference* sebesar 0.095%. Dalam penelitian ini, penulis menggunakan dua responden. Tabel 2 terlampir informasi pasien

Tabel 2. Informasi Pasien

User	Jenis Kelamin	Usia	Profesi
Pasien 1	Wanita	21	Mahasiswi
Pasien 2	Pria	22	Mahasiswa

Kedua responden akan melakukan pengambilan data masing-masing selama tiga hari dalam beberapa kali percobaan. Hasil pembacaan kondisi stres pasien tertera pada Tabel 3:



Tabel 3. Hasil Pengukuran Kondisi Pasien

User	Hari	Suhu	GSR	Status
Pasien 1	H-1	34,381	5,359	Cemas
Pasien 1	H-1	34,124	5,545	Cemas
Pasien 1	H-2	35,721	2,290	Tenang
Pasien 1	H-2	35,856	2,565	Tenang
Pasien 1	H-3	35,968	2,408	Tenang
Pasien 1	H-3	35,462	2,171	Tenang
Pasien 1	H-3	35,749	2,027	Tenang
Pasien 2	H-1	36,043	1,658	Rileks
Pasien 2	H-1	35,606	2,257	Tenang
Pasien 2	H-1	35,559	2,348	Tenang
Pasien 2	H-2	35,568	2,453	Tenang
Pasien 2	H-2	35,252	2,426	Tenang
Pasien 2	H-3	36,293	1,634	Rileks
Pasien 2	H-3	35,356	2,496	Tenang
Pasien 2	H-3	35,324	2,510	Tenang

Pengukuran kondisi stres masing-masing pasien dilakukan pada tiga hari yang berbeda. Berdasarkan hasil pembacaan sistem, kondisi stres tertinggi yang dihasilkan adalah “cemas” dan belum ada pasien yang masuk dalam kategori stres. Hasil pengukuran ini dapat dijadikan bahan pertimbangan bagi tenaga ahli selain dari *screening* psikologi seperti DAAS, PHQ-9, dan lain-lain. Durasi terjadinya stres dan intensitas munculnya perubahan fisiologis yang muncul akibat reaksi SNS, dapat membantu dalam menentukan apakah stres yang dialami perlu dilakukan penanganan lebih lanjut atau tidak.

### B. Fungsionalitas Keseluruhan Sistem

Untuk memastikan keseluruhan dari sistem baik dari prototipe hingga *dashboard monitoring* berfungsi dengan baik, dilakukan pengujian untuk memastikan sistem mengambil data pasien dan menampilkannya hingga *dashboard*. Gambar 7 menunjukkan tampilan prototipe saat digunakan pasien.



Gambar 7. Penggunaan Prototipe Oleh Pasien

Ketika *Filebeat* dari masing-masing prototipe telah berhasil mengirim data menuju *Logstash server*, maka data tersebut akan secara otomatis tersimpan dalam indeks dan ditampilkan *Kibana*. Untuk memaksimalkan fungsionalitas sistem, hasil data dapat diunduh dan disimpan oleh psikiater dan pengguna sebagai data rekam medis. Rekap hasil pengunduhan data pasien dapat diunduh dalam bentuk CSV *file*.

### C. Reliabilitas Jaringan

Tersampainya setiap data dari *client* menuju *server* merupakan hal yang penting. Terutama bila suatu ketika *server* dalam kondisi tidak stabil atau bahkan *down*. Untuk mengetahui reliabilitas jaringan dilakukan pengujian *packet loss* dan *packet delivery* yang dilakukan dengan pengiriman *payload* data dari satu *client*, dan pengiriman *payload* data dari dua *client* secara bersamaan. Dalam pengujian pengiriman 40 data dari 1 *client* diperoleh hasil bahwa semua data tersampaikan dan paket data yang hilang satupun. Hasil yang sama juga diperoleh dalam pengiriman data secara bersamaan dari dua *client*. Tidak ditemukan paket yang hilang selama proses transmisi data. Paket yang diterima sejumlah 40 hits (sesuai dengan jumlah data yang dikirim diawal). Hal ini menunjukkan bahwa reliabilitas jaringan cukup baik. Selain itu, pengiriman *payload* dari *client 1* dan *client 2* menunjukkan hal yang sama nilai *packet loss* 0% dan *packet delivery* 100%. Total paket diterima terlihat pada Gambar 8, total *hits Kibana dashboard*:

Field	Value
Hits	40
Hits (total)	40
Index pattern	datakesehatan
Index pattern ID	e5288da0-c057-11eb-ab1c-db403749cea0

Gambar 8. Total Hits Kibana Dashboard

Bila dilihat dari proses transmisi data, TCP berjalan dengan memastikan setiap data tersampaikan dengan baik sampai ke sisi penerima. Selain itu, dilakukan pengujian pengiriman *payload* saat *server* dimatikan. Secara otomatis, *Filebeat* memberikan informasi pada *user* bahwa *Logstash server unreachable*, informasi ini didapatkan karena terjadi proses ACK dari *server*. Ketika *server* dinyalakan, *Filebeat* akan menerima konfirmasi, dan kembali mengirimkan data yang sebelumnya disimpan dalam register *file* secara otomatis [8]. Hal ini menunjukkan bahwa *beat* memiliki *back pressure handling* yang baik. *Filebeat* menyimpan setiap registrasi data. *Prospector* memberikan *input* lokasi direktori, dan *harvester* bertugas membaca dan mendeteksi setiap terjadi perubahan ukuran *file*. Dua mekanisme ini akan memastikan bahwa setiap data telah dikirim.

### D. Waktu Transmisi Data

Pengujian *delay* dilakukan selama proses transmisi data dari *Filebeat* menuju *Logstash*. Pengujian dilakukan dalam

dua skenario, yaitu pengiriman data secara seri dari satu *client*. Dan pengiriman data dari kedua *client* secara bersamaan. Pengiriman *delay* dari satu *client* dapat dilihat pada Tabel 4, *delay* pada pengiriman satu *client*:

Tabel 4. *Delay* pada pengiriman satu *client*

Data Ke- n	Delay Pada Percobaan ke-n			
	1	2	3	4
1	9.337	1.566	3.865	1.939
2	9.324	3.633	6.980	2.079
3	1.266	6.619	1.281	1.663
4	1.018	2.710	3.504	2.390
5	1.972	4.114	7.849	5.108
6	8.314	8.201	7.516	5.884
7	7.843	6.823	9.186	0.665
8	1.300	4.008	2.481	3.188
9	8.371	4.110	6.004	6.001
10	6.910	1.400	4.832	2.734
<b>Rata-rata Delay (ms)</b>	5.448,5	4.318,4	4.963,3	3.165,1

Sedangkan hasil *delay* dalam pengiriman *payload* dari dua *client* dapat dilihat pada Tabel 5, *delay* pada pengiriman dua *client*:

Tabel 5. *Delay* Pada Pengiriman Dua *Client*

Percobaan Ke- n	Delay Data Pada Dua Client	
	Client 1	Client 2
1	521	4.810
2	2.454	6.673
3	6.825	1.098
4	389	7.521
5	8.189	2.392
6	941	5.010
7	3.719	7.980
8	2.902	7.085
9	1.725	6.012
10	6.942	1.237
11	6.221	9.164
12	8.560	1.868
13	7.868	1.892
14	5.042	8.936
15	1.392	4.778
16	1.107	4.374
17	8.549	1.837
18	8.002	4.803
19	7.535	672
20	840	4.257
<b>Rata – Rata Delay (ms)</b>	4.486,1	4.619,9

Besarnya nilai *delay* yang dihasilkan dapat disebabkan oleh beberapa faktor, di antaranya mekanisme bawaan dari sistem, atau kondisi jaringan yang bermasalah. Pengujian *bandwidth* dan *latency* jaringan dilakukan dari *client* menuju *server*. Hasil yang diperoleh dari pengujian memiliki nilai bagus, *bandwidth* yang diperoleh berkisar di atas 10Mbps sedangkan nilai *latency* yang diperoleh berada dibawah 150 ms.

Bila dilihat dari mekanisme *Filebeat*, *Beat* yang dikembangkan oleh ELK *framework* sebagai data *forwarder* dengan tidak membebani *Pipeline* akan menyesuaikan proses pengiriman paket agar tidak membuat *server* sibuk [9]. Hal ini dapat dilihat nilai *delay* dalam pengiriman dua *client*. Ketika *payload* dari *client* satu masuk, terdapat jeda dalam proses *load* pada *filebeat* di prototipe ke dua dalam mengirim data. Ketika data telah di-*load*, *Logstash* bekerja dengan cepat dalam mem-*parsing* data. Mekanisme bawaan dari *Filebeat* secara tidak langsung mempengaruhi waktu dalam pengiriman data dari *client* menuju *server*.

#### IV.KESIMPULAN

Sistem yang dibuat dapat bekerja dengan baik, mulai dari pengambilan data menggunakan prototipe, integrasi *client-server*, visualisasi data, dan penyimpanan data sebagai bagian dari rekam rekam medis. Sensor GSR memiliki *percentage difference* sebesar 0,095%, sedangkan DS18B20 memiliki *percentage difference* sebesar 0,79%. Dari sisi transmisi data, reliabilitas jaringan dalam proses pengiriman data untuk *client-server* baik, *Filebeat* menyimpan setiap registrasi *file* untuk memastikan semua data telah terkirim, hal ini terjadi karena kinerja dari dua komponen utama *Beat* yaitu *prospector* dan *harvester*. Nilai *packet loss* dari penelitian ini sebesar 0%, sedangkan *packet delivery* sebesar 100%. Ketika *server* mati, *Filebeat* kembali mengirim ulang data secara otomatis sampai *server* kembali hidup. Rata-rata nilai *delay* yang berada pada rentang 3-4 detik terjadi karena *Beat* secara otomatis akan mengatur kecepatan pengiriman agar tidak membebani *Logstash*. *Beat* juga akan meng-enkripsi setiap data yang dikirim menuju *server*, sehingga implementasi *Elastic Stack* dapat digunakan bagi *developer IoT* pemula yang memiliki keterbatasan dibidang keamanan jaringan.

#### REFERENSI

- [1] K. Fadhilah, A. Stefanus, and D. Fauzandhiiya, "Perangkat Pemantau Kesehatan Mental Berbasis IOT," *Industrial Research Workshop and National Seminar*, vol. 9, pp. 840–847, 2018.
- [2] G. J. Hernando, D. S. Ginting, and F. Syahbarudin, "Perangkat Asisten Dokter Untuk Penyakit Stres," *Industrial Research Workshop and National Seminar*, vol. 9, 2018.
- [3] Y. L. Wee, "Development of Galvanic Skin Response Sensor System to Measure Mental Stress," *Dissertation submitted in partial fulfilment of the requirements for the*

- Bachelor of Engineering (Hons) (Electrical & Electronic)*, 2014.
- [4] M. M. Idris, "Rancang Bangun Sistem Pengumpulan Data Biomedik," *Tugas Akhir Universitas Negeri Makassar*, 2019.
- [5] Y. Estrada, "Alat Pengukur Tingkat Kestressan Manusia," *Tugas Akhir Program Studi Teknik Elektro Universitas Negeri Semarang*, 2019.
- [6] D. Mathuvanthi , V. Suresh, and C. Pradeep, "IoT Powered Wearable to Assist Individuals Facing Depression Symptoms," *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 1, 2019.
- [7] M. Bajer, "Building an IoT Data Hub with Elasticsearch, Logstash, and Kibana," *International Conference on Future Internet of Things and Cloud Workshops 5th*, pp. 64–66, 2019.
- [8] ELK Team. (n.d.), "How Filebeat work?," [www.elastic.co](http://www.elastic.co). [Online]. Available: <https://www.elastic.co/guide/en/beats/filebeat/current/how-filebeat-works.html>. [Accessed: 28-Jun-2021].
- [9] Elastic Stack. (n.d.), "Elastic Stack : Elasticsearch, Beats, Kibana & Logstash," [www.elastic.co](http://www.elastic.co). [Online]. Available: <https://www.elastic.co/elastic-stack>. [Accessed: 29-Nov-2020].