

# PROTOTYPE SISTEM *MONITORING* DAN PENGANTIAN OTOMATIS *INTRAVENOUS FLUID DROPS*

Afrizal Awlan Suryandaru<sup>1</sup>, Nyoman Putra Sastra<sup>2</sup>, I Gst A Komang Diafari Djuni<sup>3</sup>

<sup>1,2,3</sup>Program Studi Teknik Elektro, Fakultas Teknik, Universitas Udayana  
Jl. Raya Kampus Unud No. 88, Jimbaran Kabupaten Badung

[afrizal.surya11638@gmail.com](mailto:afrizal.surya11638@gmail.com)<sup>1</sup>, [putra.sastra@unud.ac.id](mailto:putra.sastra@unud.ac.id)<sup>2</sup>, [igakdiafari@ee.unud.ac.id](mailto:igakdiafari@ee.unud.ac.id)<sup>3</sup>

## ABSTRAK

Pemberian cairan intravena kepada pasien dilakukan dengan cara menyalurkan cairan ke dalam pembuluh darah vena menggunakan infus set dengan jumlah dan jangka waktu tertentu. Pasien yang diberikan cairan intravena akan selalu dipantau oleh perawat secara manual dan berkala. Pada penelitian ini, dirancang sebuah sistem *monitoring* dan penggantian otomatis *Intravenous Fluid Drops* yang bertujuan untuk mempermudah perawat dalam melakukan pemantauan dan pergantian infus secara otomatis. Sistem *monitoring* yang dirancang memanfaatkan aplikasi komunikasi Telegram yang terintegrasi dengan NodeMCU ESP8266, sedangkan sistem penggantian otomatis memanfaatkan sensor inframerah sebagai pendeteksi ketersediaan cairan infus. Berdasarkan pada hasil penelitian, sensor inframerah akan berlogika '1' jika cairan infus lebih dari 50 ml, dan sensor akan berlogika '0' jika cairan infus kurang dari sama dengan 50 ml. Respon sistem tercepat untuk mengirimkan data sensor ke Telegram Bot mencapai 01,10 detik, sedangkan respon sistem tercepat untuk menerima perintah dari Telegram Bot mencapai 05,70 detik.

**Kata Kunci** : Cairan Intravena, Telegram, NodeMCU ESP8266

## ABSTRACT

*The giving of intravenous fluids to patients is dispense fluids into the veins using an infusion set with a certain amount and period. Patients who are given intravenous fluids will constantly be monitored by a nurse manually and regularly. In this study, a monitoring system and automatic replacement of intravenous fluids drops were designed to make it easier for nurses to monitor and change infusions automatically. The monitoring system is designed to take advantage of the Telegram communication application integrated with the NodeMCU ESP8266. In contrast, the automatic replacement system utilizes an infrared sensor to detect fluid availability. Based on the research results, the infrared sensor will have a logic '1' if infusion fluid is more than 50ml, and the sensor will have a logic '0' if infusion fluid is less than 50ml. The fastest system response to send sensor data to Telegram Bot reached 01.10 seconds, while the fastest system response to receive orders from Telegram Bot reached 05.70 seconds.*

**Keywords** : *Intravenous Fluid, Telegram, NodeMCU ESP8266*  
medis infus menjadi hal yang penting dan prioritas

## 1. PENDAHULUAN

Perkembangan teknologi di dunia kedokteran mengalami kemajuan yang sangat pesat. Alat-alat kesehatan saat ini sudah dikombinasi dengan berbagai *Personal Computer (PC)* dan *Smartphone*. Kemajuan teknologi sangat dibutuhkan dalam bidang kesehatan untuk menunjang kemampuan manusia yang terbatas. Di samping telah banyaknya penggunaan teknologi dalam bidang kesehatan, ada juga yang belum, salah satunya yaitu *Intravenous Fluid Drops* (infus). Di dunia

karena fungsinya. Hal ini dapat dilihat pada setiap pasien yang mengalami sakit dan sebagian besar menggunakan infus. Sesuai dengan fungsinya, infus umumnya diberikan kepada seseorang yang mengalami kekurangan elektrolit dalam tubuh [1]. Beberapa pasien membutuhkan asupan tambahan melalui pembuluh darah vena untuk mempercepat proses asupan tambahan ke dalam tubuh.

Infus mempunyai kapasitas dalam setiap botolnya, dan apabila infus habis harus diganti dengan yang baru, infus tidak boleh sampai benar-benar habis karena dapat mengakibatkan darah naik ke selang infus [1]. Keterlambatan penggantian infus ini sering terjadi karena tidak ada durasi waktu yang pasti untuk pergantian infus. Terutama pada malam hari, ketika pasien sudah tertidur kadang terjadi petugas lupa untuk mengganti, hal ini akan berdampak fatal bagi pasien. Karena kejadian-kejadian ini maka dipandang perlu untuk mengembangkan suatu cara untuk memberikan informasi kondisi infus yang sedang digunakan dan mengotomatiskan pergantian.

Pada penelitian yang dilakukan oleh Muljodipo [2], telah berhasil merancang sebuah sistem otomatis infus pada pasien dengan memanfaatkan mikrokontroler AVR Atmega 16. Sistem otomatis yang dirancang, berfungsi untuk mengatur laju dari cairan infus dengan cara membuat katup khusus yang didesain menggunakan motor servo. Pergerakan motor servo sebagai katup didasari oleh jumlah tetesan cairan infus yang dideteksi oleh sensor fotodiode dalam kurun waktu satu menit. Seluruh data sensor hanya ditampilkan pada LCD. Penelitian yang dilakukan oleh Muflih [3] menyebutkan bahwa, memanfaatkan teknologi IoT dapat mempermudah melakukan pemantauan pada sebuah sistem. Salah satu platform yang dapat terintegrasi dengan sistem IoT adalah aplikasi *Telegram Messenger*. Selain itu, pemilihan jenis mikrokontroler termasuk dalam faktor utama untuk menerapkan sistem IoT. Jenis mikrokontroler NodeMCU ESP8266 merupakan salah satu jenis mikrokontroler yang dilengkapi dengan fitur protokol TCP/IP sehingga dapat terhubung dengan jaringan internet.

Berdasarkan kondisi yang telah dibahas dan referensi yang ada, bahwa belum adanya model sistem pergantian otomatis dan *monitoring* infus melalui aplikasi *Telegram Messenger*. Untuk itu, penelitian ini bertujuan untuk merancang perangkat yang dapat melakukan pergantian infus secara otomatis dan melakukan monitoring terhadap tingkat ketersediaan infus untuk pasien melalui aplikasi *Telegram Messenger*.

Dalam perancangan alat ini, digunakan NodeMCU ESP8266 sebagai kendali utama

untuk memproses dan mengirim data ke aplikasi *Telegram Messenger*. Sensor yang digunakan adalah sensor inframerah yang berfungsi untuk mendeteksi ketersediaan cairan infus. Data pengukuran sensor akan dikirim oleh NodeMCU ESP8266 ke aplikasi Telegram melalui jaringan internet. Selain itu, alat ini juga dapat melakukan pergantian cairan infus secara otomatis ketika infus akan habis. Penggantian infus dilakukan apabila sudah hampir habis sesuai dengan *Standard Operasional Prosedur* (SOP) tindakan keperawatan [4]. Selanjutnya perawat dapat melakukan *monitoring* penggunaan dan kebutuhan infus pasien agar tercukupi. Sebagai contoh perawat dapat memberikan tranfusi darah, membantu pemberian nutrisi parenteral, memperbaiki keseimbangan asam basa, mempertahankan atau mengganti cairan tubuh yang mengandung air, elektrolit, vitamin, protein, lemak dan kalori yang tidak dapat dipertahankan melalui oral. Dengan adanya perancangan alat ini, diharapkan mampu memberikan kemudahan kepada perawat untuk melakukan pengawasan infus secara berkala.

## 2. TINJAUAN PUSTAKA

### 2.1 Sensor Inframerah

Sensor inframerah adalah sensor yang terdiri dari 2 bagian utama, yaitu *transmitter* yang berfungsi untuk memancarkan sinar inframerah, dan *receiver* yang berfungsi sebagai penerima sinar inframerah yang telah dipancarkan oleh *transmitter*. Sensor akan mendeteksi sebuah objek jika sinar inframerah diterima oleh bagian *receiver* sensor akibat dari pantulan sebuah objek yang berada di depan sensor [5]. Sensor inframerah dapat ditunjukkan pada Gambar 1.

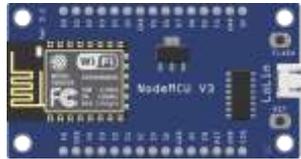


Gambar 1 Sensor Inframerah [6]

### 2.2 NodeMCU ESP8266

NodeMCU ESP8266 merupakan sebuah *board system* yang mengintegrasikan protokol TCP/IP dengan *chip* ESP8266 sebagai mikrokontroler sehingga dapat terhubung dengan jaringan

internet [7] seperti yang ditunjukkan pada Gambar 2. NodeMCU ESP8266 didukung dengan 13 pin GPIO, 10 kanal PWM, dan 1 pin analog 10 bit.



Gambar 2 NodeMCU ESP8266 [8]

### 2.3 Motor Servo

Motor servo, ditunjukkan pada Gambar 3, merupakan salah satu jenis motor yang perputaran posisi sudutnya dapat dikendalikan melalui input sinyal PWM dari mikrokontroler [9].



Gambar 3 Motor Servo [10]

### 2.4 Modul LM2596

Modul LM2596 merupakan sebuah modul elektronik yang didesain sedemikian rupa dan berfungsi sebagai *regulator step down* (penurun tegangan) yang memanfaatkan komponen IC regulator tipe LM2596 memiliki *output current* mencapai 3 A [11].

### 2.5 Telegram Messenger

Telegram merupakan platform yang menyediakan layanan pesan berbasis internet. Telegram bot merupakan fitur khusus dari aplikasi Telegram yang dapat digunakan dalam pengembangan projek IoT dan lain-lain [12].

## 3. METODE PENELITIAN

Tahapan pada penelitian ini meliputi studi literatur, perancangan *hardware*, perancangan *software*, dan pengujian, sesuai Gambar 4. Tahap pertama, dilakukan studi literatur, berasal dari berbagai sumber referensi yang berkaitan dengan penelitian ini. Tahap kedua, yakni merancang sistem *hardware* pada bagian-bagian blok sistem. Selanjutnya merancang sistem pendukung dari sistem *hardware*, yaitu perancangan sistem *software* yang berkaitan dengan pengembangan program untuk dapat

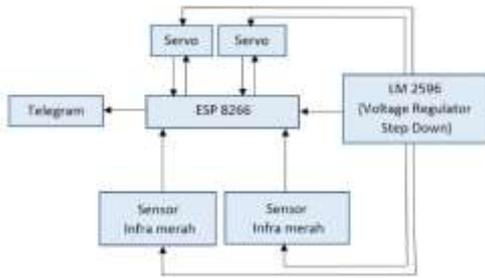
menjalankan keseluruhan sistem dengan baik. Tahapan selanjutnya adalah pengujian sistem secara keseluruhan, yang bertujuan untuk mengetahui kinerja dari sistem yang sudah dibuat. Jika terdapat *error* dalam proses pengujian, akan dilakukan perbaikan sistem sesuai dengan *error* yang terjadi.



Gambar 4 Diagram Alir Perancangan Sistem

### 3.1 Perancangan Hardware

Gambar 5 merupakan diagram blok *hardware*. Sistem dirancang menggunakan NodeMCU ESP8266 sebagai pengendali utama dan pemroses data. Terdapat dua buah sensor inframerah untuk mendeteksi ketersediaan cairan infus, dan dua buah motor servo yang berfungsi sebagai katup saluran pada masing-masing cairan infus. Data sensor yang diolah NodeMCU ESP8266 akan dikonversi menjadi sebuah informasi yang menunjukkan kondisi terkini pada ketersediaan masing-masing cairan infus. Data tersebut akan dikirim ke Telegram melalui koneksi jaringan internet.



Gambar 5 Diagram Blok Hardware

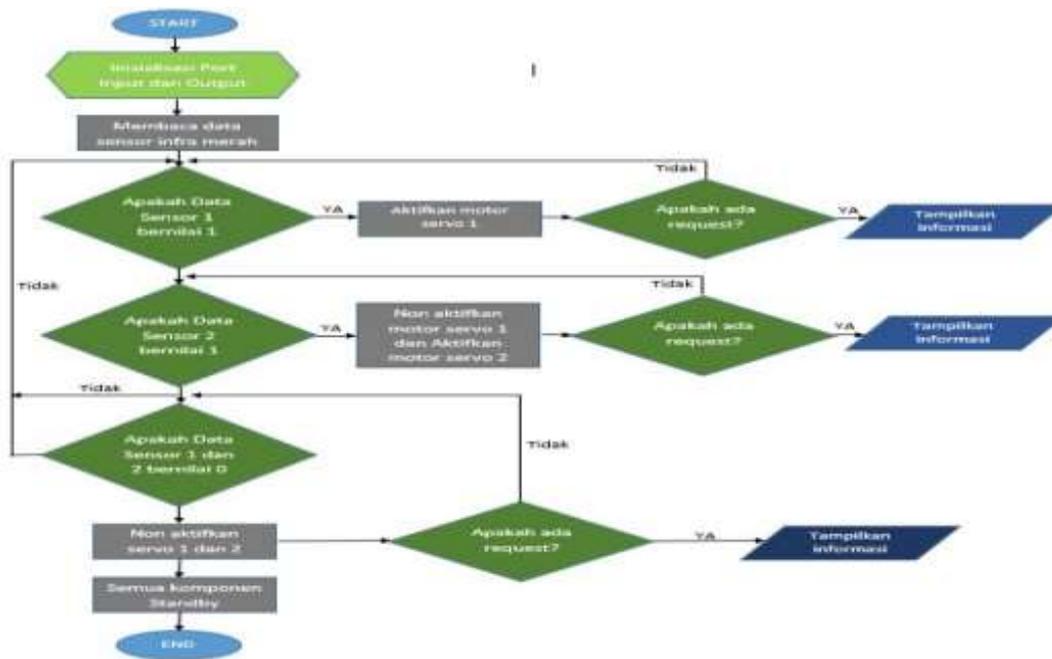
3.2 Perancangan Software

Gambar 6 merupakan diagram blok perancangan software. Inialisasi port berfungsi untuk mendeskripsikan fungsi dari masing-masing port yang akan digunakan pada NodeMCU ESP8266. Setelah inialisasi berhasil, NodeMCU ESP8266 dapat menerima data dari sensor

inframerah. Terdapat beberapa kondisi yang akan diproses oleh NodeMCU ESP8266 berdasarkan data dari sensor inframerah.

Kondisi pertama, jika sensor utama berlogika '1', maka motor servo utama akan aktif (membuka katup).

Kondisi kedua, jika sensor cadangan berlogika '1', motor servo utama dinonaktifkan (menutup katup), sedangkan motor servo cadangan akan diaktifkan (membuka katup). Pada kondisi ketiga, jika kedua sensor berlogika '0', kedua katup motor servo akan menutup.



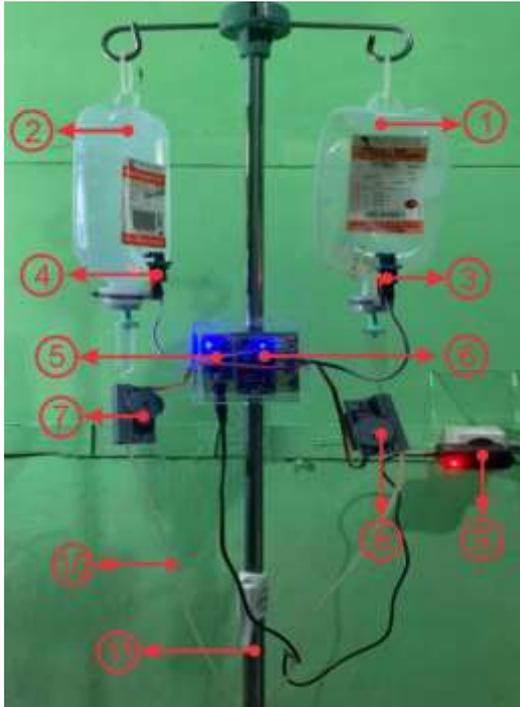
Gambar 6 Diagram Blok Perancangan Software

4. HASIL DAN PEMBAHASAN

4.1 Realisasi Perangkat Keras

Gambar 7 merupakan realisasi perangkat dari prototipe sistem monitoring dan penggantian otomatis intravenous fluid drops. Sistem kendali utama yang digunakan pada prototipe ini adalah mikrokontroler NodeMCU ESP8266. Sistem ini memanfaatkan sensor inframerah yang ditempatkan pada kedua wadah infus dan

berfungsi sebagai pendeteksi ketersediaan dari cairan infus. Pada masing-masing saluran infus terdapat katup yang telah dirancang sedemikian rupa menggunakan motor servo yang difungsikan untuk mengatur laju dari cairan infus yang akan dialirkan.



Gambar 7 Realisasi Perangkat Keras

Keterangan:

1. Infus Utama
2. Infus Cadangan
3. Sensor Inframerah pada Infus Utama
4. Sensor Inframerah pada Infus Cadangan
5. Voltage Regulator LM2596
6. NodeMCU ESP8266
7. Motor servo sebagai katup pada saluran infus cadangan
8. Motor servo sebagai katup pada saluran infus utama
9. Adaptor 9V
10. Selang Infus
11. Tiang Penyangga Infus

## 4.2 Data Hasil Pengujian

### 4.2.1 Pengujian Sensor Inframerah

Sensor inframerah digunakan untuk mendeteksi ketersediaan cairan infus pada masing-masing kantong infus. Sensor inframerah akan berlogika '1', jika ketersediaan cairan infus lebih dari 50 ml, sedangkan sensor akan berlogika '0', jika ketersediaan cairan infus kurang dari atau sama dengan 50 ml. Pengujian sensor inframerah dilakukan dengan melihat respon sensor ketika cairan infus lebih dari 50 ml, dan kurang dari atau sama dengan 50 ml. Pengujian sensor dilakukan di *software* Arduino IDE. Pengujian sensor inframerah dapat ditunjukkan pada Gambar

8, sedangkan data hasil pengujian sensor

```
COM3
22:17:20.043 -> Infus Cadangan Habis
22:17:35.503 -> Informasi Telah Terkirim
22:17:47.120 -> Infra Merah 1 : 1
22:17:47.120 -> Infra Merah 2 : 0
22:17:47.120 -> Infus Cadangan Habis
22:17:58.262 -> Informasi Telah Terkirim
22:18:10.080 -> Infra Merah 1 : 1
22:18:10.080 -> Infra Merah 2 : 0
22:18:10.080 -> Infus Cadangan Habis
22:18:21.238 -> Informasi Telah Terkirim
22:18:32.537 -> Infra Merah 1 : 1
22:18:32.537 -> Infra Merah 2 : 0
22:18:32.537 -> Infus Cadangan Habis
22:18:48.025 -> Informasi Telah Terkirim
```

inframerah ditunjukkan pada Tabel 1.

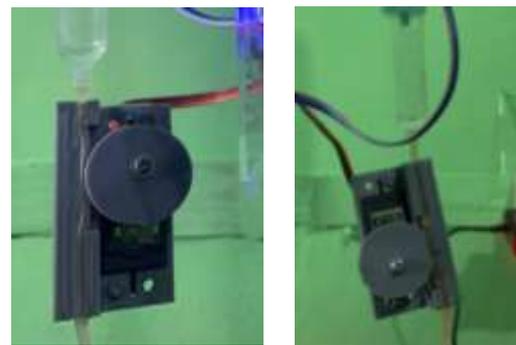
Gambar 8 Pengujian Sensor Inframerah

Tabel 1 Hasil Pengujian Sensor Inframerah

No	Inframerah 1	Inframerah 2	Keterangan
1	Bernilai 1	Bernilai 1	Semua infus terisi
2	Bernilai 1	Bernilai 0	Infus cadangan habis
3	Bernilai 0	Bernilai 1	Infus utama habis
4	Bernilai 0	Bernilai 0	Semua infus habis

### 4.2.2 Pengujian Motor Servo

Motor servo digunakan untuk membuka dan menutup katup saluran dari masing-masing cairan infus. Pengujian motor servo dilakukan dengan menguji respon dan arah putaran motor berdasarkan pada data sensor yang diterima oleh mikrokontroler. Pengujian motor servo ditunjukkan pada Gambar 9, sedangkan data hasil pengujian motor servo ditunjukkan pada Tabel 2.



(a)

(b)

Gambar 9 Pengujian Respon dan Putaran Motor Servo. (a) Motor Servo Utama Menutup 90°. (b) Motor Servo Cadangan Membuka 180°

**Tabel 2** Data Hasil Pengujian Motor Servo

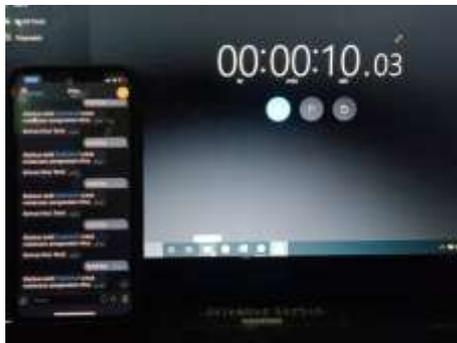
Logika Sensor		Respon Motor		Ket
Sensor Utama	Sensor Cadangan	Motor Utama	Motor Cadangan	
0	0	90° CCW	90° CCW	Baik
0	1	90° CCW	180° CCW	Baik
1	0	180° CCW	90° CCW	Baik
1	1	180° CCW	90° CCW	Baik

**4.2.3 Uji Pengiriman Data dari Telegram**

Pengujian ini dilakukan untuk menguji waktu respon sistem ketika menerima data yang dikirim dari Telegram. Data yang dikirim dari Telegram merupakan data *request*, yaitu instruksi khusus yang meminta sistem untuk memberikan informasi terkait ketersediaan cairan infus.

**a. Pengujian Respon Semua Infus Terisi**

Pengujian dilakukan dengan menghitung waktu respon sistem ketika menerima instruksi "/cekinfus" dari Bot\_Infus Telegram saat keadaan semua infus terisi. Pengujian waktu respon semua infus terisi ditunjukkan pada Gambar 10. Pengujian ini dilakukan sebanyak 10 kali percobaan. Sedangkan data hasil pengujian waktu respon sistem ketika semua infus terisi ditunjukkan pada Tabel 3.



**Gambar 10** Hasil Pengujian Waktu Respon Semua Infus Terisi

**Tabel 3** Data Hasil Pengujian Waktu Respon Semua Infus Terisi

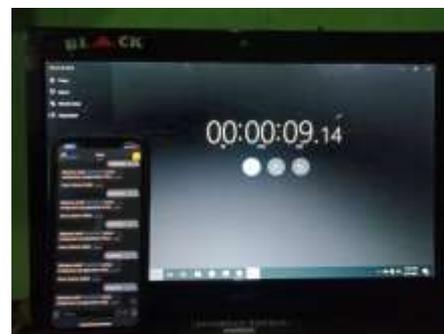
Pengujian Ke-	Data yang dikirim	Waktu Respon (Detik)
1	/cekinfus	09,81
2	/cekinfus	10,62
3	/cekinfus	10,28
4	/cekinfus	10,51
5	/cekinfus	10,95
6	/cekinfus	12,14
7	/cekinfus	12,81
8	/cekinfus	08,19
9	/cekinfus	10,80

10	/cekinfus	09,22
----	-----------	-------

Berdasarkan pengujian yang dilakukan, waktu respon tercepat terjadi pada pengujian ke-8 dengan waktu respon mencapai 08,19 detik, sedangkan waktu respon terlama terjadi pada pengujian ke-7 dengan waktu respon mencapai 12,81 detik. Waktu respon rata-rata dari 10 kali pengujian waktu pengiriman data adalah 10,53 detik. Penyebab terjadinya perbedaan waktu respon dari 10 kali pengujian, adalah karena beberapa hal yang tidak bisa dikontrol oleh sistem secara langsung, sebagai contoh, kualitas koneksi internet dan Telegram yang digunakan oleh banyak orang, sehingga proses pengiriman data mengalami *delay*.

**b. Pengujian Respon Infus Utama Habis**

Pengujian dilakukan dengan menghitung waktu respon sistem ketika menerima instruksi "/cekinfus" dari Bot\_Infus Telegram saat keadaan infus utama habis. Pengujian waktu respon infus utama habis ditunjukkan pada Gambar 11. Pengujian ini dilakukan sebanyak 10 kali percobaan. Sedangkan data hasil pengujian waktu respon sistem ketika infus utama habis ditunjukkan pada Tabel 4.



**Gambar 11** Hasil Pengujian Waktu Respon Infus Utama Habis

**Tabel 4** Data Hasil Pengujian Waktu Respon Infus Utama Habis

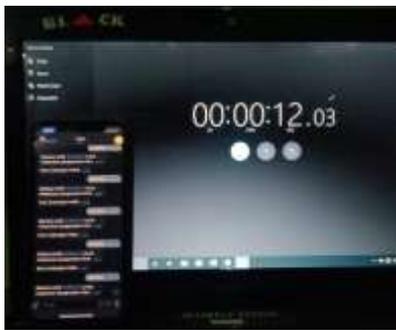
Pengujian Ke-	Data yang dikirim	Waktu Respon (Detik)
1	/cekinfus	09,00
2	/cekinfus	06,00
3	/cekinfus	09,90
4	/cekinfus	09,05
5	/cekinfus	14,00
6	/cekinfus	09,16
7	/cekinfus	10,26
8	/cekinfus	38,49

9	/cekinfus	05,70
10	/cekinfus	07,65

Berdasarkan pengujian yang dilakukan, waktu respon tercepat terjadi pada pengujian ke-9 dengan waktu respon mencapai 05,70 detik, sedangkan waktu respon terlama terjadi pada pengujian ke-8 dengan waktu respon mencapai 3,49 detik. Waktu respon rata-rata dari 10 kali pengujian waktu pengiriman data adalah 11,9 detik. Penyebab terjadinya perbedaan waktu respon dari 10 kali pengujian, adalah karena beberapa hal yang tidak bisa dikontrol oleh sistem secara langsung, sebagai contoh, kualitas koneksi internet dan Telegram yang digunakan oleh banyak orang, sehingga proses pengiriman data mengalami *delay*.

**c. Pengujian Respon Infus Cadangan Habis**

Pengujian dilakukan dengan menghitung waktu respon sistem ketika menerima instruksi "/cekinfus" dari Bot\_Infus Telegram saat keadaan infus cadangan habis. Pengujian respon infus cadangan habis ditunjukkan pada Gambar 12. Pengujian ini dilakukan sebanyak 10 kali percobaan. Sedangkan data hasil pengujian waktu respon sistem ketika infus cadangan habis ditunjukkan pada Tabel 5.



Gambar 12 Hasil Pengujian Waktu Respon Infus Cadangan Habis

Tabel 5 Data Hasil Pengujian Waktu Respon Infus Cadangan Habis

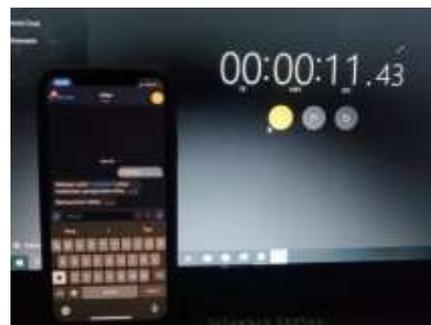
Pengujian Ke-	Data yang dikirim	Waktu Respon (Detik)
1	/cekinfus	11,41
2	/cekinfus	20,21
3	/cekinfus	10,86
4	/cekinfus	06,76
5	/cekinfus	13,83
6	/cekinfus	12,14
7	/cekinfus	18,79

8	/cekinfus	17,86
9	/cekinfus	05,95
10	/cekinfus	12,33

Berdasarkan pengujian yang dilakukan, waktu respon tercepat terjadi pada pengujian ke-9 dengan waktu respon mencapai 05,95 detik, sedangkan waktu respon terlama terjadi pada pengujian ke-2 dengan waktu respon mencapai 20,21 detik. Waktu respon rata-rata dari 10 kali pengujian waktu pengiriman data adalah 13,01 detik. Penyebab terjadinya perbedaan waktu respon dari 10 kali pengujian, adalah karena beberapa hal yang tidak bisa dikontrol oleh sistem secara langsung, sebagai contoh, kualitas koneksi internet dan Telegram yang digunakan oleh banyak orang, sehingga proses pengiriman data mengalami *delay*.

**d. Pengujian Respon Semua Infus Habis**

Pengujian dilakukan dengan menghitung waktu respon sistem ketika menerima instruksi "/cekinfus" dari Bot\_Infus Telegram saat keadaan semua infus habis. Pengujian waktu respon semua infus habis ditunjukkan pada Gambar 13. Pengujian ini dilakukan sebanyak 10 kali percobaan. Sedangkan data hasil pengujian waktu respon sistem ketika semua infus habis ditunjukkan pada Tabel 6.



Gambar 13 Hasil Pengujian Waktu Respon Semua Infus Habis

Tabel 6 Data Hasil Pengujian Waktu Respon Infus Cadangan Habis

Pengujian Ke-	Data yang dikirim	Waktu Respon (Detik)
1	/cekinfus	11,43
2	/cekinfus	12,07
3	/cekinfus	12,30
4	/cekinfus	08,25
5	/cekinfus	15,61
6	/cekinfus	08,14

7	/cekinfus	11,48
8	/cekinfus	21,13
9	/cekinfus	11,12
10	/cekinfus	08,39

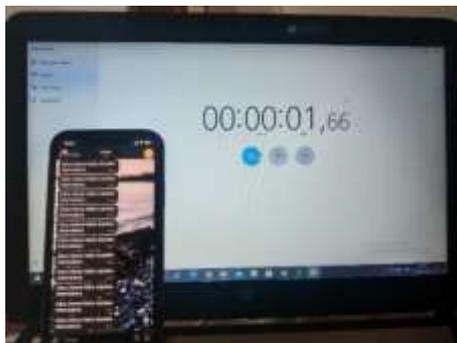
Berdasarkan pengujian yang dilakukan, waktu respon tercepat terjadi pada pengujian ke-4 dengan waktu respon mencapai 08,25 detik, sedangkan waktu respon terlama terjadi pada pengujian ke-8 dengan waktu respon mencapai 21,13 detik. Waktu respon rata-rata dari 10 kali pengujian waktu pengiriman data adalah 13,29 detik. Penyebab terjadinya perbedaan waktu respon dari 10 kali pengujian, adalah karena beberapa hal yang tidak bisa dikontrol oleh sistem secara langsung, sebagai contoh, kualitas koneksi internet dan Telegram yang digunakan oleh banyak orang, sehingga proses pengiriman data mengalami *delay*.

**4.2.4 Uji Penerimaan Data dari NodeMCU**

Pengujian ini dilakukan untuk menguji apakah NodeMCU ESP8266 dapat mengirim data ke bot\_infus Telegram sesuai dengan kondisi dari ketersediaan cairan infus. Pengujian ini dilakukan dalam beberapa tahap dengan masing-masing tahapan dilakukan 10 kali percobaan.

**a. Penerimaan Data Secara Otomatis saat Infus Utama Habis**

Pengujian dilakukan dengan menghitung waktu respon penerimaan data yang dikirim NodeMCU ESP8266 ke bot Telegram secara otomatis ketika infus utama habis sebanyak 1 kali dalam rentang waktu 1 menit. Pengujian waktu respon penerimaan data ketika infus utama habis ditunjukkan pada Gambar 14. Sedangkan data hasil pengujian waktu respon penerima data saat infus utama habis ditunjukkan pada Tabel 7.



**Gambar 14** Pengujian Waktu Respon Penerimaan Data Saat Infus Utama Habis

**Tabel 7** Data Hasil Pengujian Waktu Respon Penerimaan Data Saat Infus Utama Habis

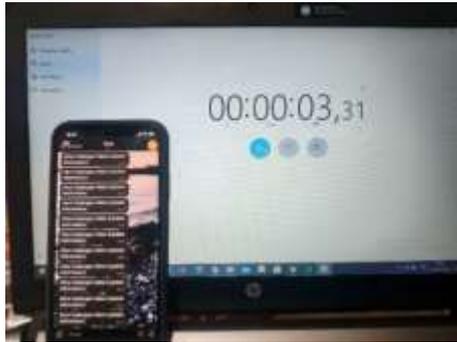
Pengujian Ke-	Data yang Diterima	Waktu Respon (Detik)
1	"Infus Utama Habis"	01,44
2	"Infus Utama Habis"	01,30
3	"Infus Utama Habis"	01,40
4	"Infus Utama Habis"	02,76
5	"Infus Utama Habis"	03,02
6	"Infus Utama Habis"	02,54
7	"Infus Utama Habis"	03,85
8	"Infus Utama Habis"	03,30
9	"Infus Utama Habis"	04,80
10	"Infus Utama Habis"	06,50

Berdasarkan pengujian yang telah dilakukan, waktu respon tercepat terjadi pada pengujian ke-2 dengan waktu respon mencapai 01,30 detik, sedangkan waktu respon terlama terjadi pada pengujian ke-10 dengan waktu respon mencapai 06,50 detik. Waktu respon rata-rata penerimaan data NodeMCU ESP8266 ke bot\_infus Telegram dari 10 kali pengujian adalah 3 detik. Penyebab terjadinya perbedaan waktu respon dari 10 kali pengujian, adalah karena beberapa hal yang tidak bisa dikontrol oleh sistem secara langsung, sebagai contoh, kualitas koneksi internet dan Telegram yang digunakan oleh banyak orang, sehingga proses pengiriman data mengalami *delay*.

**b. Penerimaan Data Secara Otomatis saat Infus Cadangan Habis**

Pengujian dilakukan dengan menghitung waktu respon penerimaan data yang dikirim NodeMCU ESP8266 ke bot Telegram secara otomatis ketika infus cadangan habis sebanyak 1 kali dalam rentang waktu 1 menit. Pengujian waktu respon penerimaan data ketika infus cadangan habis ditunjukkan pada Gambar 15. Sedangkan data hasil pengujian waktu

respon penerima data saat infus cadangan habis ditunjukkan pada Tabel 8.



**Gambar 15** Pengujian Waktu Respon Penerimaan Data Saat Infus Cadangan Habis

**Tabel 8** Data Hasil Pengujian Waktu Respon Penerimaan Data Saat Infus Cadangan Habis

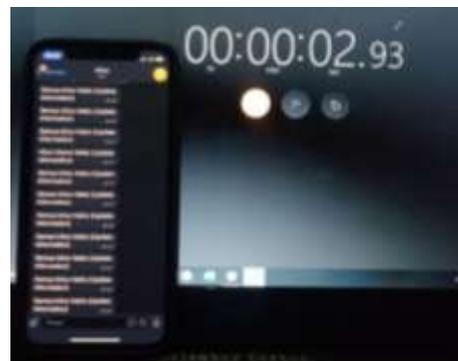
Pengujian Ke-	Data yang Diterima	Waktu Respon (Detik)
1	"Infus Cadangan Habis"	01,83
2	"Infus Cadangan Habis"	05,50
3	"Infus Cadangan Habis"	03,25
4	"Infus Cadangan Habis"	01,12
5	"Infus Cadangan Habis"	01,10
6	"Infus Cadangan Habis"	01,30
7	"Infus Cadangan Habis"	02,25
8	"Infus Cadangan Habis"	02,40
9	"Infus Cadangan Habis"	02,04
10	"Infus Cadangan Habis"	02,10

Berdasarkan pengujian yang telah dilakukan, waktu respon tercepat terjadi pada pengujian ke-5 dengan waktu respon mencapai 01,10 detik, sedangkan waktu respon terlama terjadi pada pengujian ke-2 dengan waktu respon mencapai 05,50 detik. Waktu respon rata-rata penerimaan data NodeMCU ESP8266 ke bot infus Telegram dari 10 kali pengujian adalah 2,2 detik. Penyebab terjadinya perbedaan waktu respon dari 10 kali pengujian, adalah karena beberapa hal yang tidak bisa dikontrol oleh sistem secara langsung, sebagai contoh, kualitas koneksi internet

dan Telegram yang digunakan oleh banyak orang, sehingga proses pengiriman data mengalami *delay*.

**c. Penerimaan Data Secara Otomatis saat Semua Infus Habis**

Pengujian dilakukan dengan menghitung waktu respon penerimaan data yang dikirim NodeMCU ESP8266 ke bot Telegram secara otomatis ketika infus utama habis sebanyak 1 kali dalam rentang waktu 1 menit. Pengujian waktu respon penerimaan data ketika semua infus habis ditunjukkan pada Gambar 16. Sedangkan data hasil pengujian waktu respon penerima data saat infus cadangan habis ditunjukkan pada Tabel 9.



**Gambar 16** Pengujian Waktu Respon Penerimaan Data Saat Semua Infus Habis

**Tabel 9** Data Hasil Pengujian Waktu Respon Penerimaan Data Saat Semua Infus Habis

Pengujian Ke-	Data yang Diterima	Waktu Respon (Detik)
1	"Semua Infus Habis"	02,15
2	"Semua Infus Habis"	01,12
3	"Semua Infus Habis"	01,70
4	"Semua Infus Habis"	02,15
5	"Semua Infus Habis"	03,00
6	"Semua Infus Habis"	02,40
7	"Semua Infus Habis"	03,50
8	"Semua Infus Habis"	04,10
9	"Semua Infus Habis"	02,42
10	"Semua Infus Habis"	02,20

Berdasarkan pengujian yang telah dilakukan, waktu respon tercepat terjadi pada pengujian ke-2 dengan waktu respon mencapai 01,12 detik, sedangkan waktu respon terlama terjadi pada pengujian ke-8 dengan waktu respon mencapai 04,10 detik. Waktu respon rata-rata penerimaan data NodeMCU ESP8266 ke bot\_infus Telegram dari 10 kali pengujian adalah 2,4 detik. Penyebab terjadinya perbedaan waktu respon dari 10 kali pengujian, adalah karena beberapa hal yang tidak bisa dikontrol oleh sistem secara langsung, sebagai contoh, kualitas koneksi internet dan Telegram yang digunakan oleh banyak orang, sehingga proses pengiriman data mengalami *delay*.

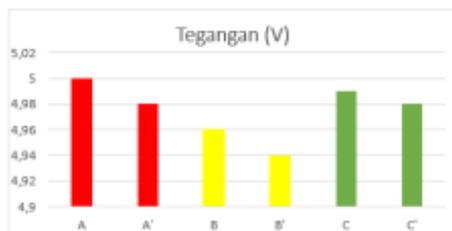
**4.2.5 Pengujian Konsumsi Daya Listrik**

Pengujian konsumsi daya listrik dilakukan untuk mengetahui jumlah besaran daya yang dibutuhkan oleh sistem ketika sedang beroperasi. Pengukuran besar tegangan dan arus pada sistem menggunakan multimeter digital. Tabel 10 menunjukkan data hasil pengujian konsumsi daya.

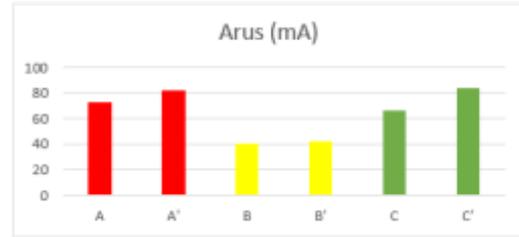
**Tabel 10** Data Hasil Pengujian Konsumsi Daya

Jenis Beban	Kondisi Beban	Volt (V)	Arus (mA)	Daya (Watt)
NodeMCU ESP8266	Standby (A)	5	73	0.37
	Terhubung Internet (A')	4.98	82	0.41
Sensor Inframerah	Logika 0 (B)	4.96	40	0.20
	Logika 1 (B')	4.94	42	0.21
Motor Servo	Standby (C)	4.99	66	0.33
	Berputar 90° dan 180° (C')	4.98	84	0.42

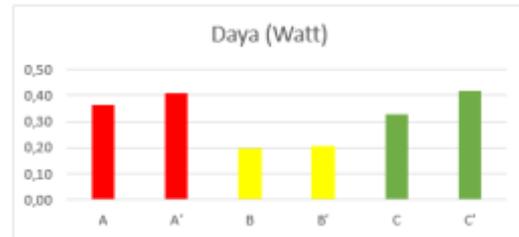
Grafik pengujian konsumsi daya ditunjukkan pada Gambar 17, Gambar 18, dan Gambar 19.



**Gambar 17** Grafik Tegangan Pada Masing-masing Beban



**Gambar 18** Grafik Arus Pada Masing-masing Beban



**Gambar 19** Grafik Konsumsi Daya Pada Masing-masing Beban

Berdasarkan Tabel 10, dan grafik pada Gambar 17, Gambar 18, dan Gambar 19, konsumsi daya terbesar terjadi ketika motor servo beroperasi sebagai katup saluran pada cairan infus yang mencapai daya sebesar 0,42 watt. Sedangkan konsumsi daya terendah yaitu 0,20 watt, ketika sensor inframerah berlogika 0.

**5. SIMPULAN**

Hasil menunjukkan bahwa prototipe sistem *monitoring intravenous fluid drops* dapat bekerja dengan baik dan sesuai dengan rancangan. Hal ini, sesuai dengan hasil pengujian komunikasi antara prototipe dengan Telegram Bot yang menunjukkan respon sistem tercepat untuk mengirimkan data sensor ke Telegram Bot mencapai 01,10 detik. Sedangkan respon sistem tercepat untuk menerima perintah dari Telegram Bot mencapai 05,70 detik.

Hasil pengujian sistem pergantian otomatis *intravenous fluid drops* yang menggunakan motor servo sebagai katup pada saluran infus dapat bekerja dengan baik, yaitu motor servo dapat membuka dan menutup saluran infus sesuai dengan kondisi dari ketersediaan cairan infus pada masing-masing kantong infus.

Konsumsi daya terbesar terjadi ketika NodeMCU ESP8266 terhubung dengan internet dan motor servo berputar untuk menutup katup saluran infus (berputar 90°) yaitu sebesar 0.41 Watt dan 0.42 Watt.

## 6. DAFTAR PUSTAKA

- [1] Zainuri, A., Santoso, D. R. & Muslim, M. A., 2012. *Monitoring dan Identifikasi Gangguan Infus Menggunakan Mikrokontroler AVR*. *Jurnal EECCIS*, VI(1), pp. 49-54.
- [2] Muljodipo, N., Sherwin, Sompie & Robot, R. F., 2015. Rancang Bangun Otomatis Sistem Infus. *E-Journal Teknik Elektro dan Komputer*, IV(4), pp. 12-22.
- [3] Muflih, G. Z., Sunardi & Yudhana, A., 2019. *Perancangan Sistem Monitoring Hujan Berbasis Arduino Uno Dan Telegram Messenger*. Semnas Ristek 2019.
- [4] Fauzia, N. & Risna, 2020. Tingkat Kepatuhan Perawat Dalam Melaksanakan Standar Operasional Prosedur Pemasangan Infus. *Jurnal Real Riset*, II(2), pp. 69-80.
- [5] Nataliana, D., Taryana, N. & Riandita, E., 2016. Alat *Monitoring* Infus Set pada Pasien Rawat Inap Berbasis Mikrokontroler ATmega 8535. *Jurnal ELKOMIKA*, IV(1), pp. 1-15.
- [6] IR Infrared, 2018. *Arduino IR Infrared Obstacle Avoidance Sensor Module*. [Online] Tersedia di: <https://5.imimg.com/data5/YT/KV/MY-1833510/arduino-ir-infrared-obstacle-avoidance-sensor-module.pdf> [Diakses Pada 7 Juni 2021].
- [7] Dwitama, A. P., Janardana, I. G. N. & Wijaya, I. W. A., 2021. Rancang Bangun Prototipe Pemantau Kebocoran Gas Menggunakan Sensor MQ-6 Berbasis NodeMCU ESP8266. *Jurnal SPEKTRUM*, VIII(1), pp. 9-14.
- [8] Component101, 2020. *NodeMCU ESP8266 Datasheet*. [Online] Tersedia di: <https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet> [Diakses Pada 7 Juni 2021].
- [9] Darmika, A. A. A., Raka Agung, I. G. A. P. & Divayana, Y., 2019. Prototipe Pemberi Pakan Ikan Dan Penggantian Air Pada Akuarium Berbasis Mikrokontroler ATmegaA328P. *Jurnal SPEKTRUM*, VI(2), pp. 72-77.
- [10] Cornelam, 2020. *Arduino Servo Motors: 5 Steps (with Pictures)*. [Online] Tersedia di: <https://www.instructables.com/Arduino-Servo-Motors/> [Diakses Pada 7 Juni 2021].
- [11] Setiawan, D. A., 2019. Prototype Home Security System Dengan Autentifikasi KTP-EL. *Jurnal Elektronik Pendidikan Teknik Elektronika*, VIII(1), pp. 19-32.
- [12] Wibowo, A. K. N. & Kurniawan, Y. I., 2019. Bot Telegram Sebagai Media Alternatif Akses Informasi Akademik. *Jurnal Ilmiah Komputer dan Informatika*, VIII(1), pp. 1-10.