



## **TUGAS AKHIR**

# **IMPLEMENTASI SISTEM OTOMATIS DAN MONITORING INFUS BERBASIS IOT**

**13322008 PAIAN MANALU**

**13322020 ERNA MELIANA MANURUNG**

**13322029 BATARA HASINTONGAN NADAPDAP**

**PROGRAM STUDI DIPLOMA III TEKNOLOGI KOMPUTER  
FAKULTAS VOKASI  
INSTITUT TEKNOLOGI DEL  
2025**



## **TUGAS AKHIR**

### **IMPLEMENTASI SISTEM OTOMATIS DAN MONITORING INFUS BERBASIS INTERNET OF THINGS (IOT)**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Ahli  
Madya Teknik (A.Md.T)**

**13322008 PAIAN MANALU**

**13322020 ERNA MELIANA MANURUNG**

**13322029 BATARA HASINTONGAN NADAPDAP**

**PROGRAM STUDI DIPLOMA III TEKNOLOGI KOMPUTER  
FAKULTAS VOKASI  
INSTITUT TEKNOLOGI DEL  
2025**

## **PERNYATAAN ORISINALITAS**

**Tugas akhir ini adalah hasil karya kami sendiri. Seluruh sumber informasi yang dikutip atau dirujuk telah kami cantumkan dengan benar.**

1. Nama : Paian Manalu  
NIM : 13322008  
Tanda Tangan :

Tanggal : 31 Juli 2025

2. Nama : Erna Meliana Manurung  
NIM : 13322020  
Tanda Tangan :

Tanggal : 31 Juli 2025

3. Nama : Batara Hasintongan Nadapdap  
NIM : 13322029  
Tanda Tangan :

Tanggal : 31 Juli 2025

## **HALAMAN PENGESAHAN**

Tugas Akhir ini diajukan oleh:

1. Nama : Paian Manalu  
NIM : 13322008
2. Nama : Erna Meliana Manurung  
NIM : 13322020
3. Nama : Batara Hasintongan Nadapdap  
NIM : 13322029

Program Studi : Diploma III Teknologi Komputer

Judul Tugas Akhir : Implementasi Sistem Otomatis dan Monitoring Infus Berbasis *Internet of Things* (IoT)

**Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya Teknik (A.Md.T) pada Program Studi Diploma III Teknologi Komputer, Fakultas Vokasi, Institut Teknologi Del.**

## **DEWAN PENGUJI**

Pembimbing 1 : Frengki Simatupang, S.Tr. Kom., M.T. ( )

Penguji 1 : Pandapotan Siagian, ST, M.Eng ( )

Penguji 2 : Sari Muthia Silalahi, S.Pd., M.Ed ( )

Ditetapkan di : Laguboti

Tanggal : 31 Juli 2025

## **PERNYATAAN PERSETUJUAN PUBLIKASI**

Kami yang bertanda tangan di bawah ini, sebagai civitas akademik Institut Teknologi Del:

1. Nama : Paian Manalu  
NIM : 133  
22008
2. Nama : Erna Meliana Manurung  
NIM : 13322020
3. Nama : Batara Hasintongan Nadapdap  
  
NIM : 13322029  
Program Studi : Diploma III Teknologi Komputer  
Fakultas : Fakultas Vokasi  
Jenis Karya : Tugas Akhir

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Del **Hak Bebas Royalti Non Eksklusif** atas Tugas Akhir yang berjudul: **Implementasi Sistem Otomatis dan Monitoring Infus Berbasis Internet of Things (IoT)** beserta seluruh perangkat yang menyertainya. Dengan Hak Non Eksklusif ini, Institut Teknologi Del berhak menyimpan, mengalihmediakan, mengubah format, serta mempublikasikannya dalam bentuk pangkalan data apa pun, selama tetap mencantumkan nama kami sebagai penulis/ dan pemilik Hak Cipta.

Demikian pernyataan ini kami buat dengan sebenarnya.

Laguboti, 31 Juli 2024

Yang menyatakan:

Paian Manalu

Erna Meliana Manurung

Batara Hasintongan Nadapdap

## ABSTRAK

Nama	:	1. Paian Manalu 2. Erna Meliana Manurung 3. Batara Hasintongan Nadapdap
Program Studi	:	Diploma III Teknologi Komputer
Judul	:	Implementasi Sistem Otomatis dan Monitoring Infus Berbasis Internet of Things (IOT)

Perkembangan teknologi *Internet of Things* (IoT) telah mendorong inovasi di bidang layanan kesehatan, khususnya dalam sistem pemantauan dan pengendalian infus otomatis. Penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan sistem pemantauan infus otomatis berbasis IoT yang dapat meningkatkan efisiensi dan akurasi layanan kesehatan di rumah sakit. Sistem dirancang menggunakan mikrokontroler ESP32, sensor load cell untuk mengukur volume cairan infus, sensor photodiode untuk mendeteksi laju tetesan, serta motor servo untuk mengatur aliran infus secara otomatis. Data hasil pengukuran dikirimkan secara *real-time* ke *platform* situs web melalui protokol MQTT dan dapat diakses oleh petugas medis untuk pemantauan jarak jauh. Sistem ini juga dilengkapi dengan fitur notifikasi berupa buzzer dan tampilan peringatan pada situs web ketika volume cairan infus mendekati ambang batas minimum. Hasil pengujian menunjukkan bahwa sistem mampu mengontrol aliran infus secara otomatis, memantau volume cairan dan laju tetesan secara akurat, serta memberikan notifikasi tepat waktu ketika cairan infus hampir habis. Dengan demikian, sistem ini dapat membantu mengurangi beban kerja tenaga medis, meminimalkan risiko keterlambatan penggantian infus, serta meningkatkan keselamatan dan kenyamanan pasien selama terapi infus. Implementasi sistem ini diharapkan dapat menjadi solusi modern dalam manajemen infus di fasilitas pelayanan kesehatan serta mendukung transformasi digital dalam layanan medis.

**Kata Kunci :** *Internet of Things* (IoT), Sistem Pemanfaatan Infus Otomatis, Transmisi Data Real-time, Protokol MQTT

## ABSTRACT

Name	:	1. Paian Manalu 2. Erna Meliana Manurung 3. Batara Hasintongan Nadapdap
Study Program	:	Associate's Degree in Computer Technology
Title	:	Implementation of an Automated and Monitoring Infusion System Based on Internet Of Things (IOT)

The development of Internet of Things (IoT) technology has driven innovation in the field of healthcare, particularly in the area of automated infusion monitoring and control systems. This study aims to develop and implement an IoT-based automated infusion monitoring system that can improve the efficiency and accuracy of healthcare services in hospitals. The system is designed using an ESP32 microcontroller, a load cell sensor to measure the volume of infusion fluid, a photodiode sensor to detect the drip rate, and a servo motor to automatically regulate the infusion flow. Measurement data is transmitted in real-time to a website platform via the MQTT protocol and can be accessed by medical staff for remote monitoring. The system also features a notification system with a buzzer and warning display on the website when the infusion fluid volume approaches the minimum threshold. Test results show that the system can automatically control the infusion flow, accurately monitor fluid volume and drip rate, and provide timely notifications when the infusion is nearly depleted. As a result, this system can help reduce the workload of medical staff, minimise the risk of delayed infusion replacement, and enhance patient safety and comfort during infusion therapy. The implementation of this system is expected to serve as a modern solution for infusion management in healthcare facilities and support the digital transformation of medical services.

**Kata Kunci :** Internet of Things (IoT), Automated Infusion Monitoring System, Real-time Data Transmission, MQTT Protocol

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa karena atas rahmat, kasih, dan karunia-Nya, penulis dapat menyelesaikan penyusunan Tugas Akhir yang berjudul “Implementasi Sistem Otomatis dan Monitoring Infus Berbasis *Internet of Things (IoT)*”. Tugas Akhir ini disusun sebagai salah satu syarat untuk menyelesaikan studi pada program Diploma III Teknologi Komputer di Institut Teknologi Del. Dalam proses

penyusunan dan pelaksanaan penelitian ini, penulis mendapatkan banyak dukungan, bimbingan, serta masukan yang sangat berarti dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih dan penghargaan sebesar-besarnya kepada:

1. Bapak Frengki Simatupang, S.Tr.Kom., M.T. selaku dosen pembimbing yang telah dengan penuh kesabaran memberikan arahan, bimbingan, dan motivasi selama proses penyusunan hingga selesaiya Tugas Akhir ini.
2. Bapak Pandapotan Siagian, S.T., M.Eng. selaku dosen penguji 1, dan Ibu Sari Muthia Silalahi, S.Pd., M.Ed. selaku dosen penguji 2, atas kritik konstruktif, saran yang membangun, serta waktu dan perhatian yang telah diberikan selama proses ujian Tugas Akhir.
3. Orang tua dan keluarga tercinta, atas segala doa, cinta kasih, dukungan moril maupun materil, serta
4. semangat yang tak henti-hentinya dalam setiap langkah penulis.
5. Teman-teman Kelompok 8 Tugas Akhir, atas kerjasama, kebersamaan, dan semangat tim yang luar biasa dalam menyelesaikan proyek ini secara maksimal.

Laguboti, 31 Juli 2025

Penulis 1

Penulis 2

Penulis 3

Paian Manalu

Erna Meliana Manurung

Batara Hasintongan Nadapdap

## DAFTAR ISI

<b>PERNYATAAN ORISINALITAS.....</b>	<b>ii</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>iii</b>
<b>PERNYATAAN PERSETUJUAN PUBLIKASI.....</b>	<b>iv</b>
<b>ABSTRAK .....</b>	<b>v</b>
<b>ABSTRACT .....</b>	<b>vi</b>
<b>KATA PENGANTAR.....</b>	<b>vii</b>
<b>DAFTAR ISI.....</b>	<b>viii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xii</b>
<b>DAFTAR TABEL .....</b>	<b>xii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1    Latar Belakang .....	1
1.2    Tujuan .....	3
1.3    Rumusan Masalah .....	3
1.4    Batasan Penelitian .....	3
1.5    Hasil yang Diharapkan.....	4
1.6    Sistematika Penyajian .....	5
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>6</b>
2.1    Landasan Teori.....	6
2.1.1    Sistem Otomatis .....	6
2.1.2    Sistem Monitoring.....	7
2.1.3    Internet of Things (IoT) .....	8
2.1.4    Sistem Infus.....	9
2.1.5    Sistem Infus Manual .....	10
2.1.6    Sistem Infus Berbasis IoT .....	11
2.2    Perangkat Keras ( <i>Hardware</i> ) .....	12
2.2.1    ESP32.....	12
2.2.2    Sensor Load Cell.....	13
2.2.3    Modul HX711 .....	14
2.2.4    Photodioda .....	14

2.2.5	Keypad .....	15
2.2.6	Motor Servo .....	16
2.2.7	Buzzer .....	17
2.2.8	Modul I2C LCD .....	17
2.2.9	PCB (Printed Circuit Board) .....	18
2.3	Perangkat Lunak ( <i>Software</i> ) .....	19
2.3.1	Arduino IDE .....	19
2.3.2	MySQL .....	20
2.3.3	Visual Studio Code .....	20
2.3.4	MQTT .....	21
2.3.5	Mosquitto .....	22
2.3.6	Node-RED .....	23
2.4	Related Work .....	24
2.4.1	Monitoring Cairan Infus Menggunakan Load Cell Berbasis IoT ..	24
2.4.2	Sistem Kontrol dan Monitoring Infus Berbasis NodeMCU .....	24
2.4.3	Alat Monitoring Tetesan Infus Menggunakan Web Berbasis ESP8266 .....	25
2.4.4	Sistem Monitoring dan Controlling Cairan Infus Berbasis Web ..	27
2.4.5	System Monitoring Infus Menggunakan HX711 .....	27
	<b>BAB III ANALISIS DAN DESAIN SISTEM .....</b>	<b>29</b>
3.1	Analisis .....	29
3.1.1	Analisis Masalah .....	29
3.1.2	Analisis Pemecahan Masalah .....	30
3.1.3	Analisis Kebutuhan Sistem .....	31
3.2	Desain .....	38
3.2.1	Topologi Sistem Pemantauan Infus Otomatis .....	38
3.2.2	Desain <i>Flowchart</i> Sistem Infus .....	41
3.2.3	Desain Hardware .....	43
3.2.4	Perancangan Penempatan Loadcell .....	46
3.2.5	Desain Software .....	47
3.2.6	Desain Case Pengatur Tetesan .....	51

3.2.7	Desain Perancangan <i>Case</i> .....	52
3.2.8	Desain Fisik Sistem Infus Otomatis.....	53
3.2.9	Kalibrasi Loadcell .....	54
3.2.10	Kalibrasi Motor Servo.....	58
3.2.11	Skenario Pengujian Penginputan Data melalui Keypad.....	59
3.2.12	Skenario Pengujian Menghitung Volume Infus.....	61
3.2.13	Skenario Pengujian Penginputan Data Pasien Melalui Website ...	62
3.2.14	Skenario Pengujian Menghitung Tetesan Infus .....	64
3.2.15	Skenario Pengujian Notifikasi melalui Buzzer .....	65
<b>BAB IV IMPLEMENTASI DAN HASIL PENGUJIAN.....</b>		<b>67</b>
4.1	Instalasi .....	67
4.1.1	Instalasi MQTT .....	67
4.1.2	Instalasi Node-RED .....	69
4.2	Implementasi .....	69
4.2.1	Implementasi Node-RED .....	69
4.2.2	Implementasi Inputan Data Pasien melalui Website.....	72
4.2.3	Implementasi Inputan TPM melalui Keypad .....	73
4.2.4	Implementasi Penghitungan Volume Infus melalui Loadcell....	73
4.2.5	Implementasi Pengaturan Tetesan melalui Servo Motor .....	74
4.2.6	Implementasi Penghitungan Tetesan melalui Sensor Photodiode	75
4.3	Hasil Pengujian .....	76
4.3.1	Tes Script Butir-Uji-1 .....	76
4.3.2	Tes Script Butir-Uji-2 .....	79
4.3.3	Tes Script Butir-Uji-3 .....	82
4.3.4	Tes Script Butir-Uji-4 .....	85
4.3.5	Tes Script Butir-Uji-5 .....	88
4.3.6	Hasil Pengujian Durasi Habisnya Infus .....	93
4.3.7	Hasil Pengujian Waktu Habisnya Infus Berdasarkan Volume Infus .....	95
<b>BAB V KESIMPULAN DAN SARAN .....</b>		<b>98</b>
5.1	Kesimpulan .....	98

5.2	Saran.....	98
5.3	Estimasi Biaya Penelitian.....	99
<b>DAFTAR PUSTAKA .....</b>		<b>xiv</b>
<b>LAMPIRAN.....</b>		<b>xviii</b>

## DAFTAR GAMBAR

Gambar 2. 1 ESP32 .....	12
Gambar 2. 2 Sensor Load Cell .....	13
Gambar 2. 3 Modul HX711 .....	14
Gambar 2. 4 Photodioda.....	14
Gambar 2. 5 Keypad .....	15
Gambar 2. 6 Motor Servo.....	16
Gambar 2. 7 Buzzer.....	17
Gambar 2. 8 Modul I2C LCD .....	17
Gambar 2. 9 Printed Circuit Board .....	18
Gambar 2. 10 Arduino IDE.....	19
Gambar 2. 11 MySQL.....	20
Gambar 2. 12 Visual Studio Code .....	20
Gambar 2. 13 MQTT.....	21
Gambar 2. 14 Mosquitto .....	22
Gambar 2. 15 Node-RED.....	23
Gambar 3. 1 Topologi Sistem Pemantauan Infus Otomatis .....	38
Gambar 3. 2 Desain Flowchart Sistem Infus .....	41
Gambar 3. 3 Desain Hardware .....	44
Gambar 3. 4 Perancangan Penempatan Loadcell .....	46
Gambar 3. 5 Desain Tampilan Login.....	47
Gambar 3. 6 Desain Tampilan Dashboard .....	48
Gambar 3. 7 Desain Tampilan Data Riwayat Pasien .....	49
Gambar 3. 8 Desain Tampilan Notifikasi .....	50
Gambar 3. 9 Desain Case Pengatur Tetesan .....	51
Gambar 3. 10 Desain Case .....	52
Gambar 3. 11 Desain Fisik Sistem Infus.....	53
Gambar 3. 12 Skenario Pengujian Penginputan Data melalui Keypad.....	59
Gambar 3. 13 Skenario Pengujian Menghitung Volume Infus .....	61
Gambar 3. 14 Skenario Pengujian Menghitung Tetesan Infus .....	64
Gambar 4. 1 Arsitektur Dasar Protokol MQTT .....	68
Gambar 4. 2 Implementasi Inputan Data Pasien melalui Website .....	72
Gambar 4. 3 Inputan TPM melalui Keypad .....	73
Gambar 4. 4 Implementasi Pengaturan Tetesan melalui Servo Motor .....	74
Gambar 4. 5 Implementasi Perhitungan Tetesan melalui Sensor Photodiode .....	75
Gambar 4. 6 Penginputan Nilai TPM melalui Keypad .....	78
Gambar 4. 7 Hasil Inputan Keypad.....	78
Gambar 4. 8 Pengujian Menghitung Volume Infus menggunakan Loadcell.....	81
Gambar 4. 9 Data Volume Infus dengan Volume Infus 434 gram .....	81
Gambar 4. 10 Tampilan Website dengan Volume Infus 66% .....	81
Gambar 4. 11 Input Data Pasien .....	84
Gambar 4. 12 Tampilan Data Pasien yang Sudah di Input .....	84
Gambar 4. 13 Tampilan Database Pasien yang di Input .....	84
Gambar 4. 14 Penghitungan Tetesan Infus Menggunakan Photodiode .....	87
Gambar 4. 15 Tampilan Nilai TPM Sesuai Deteksi Tetesan Cairan.....	87
Gambar 4. 16 Rangkaian Sistem Buzzer.....	90

Gambar 4. 17 Data Volume Infus dengan Kondisi > 145.....	90
Gambar 4. 18 Tampilan Website dengan Kondisi Volume Infus 37%.....	91
Gambar 4. 19 Data Volume Infus dengan Kondisi < 135 gram .....	91
Gambar 4. 20 Tampilan Website dengan Kondisi Volume Infus Habis.....	91
Gambar 4. 21 Tampilan Notifikasi pada Website.....	92
Gambar 4. 22 Grafik Estimasi Durasi Habisnya Infus.....	95
Gambar 4. 23 Hasil Pengujian Waktu Habisnya Infus Berdasarkan Volume Infus .....	96

## **DAFTAR TABEL**

Tabel 3. 1 Analisis Kebutuhan Perangkat Keras ( <i>Hardware</i> ).....	31
Tabel 3. 2 Analisis Kebutuhan Perangkat Lunak ( <i>Software</i> ) .....	36
Tabel 3. 3 Kalibrasi Loadcell .....	54
Tabel 3. 4 Kalibrasi Motor Servo.....	58
Tabel 3. 5 Skenario Pengujian Penginputan Data.....	60
Tabel 3. 6 Skenario Pengujian Menghitung Volume Infus.....	61
Tabel 3. 7 Skenario Pengujian Penginputan Data Pasien melalui Website .....	63
Tabel 3. 8 Skenario Pengujian Menghitung Tetesan Infus .....	64
Tabel 3. 9 Skenario Pengujian Notifikasi melalui Buzzer .....	65
Tabel 4. 1 Pengujian Penginputan Data melalui Keypad .....	76
Tabel 4. 2 Pengujian Menghitung Volume Infus .....	79
Tabel 4. 3 Pengujian Penginputan Data Pasien Melalui Website .....	82
Tabel 4. 4 Pengujian Menghitung Tetesan Infus .....	85
Tabel 4. 5 Pengujian Notifikasi melalui Buzzer .....	88
Tabel 4. 6 Hasil Pengujian Volume Cairan Infus terhadap Aktivasi Notifikasi ...	92
Tabel 4. 7 Tabel Hasil Pengujian Durasi Habisnya Infus .....	94
Tabel 4. 8 Hasil Pengujian Waktu Infus Berdasarkan Volume Infus .....	96
Tabel 5. 1 Estimasi Biaya.....	99

## **BAB I**

### **PENDAHULUAN**

Pada bab ini berisi penjelasan mengenai latar belakang pemilihan dari topik permasalahan yang akan diselesaikan, tujuan pelaksanaan Tugas Akhir, ruang lingkup penggerjaan yang akan diselesaikan, ruang lingkup penggerjaan yang akan dibangun, pendekatan yang dilakukan dan sistematika penulisan laporan yang dilakukan pada penggerjaan Tugas Akhir.

#### **1.1 Latar Belakang**

Perkembangan teknologi yang semakin pesat dan canggih dalam era modern saat ini telah membawa dampak yang signifikan dalam berbagai bidang, termasuk dalam sektor kesehatan [1]. Di bidang medis, kebutuhan akan perangkat-perangkat yang lebih canggih, efisien, dan mendukung proses perawatan pasien secara optimal semakin meningkat. Salah satu perangkat yang saat ini sangat dibutuhkan adalah alat pemantauan dan pengontrol infus secara jarak jauh, yang menawarkan solusi lebih efisien bagi tenaga medis dalam melakukan pemantauan pasien [2].

Infus merupakan salah satu alat kesehatan yang sangat penting dan sering digunakan dalam berbagai kondisi medis [3]. Alat ini berfungsi untuk menggantikan cairan yang hilang dari tubuh dan membantu menjaga keseimbangan cairan tubuh yang terganggu akibat penyakit atau kondisi medis tertentu. Namun, pengelolaan infus secara manual menghadapi sejumlah kendala. Berdasarkan survei di RSUD Porsea, pemantauan infus dilakukan secara berkala dengan frekuensi setiap 30 menit hingga 2 jam. Tantangan yang sering dihadapi meliputi pengaturan tetesan yang tidak akurat, waktu pengawasan yang signifikan terutama dalam kondisi beban kerja tinggi serta Potensi keterlambatan penggantian cairan infus, yang berisiko bagi kesehatan pasien.

Data dari beberapa rumah sakit seperti RS Awal Bros, RS Paru, RS Murni Teguh, dan RS Adam Malik menunjukkan bahwa penggunaan infus otomatis (*infus pump*) secara signifikan meningkatkan efisiensi perawatan. Tenaga medis melaporkan

waktu pengaturan yang jauh lebih singkat, akurasi pemberian cairan yang lebih tinggi, serta penurunan risiko komplikasi seperti infeksi atau kesalahan dosis. Namun, penggunaan infus otomatis belum sepenuhnya diadopsi secara luas. Di RSUD Porsea, pengawasan manual masih menjadi metode utama, yang tidak hanya memakan waktu tetapi juga berpotensi meningkatkan risiko keterlambatan akibat beban kerja tenaga medis yang tinggi. Dalam kondisi seperti ini, implementasi sistem infus berbasis *Internet of Things* (IoT) menjadi solusi potensial. Dengan dukungan teknologi IoT, proses pemantauan dan pengontrolan infus dapat dilakukan secara otomatis dan *real-time* [4]. Alat ini akan dipasang pada perangkat infus pasien dan dihubungkan dengan jaringan internet, sehingga setiap perubahan pada kondisi cairan infus dapat langsung terdeteksi. Informasi yang dihasilkan oleh alat ini, seperti jumlah cairan yang tersisa serta laju tetesan infus, akan dikirimkan ke sebuah platform berbasis *website*. Tenaga medis kemudian dapat mengakses *website* tersebut untuk memantau kondisi infus tanpa harus melakukan pengecekan manual di ruangan pasien. Sistem ini memungkinkan pemantauan infus dilakukan secara lebih efisien, sehingga risiko keterlambatan dalam penggantian infus dapat diminimalkan.

Sistem pemantauan infus berbasis IoT ini menggunakan sensor *load cell* untuk mengukur berat cairan infus serta jumlah tetesan cairan yang mengalir [5]. Data yang dihasilkan oleh sensor tersebut kemudian diproses dan dikirimkan ke *website* yang terintegrasi dengan sistem. Melalui data ini, tenaga medis dapat melihat berapa banyak cairan yang tersisa dalam kantong infus serta memantau laju tetesan cairan secara *real-time*. Dengan pemantauan yang lebih akurat dan berkesinambungan, petugas medis dapat lebih cepat tanggap dalam mengganti cairan infus yang hampir habis, sehingga perawatan pasien menjadi lebih optimal.

Berdasarkan permasalahan yang ada, sistem pemantauan dan pengontrolan infus berbasis IoT ini dirancang dengan tujuan utama untuk meningkatkan kualitas pelayanan kesehatan di rumah sakit dan meminimalisir risiko keterlambatan dalam perawatan pasien. Dengan sistem ini, perawat dapat lebih mudah memantau dan

mengontrol infus dari jarak jauh melalui platform yang tersedia. Alat ini tidak hanya membantu meningkatkan efisiensi kerja tenaga medis, tetapi juga berkontribusi terhadap peningkatan keamanan dan kenyamanan pasien selama menjalani terapi infus.

Dalam jangka panjang, penerapan teknologi ini diharapkan dapat mendukung upaya rumah sakit dalam memberikan layanan kesehatan yang lebih baik dan responsif. Dengan pemanfaatan IoT dalam perangkat medis seperti infus, proses perawatan menjadi lebih modern dan terintegrasi, sehingga kualitas pelayanan dapat ditingkatkan secara keseluruhan.

## 1.2 Tujuan

Tujuan dari Tugas Akhir ini antara lain :

1. Mengontrol aliran cairan infus secara otomatis: Sistem secara otomatis mengatur aliran cairan infus sesuai kebutuhan pasien, meminimalkan kesalahan manual, dan memungkinkan pengaturan jarak jauh melalui website.
2. Memantau volume cairan infus: Sistem memantau volume cairan infus secara real-time dan mengirimkan notifikasi saat cairan hampir habis, memastikan penggantian tepat waktu untuk kelancaran perawatan pasien.

## 1.3 Rumusan Masalah

Adapun rumusan masalah dalam penggeraan tugas akhir ini adalah sebagai berikut:

1. Bagaimana sistem otomatis bisa mengatur aliran cairan infus dengan tepat dan mengurangi kesalahan manual?
2. Bagaimana sistem monitoring dapat memantau volume cairan infus secara *real-time* dan memberikan notifikasi sebelum cairan habis?

## 1.4 Batasan Penelitian

1. Sistem hanya akan menggunakan platform berbasis web untuk pemantauan, tanpa mencakup aplikasi *mobile* atau metode pemantauan lainnya
2. Sistem pemantauan infus yang dikembangkan hanya terbatas pada pemantauan volume cairan infus dan laju tetesan melalui sensor *load cell*.

Pemantauan kondisi lain seperti tekanan darah atau suhu tubuh pasien tidak termasuk dalam penelitian ini.

3. Sistem infus otomatis berbasis IoT ini dirancang sebagai alat bantu untuk mendukung tugas perawat dalam memantau dan mengelola infus secara efisien. Sistem ini tidak menggantikan tanggung jawab perawat, melainkan hanya membantu meningkatkan akurasi dan efisiensi kerja.
4. Pengisian ulang atau penggantian cairan infus tetap dilakukan secara manual oleh tenaga medis.
5. Sistem ini dikhkusukan hanya untuk 1 jenis kantong infus, yaitu kantong infus yang terbuat dari bahan plastik.
6. Jumlah loadcell yang digunakan pada sistem maksimal 2

### **1.5 Hasil yang Diharapkan**

Adapun hasil yang diharapkan dari penelitian ini adalah :

1. Diharapkan bahwa sistem pemantauan infus berbasis IoT yang dikembangkan akan memungkinkan perawat untuk dengan mudah memasang infus pada pasien dan mengatur laju tetesan per menit menggunakan keypad yang terintegrasi dengan sistem.
2. Diharapkan bahwa sistem ini dapat berjalan secara otomatis dan mengirimkan data *real-time* mengenai volume sisa cairan dan laju tetesan infus ke *database*, yang selanjutnya akan ditampilkan di platform *website* yang mudah diakses oleh tenaga medis.
3. Diharapkan bahwa sistem ini akan mampu memberikan peringatan di *website* serta mengaktifkan bunyi buzzer pada alat ketika volume infus hampir habis, sehingga perawat dapat segera mengganti cairan infus tanpa keterlambatan.

Dengan demikian, sistem ini diharapkan dapat meningkatkan efisiensi dan akurasi dalam pemantauan infus serta membantu tenaga medis dalam memberikan perawatan yang lebih optimal kepada pasien.

## **1.6 Sistematika Penyajian**

Secara garis besar dokumen ini menyajikan enam bab yang disusun dalam sistematika sebagai berikut :

### **1. BAB I PENDAHULUAN**

Pada bab ini berisi penjelasan mengenai latar belakang masalah yang diambil, tujuan, rumusan masalah, batasan penelitian, serta hasil yang diharapkan.

### **2. BAB II TINJAUAN PUSTAKA**

Pada bab ini berisi uraian setiap dasar teori, uraian yang berisi tentang ungkapan-ungkapan atau ringkasan yang berasal dari sumber yang digunakan dalam penggerjaan Tugas Akhir.

### **3. Bab III ANALISIS DAN DESAIN**

Pada bab ini menjelaskan analisis mengenai permasalahan siber dan langkah-langkah perancangan produk yang relevan sesuai dengan hasil analisis yang tercantum dalam metode penelitian. Diimplementasikan pada penggerjaan Tugas Akhir sesuai dengan hasil analisis yang dibuat.

### **4. Bab IV IMPLEMENTASI DAN HASIL PENGUJIAN**

Pada bab ini berisi tentang implementasi dan pengujian terhadap proyek atau sistem yang sudah dibangun.

### **5. Bab V KESIMPULAN DAN SARAN**

Pada bab ini berisi mengenai kesimpulan yang diperoleh setelah menyelesaikan proyek Tugas Akhir ini berisi saran mengenai pengembangan proyek yang dibangun untuk kedepannya.

## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini dijelaskan mengenai tinjauan pustaka sebagai dasar teori yang akan dipergunakan sebagai landasan utama dalam perancangan dan Pembangunan produk yang digunakan dalam pembuatan proyek akhir ini.

#### **2.1 Landasan Teori**

Pada bagian ini akan memuat konsep-konsep dasar dan teori yang relevan dengan penelitian. Berikut adalah struktur isi yang tepat untuk sub-bab ini.

##### **2.1.1 Sistem Otomatis**

Sistem otomatis merupakan sebuah sistem yang dirancang untuk menjalankan tugas tanpa intervensi manual, menggunakan sensor, aktuator, dan algoritma kontrol untuk mengelola proses secara efisien. Sistem ini mampu beradaptasi dengan perubahan lingkungan atau kondisi yang terpantau. Dalam konteks infus berbasis IoT di rumah sakit, sistem otomatis mengatur, memantau, dan mengontrol aliran cairan infus serta mendeteksi gangguan yang mungkin terjadi selama proses infus. Tujuannya adalah meningkatkan efisiensi, akurasi, dan keamanan perawatan pasien, serta memungkinkan monitoring jarak jauh secara *real-time*.

Pada pengaturan laju tetesan (*Drip Rate Control*), sistem otomatis dapat mengatur laju tetesan infus sesuai nilai yang diinginkan, misalnya 62.5 tetes per menit. Sistem ini menggunakan sensor untuk memantau laju tetesan dan aktuator untuk mengontrol aliran infus. Jika laju tetesan terlalu cepat atau lambat, sistem secara otomatis menyesuaikan katup agar laju kembali sesuai dengan yang ditetapkan. Dengan fitur ini, tenaga medis dapat mengatur dan memantau laju infus melalui *website*, memberikan fleksibilitas dan efisiensi dalam penanganan pasien.

Selain itu, sistem otomatis dilengkapi dengan deteksi gangguan atau masalah. Jika selama periode tertentu tidak ada tetesan cairan terdeteksi, tanda adanya gangguan seperti penyumbatan selang atau masalah pada katup, sistem akan memberikan peringatan berupa bunyi buzzer pada perangkat infus dan notifikasi pada *website*.

Deteksi gangguan ini penting untuk memastikan keselamatan pasien selama proses infus.

### 2.1.2 Sistem Monitoring

Sistem monitoring merupakan sebuah sistem terintegrasi yang dirancang untuk memantau, mengawasi, dan mengumpulkan data secara otomatis dalam waktu nyata (*real-time*) [6][7]. Sistem monitoring berbasis IoT menggunakan sensor dan perangkat yang terhubung untuk memantau berbagai operasi secara otomatis dengan mengumpulkan dan menganalisis data secara *real-time*. Cara kerja sistem monitoring berbasis IoT dimulai dengan sensor-sensor IoT yang dipasang pada objek atau lingkungan tertentu untuk mengumpulkan data *real-time*, seperti suhu, kelembaban, tekanan, lokasi, dan parameter lainnya [8].

Dalam konteks sistem monitoring berbasis IoT, perangkat dan sensor berperan dalam memantau operasi dengan mengumpulkan serta menganalisis data secara *real-time*. Implementasi sistem otomatis dan monitoring infus di rumah sakit berbasis IoT memanfaatkan sensor-sensor seperti *load cell* untuk mengukur laju tetesan infus dan volume cairan infus yang tersisa dalam botol. Data volume cairan ini dikumpulkan dan dikirimkan secara otomatis ke platform IoT melalui jaringan nirkabel untuk diproses dan ditampilkan di *website* yang dapat diakses oleh tenaga medis. Dengan ini, tenaga medis dapat memantau volume cairan infus pasien dari jarak jauh secara *real-time*, tanpa perlu melakukan pengecekan manual secara langsung.

Sistem ini juga dirancang untuk mendeteksi anomali seperti volume cairan infus yang hampir habis. Saat mendekati batas minimal, sistem akan memberikan notifikasi otomatis dalam bentuk peringatan melalui bunyi buzzer pada perangkat infus dan melalui notifikasi pada *website*. Notifikasi ini memungkinkan tenaga medis merespons dengan cepat dan segera mengganti infus sebelum cairan habis sepenuhnya. Selain itu, sistem ini mencatat riwayat penggunaan infus seperti volume cairan yang digunakan dan durasi pemakaian, yang dapat dianalisis lebih lanjut untuk evaluasi perawatan pasien atau keperluan pengambilan keputusan medis. Dengan penggunaan teknologi IoT ini, proses monitoring menjadi lebih

efisien, akurat, dan dapat mengurangi potensi kesalahan manusia dalam memantau serta mengganti infus pasien di rumah sakit.

### 2.1.3 Internet of Things (IoT)

*Internet of Things* (IoT) adalah jaringan dari perangkat fisik yang saling terhubung melalui internet, dimana setiap perangkat dilengkapi dengan sensor, perangkat lunak, dan teknologi lain yang memungkinkan pertukaran data secara otomatis tanpa memerlukan campur tangan manusia. Konsep ini memungkinkan berbagai perangkat untuk berkomunikasi satu sama lain, sehingga menciptakan ekosistem yang lebih cerdas dan terintegrasi. Perangkat IoT dapat mencakup berbagai objek seperti alat rumah tangga hingga perangkat industri, yang semuanya terhubung dalam jaringan internet.

Pada implementasinya, IoT memanfaatkan teknologi seperti sensor, konektivitas nirkabel, dan *cloud computing* untuk mengumpulkan, menyimpan, dan memproses data secara *real-time*. Hal ini memungkinkan pemantauan dan pengendalian jarak jauh serta memberikan wawasan berbasis data yang lebih mendalam. IoT tidak hanya meningkatkan efisiensi dan otomatisasi, tetapi juga berpotensi menciptakan inovasi dalam berbagai sektor dengan memanfaatkan data yang dihasilkan dari interaksi antar perangkat.

*Internet of Things* (IoT) dalam bidang kesehatan merujuk pada jaringan perangkat medis yang saling terhubung melalui internet untuk mengumpulkan, mengirim, dan menganalisis data kesehatan pasien secara *real-time* [9]. Teknologi ini memfasilitasi pemantauan dan pengendalian perangkat medis secara otomatis dan jarak jauh. IoT dalam kesehatan mencakup perangkat *wearable* seperti monitor detak jantung, hingga sistem otomatisasi peralatan medis seperti infus yang terhubung melalui jaringan, memungkinkan tenaga medis mendapatkan akses terhadap data pasien dengan lebih cepat dan akurat.

Dengan menggunakan IoT, proses diagnosis dan pemantauan kesehatan menjadi lebih efisien, karena data dapat dikumpulkan secara kontinu dan otomatis tanpa

campur tangan manual. IoT juga memudahkan interaksi antarperangkat medis dan sistem elektronik di rumah sakit, sehingga dapat meminimalisir kesalahan dan meningkatkan respons terhadap kondisi kritis pasien. Selain itu, IoT memungkinkan pengambilan keputusan berbasis data yang *real-time*, yang sangat penting dalam situasi darurat [10].

#### 2.1.4 Sistem Infus

Infus cairan intravena (*intravenous fluids infusion*) adalah pemberian sejumlah cairan kedalam tubuh, melalui sebuah jarum, kedalam sebuah pembuluh vena (pembuluh balik)

untuk menggantikan kehilangan cairan atau zat-zat makanan [11][12].

Dalam pemberian tetes cairan infus didasarkan atas :

1. Kebutuhan cairan (*advice* oleh dokter).
2. Faktor Tetesan .

Dalam faktor tetes ini dibagi menjadi 2 yaitu faktor tetes makro dan mikro

Makro : Dewasa (20 tetes = 1 ml), atau

Makro : Dewasa (15 tetes = 1 ml)

Mikro : Anak-anak ( 60 tetes = 1 ml)

3. Waktu Pemberian.

**Rumus menghitung tetesan infus[13] :**

$$\boxed{\text{Jumlah tetesan per menit (TPM)} = \frac{V \times FT}{t(\text{jam}) \times 60 \text{ menit}}}$$

Keterangan : TPM = Tetesan per menit

T = Lama pemberian infus (jam)

V = Kebutuhan cairan (ml)

FT = Faktor tetes atau jumlah tetes per 1 ml

Sistem infus merupakan perangkat medis yang digunakan untuk mengalirkan cairan, obat-obatan, atau nutrisi langsung ke dalam pembuluh darah pasien melalui tetesan yang dikontrol.

Perangkat ini berperan penting dalam menjaga kestabilan kondisi pasien yang memerlukan asupan cairan secara terus-menerus.

Secara umum, sistem infus terbagi menjadi dua jenis, yaitu sistem manual yang memerlukan pengawasan langsung oleh tenaga medis, dan sistem berbasis IoT yang memungkinkan pemantauan serta pengendalian secara otomatis.

Dalam konteks sisa infus, penghitungan persentase volume cairan infus yang tersisa menggunakan rumus berikut:

$$\text{Sisa Cairan Infus} = \left( \frac{\text{Sisa Cairan Infus}}{\text{Kapasitas Total}} \right) \times 100\%$$

Rumus ini didasarkan pada konsep dasar matematika untuk menghitung persen, yaitu membandingkan bagian tertentu dengan total keseluruhan, kemudian mengalikan hasilnya dengan 100 untuk mendapatkan nilai dalam bentuk persentase [14].

Keterangan:

- **Sisa Cairan Infus:** Volume cairan infus yang masih ada di dalam kantong pada saat pengukuran dilakukan, yang dapat diperoleh menggunakan sensor berat seperti load cell.
- **Kapasitas Total:** Volume cairan infus awal yang diisi ke dalam kantong pada saat pemasangan pertama.

Dengan menggunakan rumus ini, sistem dapat mengkonversi data volume cairan infus ke dalam bentuk persentase, sehingga lebih mudah dipahami dan dipantau oleh tenaga medis.

Berikut ini akan dijelaskan lebih lanjut mengenai sistem infus manual dan sistem infus berbasis IoT.

### 2.1.5 Sistem Infus Manual

Sistem infus manual adalah metode konvensional dimana tenaga medis, seperti perawat, bertanggung jawab atas pengaturan dan pemantauan aliran cairan infus secara langsung. Proses ini melibatkan penyesuaian laju tetesan infus melalui regulator mekanis pada selang infus untuk mencapai laju aliran yang sesuai dengan kebutuhan pasien.

Laju aliran ini ditentukan berdasarkan kondisi medis pasien, dan perawat harus secara berkala memeriksa apakah tetesan infus berjalan sesuai rencana. Selain itu, perawat juga harus memantau volume cairan yang tersisa dalam kantong infus dan memastikan bahwa tidak terjadi masalah seperti sumbatan atau

aliran yang berhenti. Ketika volume cairan hampir habis, perawat perlu mengganti kantong infus secara manual. Pengawasan yang konstan diperlukan untuk mencegah kesalahan dalam laju aliran cairan yang dapat berdampak negatif pada kondisi pasien. Kelemahan dari sistem ini adalah ketergantungan pada intervensi manusia yang berpotensi menyebabkan keterlambatan dalam respons, terutama jika terjadi masalah mendadak.

#### 2.1.6 Sistem Infus Berbasis IoT

Sistem infus berbasis *Internet of Things* (IoT) adalah solusi modern yang dirancang untuk meningkatkan efisiensi dan akurasi dalam pemantauan serta pengontrolan infus secara *real-time*. Dengan memanfaatkan teknologi IoT, sistem ini dilengkapi dengan sensor *load cell* yang dapat mengukur jumlah cairan yang tersisa dalam kantong infus. Data dari sensor *load cell* ini kemudian dikirimkan ke platform berbasis web yang memungkinkan tenaga medis untuk mengakses informasi tersebut dari jarak jauh, tanpa perlu melakukan pengecekan manual di ruang pasien. Hal ini sangat membantu dalam mengurangi risiko keterlambatan penggantian infus yang dapat berdampak pada kesehatan pasien, terutama dalam situasi dimana tenaga medis terbatas.

Sistem ini dirancang untuk mengatasi keterbatasan sistem infus manual, yang memerlukan pemantauan fisik dan sering kali menjadi tantangan di rumah sakit dengan jumlah pasien yang banyak. Dengan IoT, perawat dapat dengan mudah memantau kondisi infus secara efisien melalui perangkat yang terhubung dengan internet. Selain itu, sistem juga dapat memberikan notifikasi secara otomatis ketika volume cairan hampir habis atau terjadi masalah pada aliran infus. Ini meningkatkan keamanan pasien dan memastikan perawatan yang lebih cepat dan optimal.

Dengan penerapan IoT, kualitas pelayanan kesehatan di rumah sakit dapat meningkat, menjadikan proses perawatan lebih responsif, terintegrasi, dan modern.

## 2.2 Perangkat Keras (*Hardware*)

Pada sub bab ini akan dijelaskan mengenai perangkat keras (*hardware*) yang digunakan dalam pengembangan tugas akhir ini. Perangkat keras yang

dimanfaatkan meliputi berbagai komponen penting seperti sensor *load cell*, sensor photodiode dan LED, ESP 32, Motor Servo , Keypad, Buzzer, modul I2C LCD, Modul HX711, dan *PCB (Printed Circuit Board)*. Setiap perangkat keras ini memiliki peran khusus dalam mendukung sistem otomatisasi dan monitoring infus berbasis IoT. Berikut penjelasan lengkapnya mengenai fungsi dan kegunaan dari masing-masing perangkat keras tersebut.

### 2.2.1 ESP32

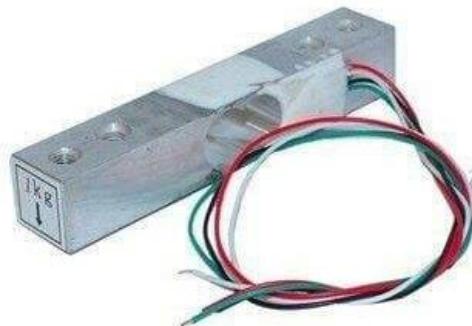


**Gambar 2. 1 ESP32**  
(Sumber : [15])

ESP32 adalah mikrokontroler canggih yang dikembangkan oleh Espressif Systems dan dirancang untuk mendukung berbagai aplikasi *Internet of Things* (IoT). Mikrokontroler ini menggunakan prosesor Xtensa LX6 yang dapat memiliki satu atau dua inti dengan kecepatan hingga 240 MHz, sehingga menawarkan kinerja yang tinggi. ESP32 dilengkapi dengan memori SRAM hingga 520 KB dan mendukung flash eksternal, memungkinkan pengolahan data yang kompleks.

Salah satu keunggulan utama ESP32 adalah konektivitasnya, yang mencakup Wi-Fi 802.11 b/g/n untuk komunikasi nirkabel dengan internet dan Bluetooth 4.2 serta BLE (*Bluetooth Low Energy*) untuk koneksi dengan perangkat lain. Selain itu, ESP32 menyediakan berbagai antarmuka komunikasi seperti UART, SPI, dan I2C, serta banyak GPIO (*General Purpose Input/Output*) untuk menghubungkan sensor dan aktuator. Dengan fitur-fitur ini, ESP32 menjadi solusi yang fleksibel dan efisien untuk aplikasi seperti sistem otomatisasi, pengendalian perangkat pintar, dan pengumpulan data jarak jauh.

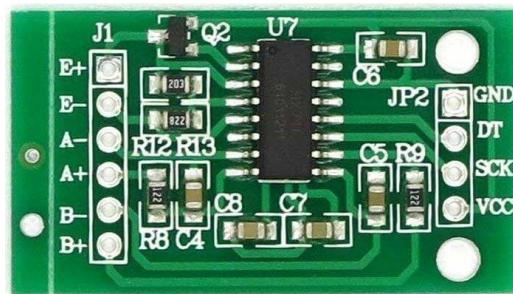
### 2.2.2 Sensor Load Cell



**Gambar 2. 2 Sensor Load Cell**  
(Sumber : [16])

Load cell adalah sensor yang mengukur berat atau beban dengan mengubah tekanan menjadi sinyal listrik. Dalam sistem monitoring infus, load cell dipasang pada kantong infus untuk mengukur berat cairan secara langsung. Perubahan berat ini dihitung untuk mengetahui sisa cairan dalam infus. Dengan teknologi IoT, data dari load cell bisa dikirim ke perangkat pemantauan jarak jauh, sehingga tenaga medis dapat memantau kondisi infus secara otomatis. Jika cairan hampir habis, sistem akan memberi peringatan melalui buzzer dan mengirimkan notifikasi ke website, memudahkan tindakan cepat untuk mengganti infus.[17].

### 2.2.3 Modul HX711

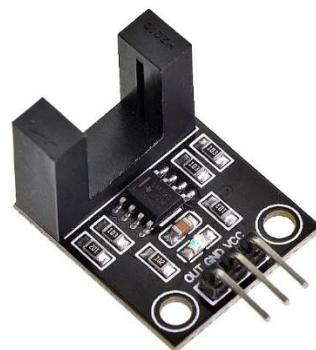


**Gambar 2. 3 Modul HX711**  
(Sumber :[18])

Modul HX711 adalah modul amplifier yang digunakan pada rangkaian timbangan berat digital. Cara kerja dari modul HX711 ini yaitu sebagai tegangan pada *load cell* yang bekerja dengan cara menghubungkannya ke mikrokontroler, kita dapat merubah resistansi dari *load cell* sensor [19]. Modul HX711 menawarkan

berbagai kemudahan, termasuk tiga nilai penguatan, yaitu 32, 64, dan 128. Untuk skala digital ini, penguatan yang digunakan adalah 128 dengan cara menghubungkan keluaran sensor ke saluran A pada modul. Modul ini menggunakan antarmuka "dua kabel", yaitu untuk sinyal jam dan data dalam proses komunikasi. Untuk memudahkan pembacaan data dari HX711, juga tersedia pustaka (library) yang dapat digunakan [20].

### 2.2.4 Photodioda



**Gambar 2. 4 Photodioda**  
(Sumber :[21])

Photodioda adalah jenis dioda yang sensitif terhadap cahaya, dimana resistansinya akan berubah ketika menerima intensitas cahaya. Ketika itu terjadi, photodioda mengalirkan arus listrik secara forward, mirip dengan dioda pada umumnya. Sebagai salah satu jenis sensor yang peka terhadap cahaya (*photodetector*), arus yang dihasilkan oleh photodioda akan membentuk hubungan linear dengan intensitas cahaya yang diterimanya, dan umumnya berbanding lurus dengan *power density* (dp). Dalam sistem infus, LED dan sensor photodioda digunakan untuk mengukur kecepatan tetesan cairan infus dengan akurat. Sensor ini berfungsi dengan mendeteksi perubahan intensitas cahaya setiap kali tetesan cairan melewati jalur infus, sehingga memungkinkan pengukuran yang presisi terhadap laju aliran cairan [22]. Sensor photodioda memiliki peran penting dalam memonitor kecepatan tetesan infus secara otomatis. Data yang dihasilkan oleh sensor ini dapat digunakan untuk memastikan pasien menerima jumlah cairan yang sesuai dengan pengaturan, tanpa memerlukan pengawasan manual yang terus-menerus dari tenaga medis.

### 2.2.5 Keypad



**Gambar 2. 5 Keypad**  
(Sumber :[23])

Keypad berfungsi sebagai antarmuka antara pengguna dan perangkat elektronik melalui sistem HMI (*Human Machine Interface*). Dalam desain keypad, 16 tombol disusun dalam matriks, sehingga hanya membutuhkan sedikit pin input untuk menghubungkan keypad ke mikrokontroler, seperti Arduino. Hal ini membuat

pengkodean dan pengkabelan lebih efisien, mengurangi kompleksitas dan biaya proyek.

Pembacaan pada keypad menggunakan teknik *scanning*, di mana satu baris aktif pada setiap siklus dan kolom diperiksa untuk mendeteksi tombol yang ditekan. Metode ini meningkatkan kecepatan respon dan mengurangi kesalahan pembacaan, menjadikan keypad komponen andal dalam berbagai aplikasi elektronik dan otomasi.[11].

## 2.2.6 Motor Servo



**Gambar 2. 6 Motor Servo**  
(Sumber :[24])

Motor servo adalah mekanisme *closed-loop* yang menggunakan umpan balik posisi untuk mengatur gerakan dan posisi akhirnya. Input untuk kontrol terdiri dari sinyal, baik analog maupun digital, yang menunjukkan posisi yang diinginkan untuk poros output. Motor ini dilengkapi dengan *encoder* posisi yang memberikan umpan balik tentang posisi dan kecepatan. Dalam aplikasi yang paling sederhana, hanya posisi yang diukur. Posisi output saat ini dibandingkan dengan posisi yang diperintahkan, yang berfungsi sebagai input eksternal untuk pengontrol. Jika ada perbedaan antara posisi aktual dan posisi yang diinginkan, sinyal kesalahan dihasilkan, yang memicu motor untuk berputar ke arah yang tepat untuk menyelaraskan poros output dengan posisi yang diperintahkan [6][12].

### 2.2.7 Buzzer



**Gambar 2. 7 Buzzer**  
(Sumber :[25])

Buzzer adalah perangkat elektronik yang menghasilkan suara atau nada sebagai respons terhadap sinyal listrik. Buzzer biasanya terdiri dari elemen piezoelektrik atau elektromagnetik yang bergetar saat dialiri arus listrik, menghasilkan gelombang suara. Ada dua jenis utama buzzer : aktif dan pasif. Buzzer aktif dapat menghasilkan suara dengan hanya menerima tegangan DC, sementara buzzer pasif memerlukan sinyal suara atau frekuensi untuk menghasilkan nada. Buzzer sering digunakan dalam berbagai aplikasi, seperti alarm, notifikasi, dan indikator dalam perangkat elektronik, untuk memberikan peringatan atau menarik perhatian pengguna.

### 2.2.8 Modul I2C LCD

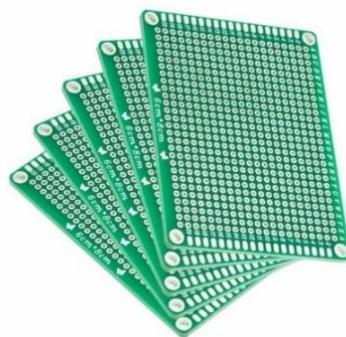


**Gambar 2. 8 Modul I2C LCD**  
(Sumber : [26])

LCD (*Liquid Crystal Display*) digunakan untuk menampilkan data, seperti volume cairan infus dan status sistem secara *real-time*, memudahkan tenaga medis

memantau informasi tanpa perangkat tambahan. I2C (*Inter-Integrated Circuit*) adalah protokol komunikasi yang menghubungkan berbagai perangkat, seperti sensor, motor servo, dan LCD, ke mikrokontroler menggunakan hanya dua kabel, mengurangi jumlah kabel dan mempermudah pengaturan rangkaian. Kedua komponen ini sangat berguna dalam sistem monitoring infus berbasis IoT, meningkatkan efisiensi dan kinerja sistem [20].

### 2.2.9 PCB (Printed Circuit Board)



**Gambar 2. 9** Printed Circuit Board  
(Sumber :[27])

PCB (*Printed Circuit Board*) adalah papan sirkuit yang digunakan untuk mendukung dan menghubungkan komponen elektronik secara elektrik dan mekanik. PCB terbuat dari bahan isolator, seperti fiberglass, yang dilapisi dengan jalur konduktif, biasanya terbuat dari tembaga. Jalur-jalur ini dirancang untuk menghubungkan komponen seperti resistor, kapasitor, dan chip mikrokontroler, sehingga membentuk rangkaian listrik yang diinginkan. PCB dapat memiliki berbagai bentuk dan ukuran, dan biasanya digunakan dalam perangkat elektronik seperti komputer, *smartphone*, dan peralatan rumah tangga. Keuntungan utama dari PCB adalah kemampuannya untuk mengurangi ukuran dan kompleksitas rangkaian elektronik, meningkatkan keandalan, dan memudahkan proses produksi.

### 2.3 Perangkat Lunak (*Software*)

Pada sub bab ini akan dijelaskan mengenai perangkat lunak (*software*) yang digunakan dalam pengembangan tugas akhir ini. Perangkat lunak yang digunakan meliputi Arduino IDE, MySQL, Visual Studio Code, dan Postman. Berikut penjelasan lengkapnya mengenai fungsi dan kegunaan dari masing-masing perangkat keras tersebut.

#### 2.3.1 Arduino IDE



**Gambar 2. 10** Arduino IDE  
(Sumber :[28])

Pemrograman untuk Arduino dilakukan menggunakan bahasa pemrograman Arduino melalui Arduino IDE. Bahasa ini berbasis pada *Processing*, sebuah bahasa pemrograman sumber terbuka yang terkenal. Sintaks Arduino mirip dengan C++, namun dengan tambahan beberapa fungsi dan konstanta yang telah ditentukan. Program yang ditulis dalam Arduino IDE disebut *sketch*. Sebuah *sketch* Arduino minimal terdiri dari dua fungsi: *setup()* dan *loop()*. Fungsi *setup()* dipanggil sekali ketika program dijalankan, biasanya setelah papan dihidupkan atau di-*reset*. Fungsi ini umumnya digunakan untuk menginisialisasi variabel, perpustakaan, atau pin I/O. Setelah *setup()* selesai dipanggil, fungsi *loop()* akan dieksekusi berulang kali hingga papan dimatikan atau di-*reset* kembali [29].

### 2.3.2 MySQL



**Gambar 2. 11 MySQL**  
**(Sumber :[30])**

MySQL adalah perangkat lunak pengelolaan basis data relasional yang menggunakan SQL sebagai bahasa utama. Sistem ini dirancang untuk memungkinkan akses, pengaturan, dan manipulasi data dengan efisien. Fungsi MySQL adalah memastikan bahwa data tersimpan dengan benar dan dapat diambil kapan saja diperlukan. Cara kerja MySQL adalah dengan menerima dan menjalankan perintah SQL yang diberikan oleh pengguna. Setelah permintaan dikirimkan, sistem akan memproses kueri tersebut dan mengambil data yang dibutuhkan dari basis data, kemudian menampilkannya kepada pengguna [31].

### 2.3.3 Visual Studio Code



**Gambar 2. 12 Visual Studio Code**  
**(Sumber :[32])**

Visual Studio Code (VSCode) adalah editor kode sumber yang gratis, ringan, namun sangat kuat, yang dapat digunakan di desktop dan web.

Editor ini tersedia untuk berbagai platform, termasuk Windows, macOS, Linux, serta Raspberry Pi OS. VSCode menawarkan dukungan bawaan untuk JavaScript, TypeScript, dan Node.js, serta memiliki ekosistem ekstensi yang luas untuk mendukung berbagai bahasa pemrograman lainnya seperti C++, C#, Java, Python, PHP, dan Go [33].

#### 2.3.4 MQTT



Gambar 2. 13 MQTT  
(Sumber : [34])

MQTT (*Message Queuing Telemetry Transport*) adalah protokol komunikasi ringan yang dirancang untuk mentransfer data antar perangkat dalam jaringan, terutama pada aplikasi *Internet of Things* (IoT). Protokol ini menggunakan model *publish-subscribe*, di mana perangkat atau aplikasi (*publisher*) mengirim pesan ke broker, yang kemudian mendistribusikan pesan tersebut ke perangkat lain (*subscriber*) yang telah berlangganan topik tertentu. MQTT sangat efisien dalam penggunaan bandwidth rendah dan sumber daya terbatas, sehingga ideal untuk perangkat IoT yang membutuhkan komunikasi cepat dan andal, seperti pada sistem pengendalian jarak jauh atau pemantauan sensor.

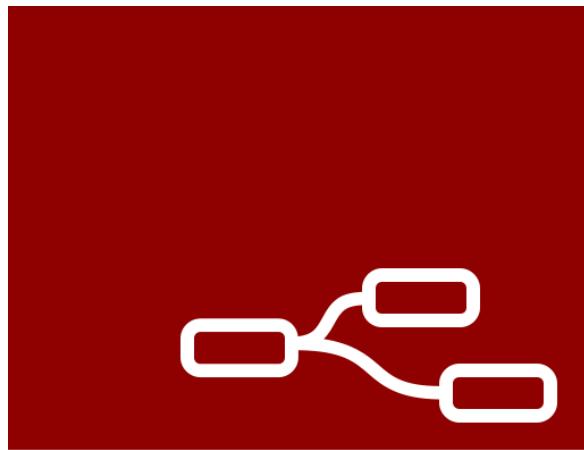
### 2.3.5 Mosquitto



Gambar 2. 14 Mosquitto  
(sumber :[35])

Mosquitto adalah salah satu implementasi *open-source* dari broker MQTT yang ringan dan mudah digunakan. Mosquitto berfungsi sebagai perantara yang mengelola komunikasi antara perangkat atau aplikasi yang menggunakan protokol MQTT. Dengan Mosquitto, perangkat yang bertindak sebagai publisher dapat mengirimkan pesan ke broker, yang kemudian mendistribusikan pesan tersebut ke perangkat lain yang bertindak sebagai subscriber berdasarkan topik yang dipilih. Mosquitto banyak digunakan dalam aplikasi *Internet of Things* (IoT) karena kemampuannya untuk mengelola komunikasi data secara efisien dengan minimal penggunaan sumber daya, dan juga mendukung berbagai fitur seperti keamanan (TLS/SSL), autentikasi, dan kualitas layanan (QoS).

### 2.3.6 Node-RED



Gambar 2. 15 Node-RED  
(Sumber :[36])

Node-RED adalah platform pengembangan perangkat lunak berbasis alur (*flow-based programming*) yang digunakan untuk membuat aplikasi dan solusi IoT dengan cara yang visual dan mudah. Dengan Node-RED, pengguna dapat membuat alur kerja (*flows*) yang menghubungkan berbagai perangkat, layanan, dan API tanpa memerlukan banyak penulisan kode. Node-RED menyediakan antarmuka grafis yang memungkinkan pengguna untuk menarik dan menghubungkan berbagai node yang mewakili perangkat, sensor, dan layanan untuk melakukan operasi seperti pemrosesan data, pengiriman pesan, atau pengendalian perangkat. Platform ini sering digunakan untuk integrasi berbagai sistem dan automasi dalam aplikasi IoT, serta mendukung berbagai protokol seperti MQTT, HTTP, dan WebSocket, yang memudahkan koneksi antar perangkat.

## 2.4 Related Work

Pada sub bab ini akan dijelaskan mengenai jurnal yang terkait dengan sistem yang akan dibangun. Berikut beberapa jurnal yang terkait dengan sistem yang akan dibangun.

### 2.4.1 Monitoring Cairan Infus Menggunakan Load Cell Berbasis IoT

Dalam penelitian ini, sistem yang dirancang bertujuan untuk memudahkan pemantauan volume infus serta aliran tetesan infus per menit dengan menggunakan teknologi *Internet of Things* (IoT). Sistem ini menggunakan sensor Load Cell untuk mengukur volume cairan infus yang tersisa dan sensor inframerah (*InfraRed*) untuk mendeteksi dan menghitung jumlah tetesan per menit. Kedua sensor ini diintegrasikan ke dalam sebuah platform web monitoring, yang memungkinkan perawat atau tenaga medis untuk memantau kondisi infus secara *real-time* melalui jaringan internet. Data yang dihasilkan oleh sensor Load Cell menunjukkan tingkat akurasi pengukuran hingga 99,8%, memastikan pengukuran volume cairan infus yang akurat. Sementara itu, sensor inframerah dapat menghitung tetesan dengan tepat, memberikan informasi yang esensial tentang kecepatan aliran infus.

Platform web yang dikembangkan untuk sistem ini memiliki kemampuan untuk memperbarui data secara berkala dengan jeda waktu antara 10 hingga 30 detik, tergantung pada kecepatan koneksi internet yang tersedia di fasilitas kesehatan. Sistem ini memungkinkan pemantauan jarak jauh, sehingga mengurangi beban perawat dalam memantau kondisi pasien secara manual, serta meningkatkan efisiensi dan akurasi dalam memberikan perawatan yang optimal. Adopsi teknologi seperti ini diharapkan dapat menurunkan risiko kesalahan medis terkait pengelolaan cairan infus serta memberikan pengalaman perawatan yang lebih aman bagi pasien [37].

### 2.4.2 Sistem Kontrol dan Monitoring Infus Berbasis NodeMCU

Infus merupakan bagian penting dari terapi cairan pada pasien rawat inap, dan pengawasan terhadap kondisi infus serta laju tetes infus secara berkala adalah tanggung jawab perawat. Jika infus habis, maka penggantian harus segera

dilakukan untuk menjaga kondisi pasien. Dalam upaya meningkatkan efisiensi pengawasan, diperlukan pengembangan sistem *prototype* yang memungkinkan pemantauan dan kontrol infus secara otomatis dari jarak jauh. Sistem ini dirancang untuk memantau dua aspek utama: status cairan infus (penuh atau hampir habis) dan laju tetes infus selama satu menit.

Untuk memantau status infus, sensor load cell digunakan untuk mengukur berat cairan infus. Sensor ini diintegrasikan dengan NodeMCU sebagai mikrokontroler yang bertugas memproses data dan mengirimkannya ke platform web. Melalui web tersebut, tenaga medis dapat memantau kondisi infus tanpa harus berada di dalam ruangan pasien. Selain itu, sistem ini juga dilengkapi dengan motor servo yang berfungsi untuk mengatur laju tetesan infus secara otomatis. Pengujian menunjukkan bahwa ketika berat infus berkurang hingga kurang dari 200 gram, sistem akan memberikan indikasi bahwa infus hampir habis dan memerlukan penggantian. Sedangkan untuk pengaturan laju tetes infus yang ideal (sekitar 20 tetes per menit), motor servo diatur pada sudut 24-26 derajat.

Hasil pengembangan sistem ini menunjukkan potensi besar untuk diterapkan di rumah sakit sebagai solusi pengawasan infus yang efisien dan akurat, mengurangi beban kerja perawat serta meminimalkan risiko keterlambatan penggantian infus yang dapat membahayakan pasien. Dengan akses *real-time* melalui platform web, tenaga medis bisa mengambil tindakan cepat ketika dibutuhkan, memastikan pasien menerima perawatan optimal [38].

#### **2.4.3 Alat Monitoring Tetesan Infus Menggunakan Web Berbasis ESP8266**

Proyek ini bertujuan untuk merancang dan mengembangkan sistem monitoring infus yang mampu mengawasi kondisi infus pasien secara *real-time* melalui web berbasis ESP8266 dan diprogram menggunakan Arduino IDE. Tujuan utamanya adalah menghindari keterlambatan penanganan, terutama pada saat cairan infus hampir habis, serta memastikan bahwa laju tetesan infus berjalan sesuai standar.

Proses pembuatan alat ini melibatkan beberapa langkah penting, seperti identifikasi dan analisis kebutuhan, perancangan blok diagram, pembuatan perangkat keras dan perangkat lunak, hingga pengujian alat untuk menilai kinerjanya.

Alat ini menggunakan NodeMCU ESP8266-E12 sebagai pengontrol utama, yang diprogram untuk mengintegrasikan seluruh sistem. Sebagai sensor pendeksi tetesan infus, digunakan modul sensor IR Obstacle Avoidance, yang bekerja dengan mendeksi tetesan infus melalui pantulan inframerah dari setiap tetesan yang jatuh. Informasi mengenai kondisi infus, seperti laju tetesan per menit dan status cairan infus (penuh, hampir habis, atau habis), dikirimkan ke platform web yang dapat diakses secara *online* oleh perawat atau petugas medis. Hal ini memungkinkan pemantauan jarak jauh tanpa harus memeriksa infus secara langsung di tempat pasien berada.

Pengujian alat ini dilakukan dengan menggunakan infus berukuran 500 ml dan infus set makro. Hasil pengujian menunjukkan bahwa sistem ini dapat berfungsi dengan baik, dengan tingkat keberhasilan mencapai 97,13%. Program yang ditulis menggunakan Arduino IDE mampu berintegrasi secara mulus dengan NodeMCU ESP8266, sehingga seluruh proses monitoring dan kontrol dapat berjalan dengan baik. Modul IR Obstacle Avoidance juga terbukti mampu mendeksi tetesan infus, meskipun terdapat tantangan dalam pengaturan jarak pantulan yang optimal terhadap objek.

Sistem ini menawarkan solusi komprehensif dalam memantau ketersediaan cairan infus, mendeksi ketika infus hampir habis, serta mendeksi adanya hambatan dalam aliran infus. Selain itu, sistem ini membantu meningkatkan efisiensi kerja perawat dengan menyediakan data yang dapat diakses secara *online*, sehingga mengurangi frekuensi pemeriksaan manual langsung pada pasien. Dengan demikian, sistem ini berpotensi besar untuk diterapkan dalam rumah sakit guna meningkatkan kualitas perawatan pasien dan mencegah potensi risiko yang disebabkan oleh keterlambatan penanganan infus [39].

#### 2.4.4 Sistem Monitoring dan Controlling Cairan Infus Berbasis Web

Penelitian ini bertujuan untuk mengatasi masalah tersebut dengan mengembangkan sistem monitoring cairan infus berbasis teknologi *Internet of Things* (IoT). Sistem ini memungkinkan tenaga medis memantau kondisi cairan infus secara *real-time* dari jarak jauh, menggunakan jaringan internet. Dengan memanfaatkan IoT, sistem secara terus-menerus mengirimkan data mengenai status cairan infus ke ruang perawat, sehingga mereka dapat lebih cepat merespons dan mengganti infus sebelum cairan habis. Hal ini memberikan keamanan lebih dalam proses pemantauan, dan membantu memastikan pergantian infus dilakukan dengan tepat waktu.

Pengujian sistem dilakukan melalui *blackbox testing*, yang menunjukkan bahwa sistem berfungsi sesuai dengan spesifikasi yang dirancang. Selain itu, pengujian *load time* menunjukkan bahwa dengan *bandwidth* 19,23 Mbps, data dapat dikirim dan diterima dalam waktu 1,00 detik, sedangkan dengan *bandwidth* 14,50 Mbps, waktu pengiriman meningkat menjadi 1,75 detik. Adapun rata-rata *delay* pengiriman data dari perangkat ke database untuk setiap tetesan infus adalah 68,38 detik, yang menunjukkan kecepatan komunikasi data yang masih dapat ditingkatkan, namun tetap cukup responsif untuk aplikasi di rumah sakit. Sistem ini diharapkan dapat memberikan kontribusi besar dalam meningkatkan efisiensi kerja tenaga kesehatan, khususnya dalam memantau kondisi pasien dengan infus [40].

#### 2.4.5 System Monitoring Infus Menggunakan HX711

Penelitian ini bertujuan untuk merancang sistem monitoring infus berbasis web yang menggunakan sensor HX711 dan Wemos untuk mengukur volume cairan infus dan mengirimkan data secara *real-time* ke sebuah situs web. Dengan adanya sistem ini, perawat dapat melihat kondisi volume infus yang tersisa dan lokasi ruangan pasien tanpa harus terus-menerus memeriksa langsung. Hal ini mengurangi frekuensi pemeriksaan manual dan membantu perawat dalam mengelola waktu lebih efisien, terutama di rumah sakit dengan banyak pasien.

Penggunaan sensor HX711 yang terintegrasi dengan Wemos memungkinkan pengukuran volume infus dengan akurasi yang baik. Data yang dikirimkan ke *website* dapat diakses dari mana saja oleh tenaga medis, sehingga mereka dapat segera mengambil tindakan jika volume infus mendekati batas habis. Inovasi ini diharapkan dapat meningkatkan kualitas pelayanan kesehatan, mengurangi risiko keterlambatan penggantian infus, dan mempermudah tugas perawat dalam pemantauan kondisi infus pasien [41].

## **BAB III**

### **ANALISIS DAN DESAIN SISTEM**

Pada bagian ini berisi uraian pelaksanaan Tugas Akhir untuk menjawab permasalahan yang ada secara sistematis.

#### **3.1 Analisis**

Pada sub bab bagian ini hal yang diamati atau dilakukan analisa kemudian diuraikan sebagai analisis masalah yang ada, analisis untuk pemecahan masalah, dan analisis kebutuhan sistem guna mengembangkan sistem yang menjadi solusi dari permasalahan yang ada.

##### **3.1.1 Analisis Masalah**

Pemantauan infus yang masih dilakukan secara manual memiliki beberapa kekurangan yang dapat mengurangi efisiensi dan kenyamanan dalam pelayanan kesehatan. Dalam metode ini, tenaga medis harus memeriksa infus pasien secara langsung untuk melihat sisa cairan dan mengatur laju tetesan. Proses ini memakan waktu dan tenaga, apalagi jika jumlah pasien yang perlu diawasi cukup banyak. Akibatnya, pemantauan manual menjadi tidak efisien dan mempengaruhi kemampuan tenaga medis dalam mengelola waktu mereka.

Selain itu, kondisi ini juga dapat mengurangi produktivitas tenaga medis. Waktu yang digunakan untuk memantau infus secara manual bisa saja lebih bermanfaat jika dialihkan ke tugas lain yang lebih mendesak atau berisiko tinggi, seperti penanganan pasien gawat darurat. Saat tenaga medis terfokus pada pemantauan infus, mereka bisa kehilangan kesempatan untuk menangani tugas lain yang membutuhkan perhatian segera.

Kenyamanan dan keamanan pasien juga dapat terganggu jika pemantauan infus tidak dilakukan dengan optimal. Misalnya, jika cairan infus habis dan tidak cepat diganti, pasien bisa merasa tidak nyaman dan bahkan berisiko mengalami masalah kesehatan lebih lanjut. Hal ini tentunya tidak ideal, terutama bagi pasien yang bergantung pada infus untuk mendapatkan obat atau nutrisi penting.

Terakhir, risiko terlambatnya pergantian infus juga menjadi perhatian. Jika infus tidak diganti tepat waktu, hal ini bisa mengganggu perawatan pasien. Dalam beberapa kasus, keterlambatan ini dapat memperburuk kondisi pasien, terutama jika infus merupakan satu-satunya sumber asupan atau obat yang dibutuhkan.

### 3.1.2 Analisis Pemecahan Masalah

Untuk mengatasi masalah-masalah dalam pemantauan infus secara manual, diperlukan solusi yang dapat meningkatkan efisiensi pemantauan, produktivitas tenaga medis, serta keamanan dan kenyamanan pasien. Sistem infus otomatis berbasis IoT dirancang sebagai solusi yang efektif untuk mengatasi berbagai kelemahan yang ada.

Dengan memanfaatkan teknologi IoT, sistem ini memungkinkan pemantauan infus secara *real-time* melalui sensor-sensor yang terhubung langsung ke perangkat pusat. Sensor seperti load cell digunakan untuk mengukur sisa cairan infus, sementara sensor tetesan memantau laju tetesan infus. Data yang dihasilkan oleh sensor ini dikirimkan secara langsung ke sistem pusat atau *website* sederhana, sehingga tenaga medis dapat memantau kondisi infus pasien tanpa harus melakukan pemeriksaan manual yang berulang. Pemantauan *real-time* ini tidak hanya meningkatkan efisiensi tetapi juga memungkinkan tenaga medis untuk mengawasi banyak pasien sekaligus, sehingga tugas pemantauan menjadi lebih praktis dan akurat.

Selain itu, sistem infus otomatis ini dilengkapi dengan fitur notifikasi otomatis yang memberikan peringatan saat volume cairan infus mendekati batas minimal atau jika ada gangguan pada laju tetesan. Peringatan ini dapat berupa bunyi buzzer pada alat dan juga notifikasi pada *website*, yang secara cepat memberitahukan tenaga medis untuk segera mengambil tindakan. Dengan adanya notifikasi ini, risiko keterlambatan pergantian infus dapat diminimalkan, sehingga proses terapi pasien berjalan tanpa hambatan yang bisa menurunkan efektivitas perawatan.

Sistem otomatis ini juga mendukung tenaga medis dalam mengelola waktu mereka dengan lebih efisien. Berkat pemantauan jarak jauh dan notifikasi otomatis, tenaga medis dapat mengurangi frekuensi pemeriksaan manual infus dan lebih fokus pada tugas lain yang lebih kritis atau mendesak. Ini berpotensi meningkatkan produktivitas tenaga medis dan memungkinkan mereka memberikan perawatan yang lebih baik kepada pasien-pasien yang membutuhkan perhatian khusus.

Keamanan dan kenyamanan pasien pun menjadi lebih terjamin dengan penerapan sistem infus otomatis ini. Dengan infus yang terpantau secara terus-menerus, pasien tidak perlu khawatir jika infus mereka hampir habis tanpa diketahui oleh tenaga medis. Hal ini memberikan rasa aman dan nyaman karena perawatan mereka berjalan dengan lancar dan tanpa gangguan yang disebabkan oleh keterlambatan penggantian infus. Selain itu, pemantauan otomatis ini juga membantu meminimalkan kesalahan yang mungkin terjadi dalam pemeriksaan manual, yang terkadang terjadi dalam lingkungan rumah sakit yang sibuk. Dengan sistem otomatis, data infus diperoleh secara akurat dan dikirimkan langsung ke sistem, memastikan bahwa perawatan yang diterima pasien selalu konsisten dan tepat.

### 3.1.3 Analisis Kebutuhan Sistem

Pada saat melakukan pengujian pada Tugas Akhir ini, maka dibutuhkan beberapa perangkat keras dan perangkat lunak, berikut ini adalah kebutuhan perangkat keras dan perangkat lunak yang dibutuhkan dalam membangun implementasi sistem otomatis dan monitoring infus berbasis IoT. Bagian ini menguraikan beberapa kebutuhan utama dalam membangun sistem yang dikerjakan, yaitu sebagai berikut:

#### 3.1.3.1 Analisis Kebutuhan Perangkat Keras (*Hardware*)

Perangkat keras yang dipakai untuk membangun sistem yang dikembangkan pada Tugas Akhir ini adalah sebagai berikut :

**Tabel 3. 1** Analisis Kebutuhan Perangkat Keras (*Hardware*)

No.	Nama Perangkat	Spesifikasi	Keterangan
1.	ESP 32	<ul style="list-style-type: none"> <li>• Prosesor : Dual-core atau single-core</li> </ul>	ESP32 adalah mikrokontroler yang

No.	Nama Perangkat	Spesifikasi	Keterangan
		<p>Xtensa LX6, dengan kecepatan hingga 240 MHz.</p> <ul style="list-style-type: none"> <li>• Memori : SRAM hingga 520 KB.</li> <li>• FLASH : 2MB (max 64MB)</li> <li>• Tegangan : 2.2V – 3.6V</li> </ul>	dikembangkan oleh Espressif System dan merupakan penerus dari ESP8266. ESP32 memiliki modul WiFi dan Bluetooth terintegrasi, sehingga sangat cocok untuk membuat sistem aplikasi <i>Internet of Things</i> (IoT)
2.	Load Cell	<ul style="list-style-type: none"> <li>• Kapasitas : 1kg</li> <li>• Tegangan Operasional: 5V hingga 12V (tergantung pada jenis load cell)</li> </ul>	Load Cell digunakan untuk menentukan volume sisa cairan dalam kantong infus
3.	Modul HX711	<ul style="list-style-type: none"> <li>• Tipe : Analog-to-Digital Converter (ADC) khusus untuk load cell</li> <li>• Tegangan Operasional : 2.6V hingga 5.5V</li> <li>• Sinyal Output : Sinyal digital (output serial)</li> <li>• Kecepatan Sampling : 0 Hz atau 80 Hz,</li> </ul>	Modul analog-to-digital converter yang digunakan untuk mengkonversi sinyal dari load cell sensor

No.	Nama Perangkat	Spesifikasi	Keterangan
		tergantung pada konfigurasi	
4.	Sensor Photodiode	<ul style="list-style-type: none"> <li>• Open Circuit Voltage : Tegangan maksimum yang dihasilkan saat photodiode tidak terhubung ke beban adalah 0.39 V.</li> <li>• Reverse Breakdown Voltage : Tegangan maksimum yang dapat diterima tanpa merusak perangkat adalah 32 V.</li> <li>• Reverse Light Current: Arus yang dihasilkan saat cahaya mengenai photodiode adalah 40 <math>\mu</math>A.</li> <li>• Reverse Dark Current : Arus yang mengalir melalui photodiode dalam kegelapan adalah 5 nA.</li> </ul>	Sensor Photodiode digunakan untuk memantau aliran tetesan infus dan menghitung jumlah tetesan per menit.
5.	Modul keypad 4x4	<ul style="list-style-type: none"> <li>• Maximum Voltage across EACH</li> </ul>	keypad digunakan sebagai alat input bagi pengguna untuk memasukkan

No.	Nama Perangkat	Spesifikasi	Keterangan
		<p>SEGMENT or BUTTON : 24V</p> <ul style="list-style-type: none"> <li>• Maximum Current through EACH SEGMENT or BUTTON : 30mA</li> <li>• Maximum Current through EACH SEGMENT or BUTTON : 30 mA</li> <li>• Maximum Operating Temperature : 0°C to +50°C</li> </ul>	pengaturan awal infus, seperti volume faktor laju tetesan dan waktu
6.	Motor Servo	<ul style="list-style-type: none"> <li>• Tegangan Operasional : Umumnya +5V</li> <li>• Torsi : Bervariasi, biasanya sekitar 2.5 kg/cm</li> <li>• Kecepatan Operasi : Sekitar 0.1 s untuk 60°</li> <li>• Rentang Rotasi : Umumnya 0° - 180°, tetapi bisa lebih</li> </ul>	Motor Servo digunakan untuk menyesuaikan posisi berdasarkan data TPM (Volume, Tekanan, Waktu) yang diterima, sehingga dapat mengontrol laju aliran infus.

No.	Nama Perangkat	Spesifikasi	Keterangan
7.	Buzzer	<ul style="list-style-type: none"> <li>• Tegangan Terukur : 6V DC</li> <li>• Tegangan Operasional : 4-8V DC</li> <li>• Arus Terukur : &lt;30mA</li> <li>• Frekuensi Resonansi : ~2300 Hz</li> <li>• Tipe Suara : Continuous Beep</li> </ul>	Buzzer diaktifkan sebagai alarm ketika terjadi masalah, seperti tidak ada tetesan, volume infus hampir habis, atau infus telah habis
8.	LCD HD44780	<ul style="list-style-type: none"> <li>• Tegangan Operasi : 4.7V hingga 5.3V</li> <li>• Konsumsi Arus : 1mA tanpa lampu latar</li> <li>• Jumlah Baris : 4 baris</li> <li>• Jumlah Baris : 4 baris</li> <li>• Mode Operasi : 8-bit dan 4-bit</li> <li>• Pilihan Lampu Latar : Tersedia dengan lampu latar Hijau dan Biru</li> </ul>	LCD digunakan untuk menampilkan informasi penting seperti TPM dan berat infus.
9.	I2C (Inter Integrated Circuit)	<ul style="list-style-type: none"> <li>• Jumlah Kabel : Menggunakan dua kabel: SDA (Serial</li> </ul>	I2C merupakan sebuah protokol komunikasi serial yang digunakan untuk

No.	Nama Perangkat	Spesifikasi	Keterangan
		<p>Data Line) dan SCL (Serial Clock Line)</p> <ul style="list-style-type: none"> <li>• Protokol Komunikasi : Komunikasi dua arah, master-slave</li> <li>• Kecepatan: Standar: 100 kHz, Fast Mode: 400 kHz, Fast Mode Plus: 1 MHz, High-Speed: 3.4 MHz</li> <li>• Pengalamatan : Setiap perangkat memiliki alamat 7-bit atau 10-bit</li> </ul>	menghubungkan berbagai komponen dalam sistem.

### 3.1.3.2 Analisis Kebutuhan Perangkat Lunak (*Software*)

Tabel 3. 2 Analisis Kebutuhan Perangkat Lunak (*Software*)

No.	Perangkat Lunak	Deskripsi
1.	Arduino IDE 2.3.3	Arduino IDE digunakan untuk membuat dan mengunggah kode ke mikrokontroler seperti Arduino Mega atau ESP32 yang mengontrol sensor dalam sistem infus otomatis berbasis IoT. Melalui kode yang diunggah dari Arduino IDE, sistem dapat mengukur berat cairan infus dan menghitung tetesan

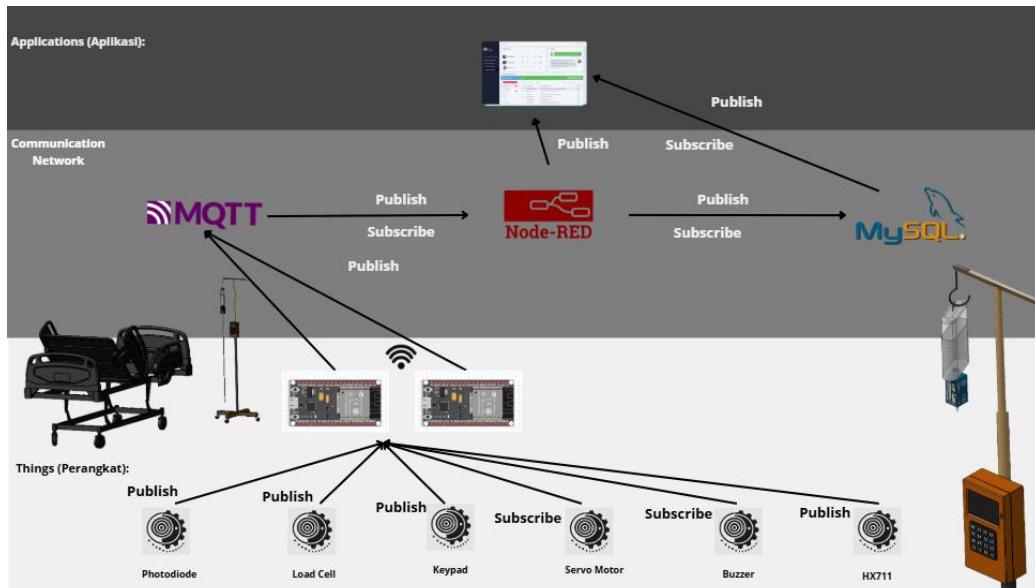
No.	Perangkat Lunak	Deskripsi
		infus secara otomatis, sehingga perawat tidak perlu melakukan pemantauan manual.
2.	MySQL 5.6	MySQL bertindak sebagai tempat penyimpanan data infus yang dikumpulkan dari sensor, seperti berat cairan dan jumlah tetesan per menit. Dengan MySQL, data ini bisa tersimpan dengan aman dan bisa diakses sewaktu-waktu oleh sistem monitoring, sehingga tenaga medis dapat memantau kondisi infus dari jarak jauh melalui aplikasi berbasis web.
3.	VSCode 1.95	VSCode membantu dalam mengembangkan aplikasi web yang digunakan untuk menampilkan data infus secara <i>real-time</i> kepada tenaga medis. Selain antarmuka, VSCode juga memfasilitasi pengembangan <i>backend</i> yang terhubung dengan database MySQL dan broker MQTT, sehingga data infus dapat dikirim dan ditampilkan dengan lancar di halaman monitoring.
4.	Node-RED 3.1	Node-RED digunakan untuk mengatur alur data dari sensor ke aplikasi monitoring secara visual. Pada sistem infus otomatis ini, Node-RED memudahkan integrasi antar perangkat dan memastikan data infus seperti volume cairan dan laju tetesan dapat diteruskan ke aplikasi monitoring secara <i>real-time</i> tanpa hambatan.
5.	MQTT 3.1	MQTT adalah protokol komunikasi utama yang memungkinkan pengiriman data infus dari mikrokontroler ke server atau aplikasi monitoring. Dengan MQTT, data kondisi infus seperti jumlah tetesan dan sisa cairan bisa dikirim dengan cepat, memungkinkan perawat atau tenaga medis

No.	Perangkat Lunak	Deskripsi
		mendapatkan informasi langsung saat cairan infus hampir habis.
6.	Mosquitto 5.0	Mosquitto bertindak sebagai broker MQTT yang mengatur lalu lintas data antara sensor dan sistem monitoring. Dalam sistem infus otomatis berbasis IoT ini, Mosquitto memastikan bahwa setiap data infus dapat dikirim ke aplikasi monitoring dengan stabil, sehingga tenaga medis bisa mengawasi kondisi infus secara tepat waktu.

### 3.2 Desain

Pada sub bagian ini menjelaskan bagaimana rancangan dari sistem yang dibuat dengan melibatkan penggunaan *software* dan *hardware* atau komponen yang membangun sistem.

#### 3.2.1 Topologi Sistem Pemantauan Infus Otomatis



Gambar 3. 1 Topologi Sistem Pemantauan Infus Otomatis

Berdasarkan gambar topologi di atas, ada 3 (tiga) bagian utama komponen yang saling terhubung yaitu *Things* (Perangkat), *Communication Network* (Jaringan Komunikasi), dan *Applications* (Aplikasi).

### 1. *Things* (Perangkat)

Lapisan ini terdiri dari perangkat keras yang berfungsi sebagai sensor dan aktuator, di mana semuanya terhubung langsung ke mikrokontroler **ESP32**. Perangkat keras yang digunakan meliputi:

- **Photodiode:** Berfungsi mendekripsi jumlah tetesan cairan infus.
- **Load Cell:** Digunakan untuk mengukur berat cairan infus dalam kantong.
- **HX711:** Modul yang membaca data dari load cell untuk dikirimkan ke ESP32.
- **Keypad:** Memungkinkan input dari pengguna untuk mengatur parameter infus, seperti jumlah tetesan per menit.
- **Servo Motor:** Mengontrol kecepatan aliran cairan infus berdasarkan perhitungan dari sensor.
- **Buzzer:** Memberikan notifikasi bunyi ketika volume cairan hampir habis atau terjadi kondisi darurat.

Semua perangkat ini terhubung langsung ke ESP32, yang berfungsi sebagai pengumpul data dari sensor dan pengontrol aktuator.

### 2. *Communication Network* (Jaringan Komunikasi)

Lapisan komunikasi bertanggung jawab untuk pertukaran data antara perangkat fisik dan aplikasi berbasis cloud. Komunikasi dilakukan melalui protocol MQTT (*Message Queuing Telemetry Transport*) dengan arsitektur sebagai berikut:

#### **ESP32:**

- Mempublikasikan data dari sensor (photodiode, load cell) ke broker MQTT.
- Subscribe instruksi atau parameter yang dikirim dari aplikasi melalui broker MQTT.

#### **Broker MQTT**

- Bertindak sebagai penghubung utama untuk mengelola data dari perangkat keras ke aplikasi.
- Data yang dipublikasikan oleh ESP32 diteruskan ke Node-RED untuk pengolahan lebih lanjut dan ke MySQL untuk penyimpanan.

### **Node-RED**

- Memproses data dari broker MQTT dan mengatur alur kerja sistem, seperti mengirim notifikasi ke aplikasi jika terjadi kondisi tertentu misalnya, infus hampir habis.
- Node-RED juga dapat mengatur instruksi kembali ke ESP32 untuk mengontrol perangkat keras.

### **MySQL Database**

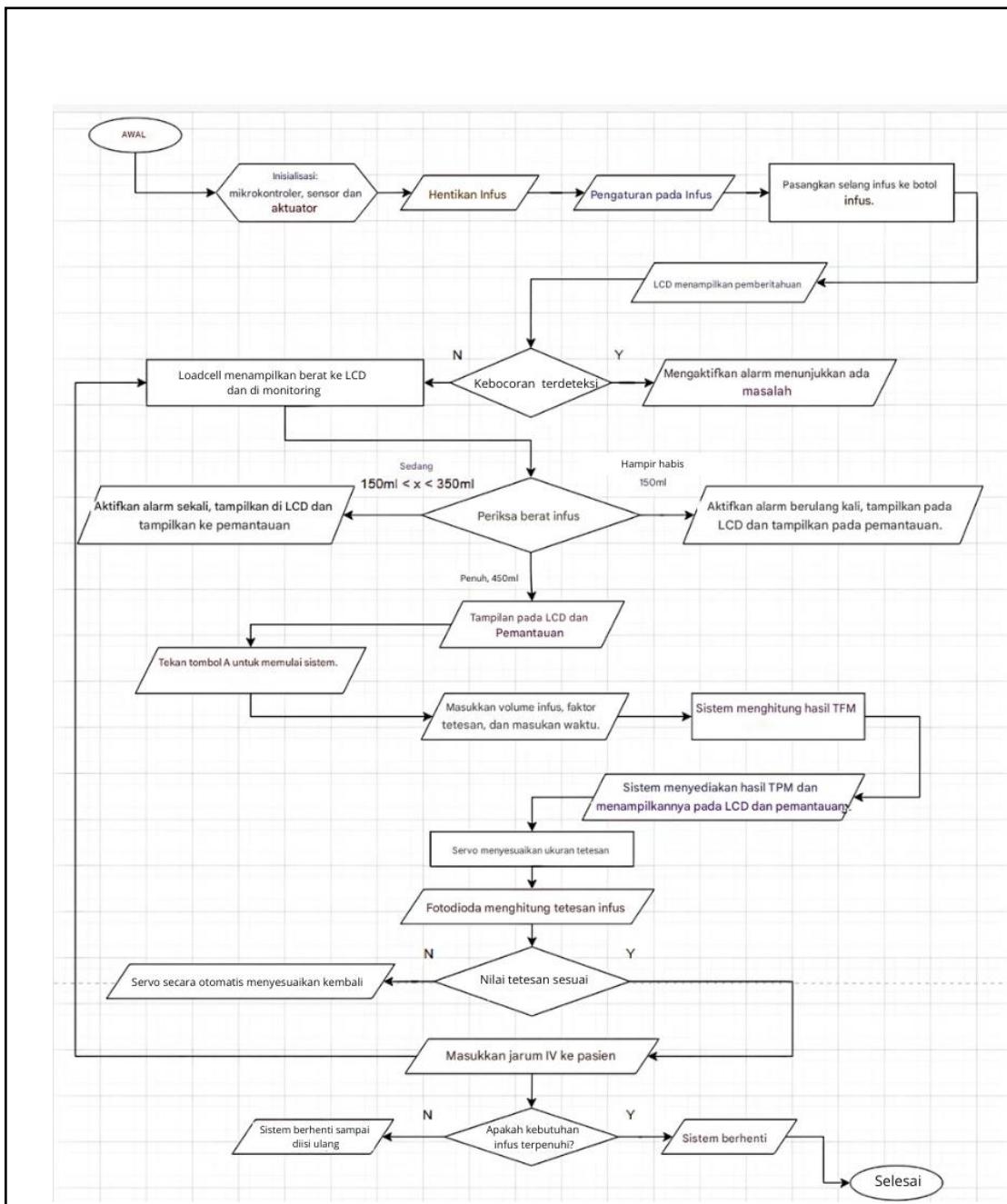
- Menyimpan data yang dipublikasikan dari Node-RED untuk kebutuhan analisis, monitoring, dan pelaporan.

### **3. *Applications* (Aplikasi)**

Lapisan ini merupakan antarmuka pengguna dan aplikasi berbasis web yang memanfaatkan data dari sistem untuk memantau dan mengontrol infus secara real-time. Fungsionalitas aplikasi meliputi:

- **Monitoring dan Notifikasi:** Menampilkan status infus, seperti jumlah tetesan, berat cairan, dan kondisi sistem (normal, hampir habis, dsb.), serta memberikan notifikasi ketika volume infus hampir habis atau terjadi kondisi darurat.
- **Pengaturan Parameter :** Memungkinkan pengguna untuk mengatur parameter infus, seperti kecepatan aliran tetesan, melalui aplikasi. Parameter ini akan diteruskan melalui broker MQTT ke ESP32 untuk dijalankan oleh perangkat keras.

### 3.2.2 Desain Flowchart Sistem Infus



Gambar 3. 2 Desain Flowchart Sistem Infus

*Flowchart* di atas menggambarkan proses otomatisasi dan pemantauan sistem infus berbasis IoT yang mencakup langkah-langkah berikut:

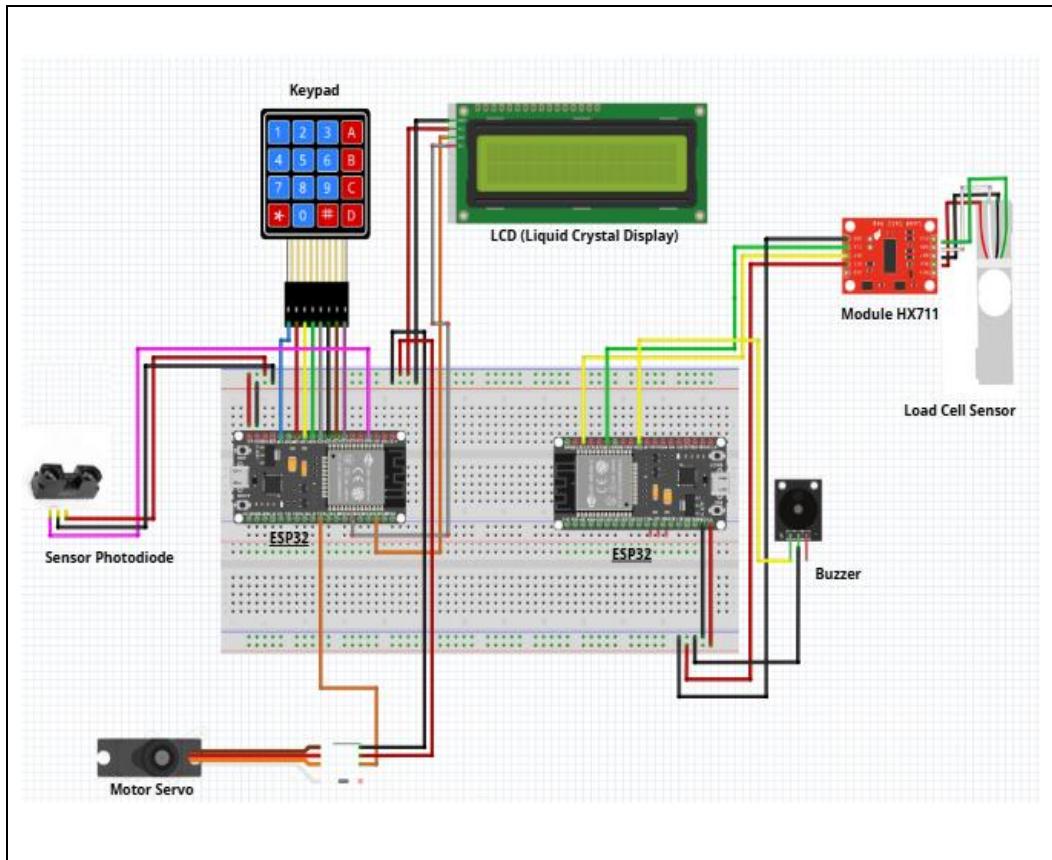
1. Awal : Sistem infus otomatis dimulai.
2. Inisialisasi : Sistem melakukan inisialisasi perangkat keras dan perangkat lunak, termasuk sensor dan komponen lainnya, untuk mempersiapkan perangkat.
3. Hentikan Infus : Infus dihentikan sementara untuk melakukan pengaturan sesuai kebutuhan.
4. Setting : Pengaturan awal infus dilakukan, diikuti dengan pengaktifan alat dan proses pemantauan.
5. Buka Selang Infus : Selang infus dibuka untuk memastikan cairan mengalir dengan lancar dan tidak ada udara yang terjebak di dalam selang.
6. Periksa Tetesan : Sistem memeriksa apakah ada tetesan infus.
  - Jika tidak ada tetesan, sistem akan mengaktifkan alarm untuk menandakan masalah (misalnya, infus macet atau selang tertutup).
  - Jika ada tetesan, sistem melanjutkan ke langkah berikutnya.
7. Load Cell Mengukur Berat Infus : Load cell mengukur berat cairan untuk menentukan sisa cairan yang ada di kantong infus.
8. Keputusan Berat Infus : Berdasarkan berat cairan, sistem akan menampilkan notifikasi pada LCD dan sistem pemantauan.
  - Jika berat lebih dari 300 mL (penuh), sistem akan menampilkan notifikasi bahwa infus penuh.
  - Jika berat antara 150 mL dan 300 mL (sedang), alarm akan aktif dan notifikasi "Sedang" ditampilkan.
  - Jika berat kurang dari 150 mL (hampir habis), alarm akan aktif dan notifikasi "Hampir Habis" ditampilkan.
9. Request Pengiriman Data TPM : Sistem meminta data terkait volume, tekanan, dan waktu pengisian infus yang dibutuhkan oleh pasien.
10. Hitung TPM : Sistem melakukan perhitungan berdasarkan data yang diterima untuk menentukan Tetesan per Menit (TPM).

11. Tampilkan TPM ke LCD dan Monitoring : Hasil perhitungan TPM ditampilkan pada LCD dan sistem monitoring.
12. Servo Menyesuaikan Berdasarkan TPM : Servo akan menyesuaikan posisi sesuai dengan nilai TPM yang dihitung.
13. Cek Kesesuaian dengan TPM : Sistem memeriksa apakah posisi servo sudah sesuai dengan nilai TPM.
  - Jika ya, data baru disimpan.
  - Jika tidak, penyesuaian servo akan diulang.
14. Cek Apakah Infus Habis : Sistem memeriksa apakah infus sudah habis.
  - Jika infus belum habis, sistem akan kembali memeriksa tetesan infus.
  - Jika infus habis, alarm akan aktif, dan notifikasi akan dikirim ke sistem pemantauan bahwa infus telah habis dan selesai.
15. Selesai : Proses pemantauan infus selesai, dan sistem kembali ke status awal.

Dengan proses ini, sistem dapat memantau dan mengelola infus pasien secara otomatis dan efisien, memastikan ketepatan waktu dan volume infus, serta memberikan peringatan jika ada masalah.

### 3.2.3 Desain Hardware

Desain *hardware* pada implementasi sistem otomatis dan monitoring infus berbasis IoT mengacu pada proses merancang dan mengembangkan komponen fisik yang diperlukan untuk menjalankan sistem tersebut. *Hardware* ini terdiri dari berbagai perangkat yang bekerja bersama untuk mengukur, memantau, dan mengontrol aliran infus serta mengirimkan data ke sistem monitoring berbasis IoT. Berikut desain *hardware* yang dirancang dalam implementasi sistem otomatis dan monitoring infus :



Gambar 3. 3 Desain Hardware

Berdasarkan gambar diatas, sistem ini dirancang untuk memonitor dan mengontrol cairan infus secara otomatis menggunakan berbagai komponen elektronik yang saling terhubung. Komponen yang digunakan antara lain ESP32, keypad, LCD (Liquid Crystal Display), sensor photodiode, Load Cell Sensor dengan Module HX711, Buzzer, dan Motor Servo . Berikut penjelasan rinci setiap komponen dan fungsinya:

## 1. ESP32 (Dua Unit)

- Berfungsi sebagai mikrokontroler utama yang mengintegrasikan semua sensor, modul, dan aktuator dalam sistem.
- ESP32 mendukung komunikasi Wi-Fi dan Bluetooth untuk pengiriman data ke perangkat lain atau server.

## 2. Keypad

- Digunakan untuk menginput data TPM yaitu, volume infus, faktor tetes, dan waktu yang dibutuhkan

## 3. LCD (*Liquid Crystal Display*)

- Menampilkan informasi , seperti jumlah tetesan cairan, berat infus, status sistem, atau peringatan.

## 4. Sensor Photodiode

- Digunakan untuk mendeteksi jumlah tetesan cairan infus

## 5. Load Cell Sensor dengan Module HX711

- Load cell mengukur berat cairan infus secara *real-time*.
- Modul HX711 adalah penguat sinyal dan ADC (*Analog-to-Digital Converter*) yang menghubungkan load cell ke ESP32.

## 6. Buzzer

- Berfungsi sebagai alat pemberi peringatan suara jika terjadi kondisi tertentu, seperti volume cairan infus hampir habis

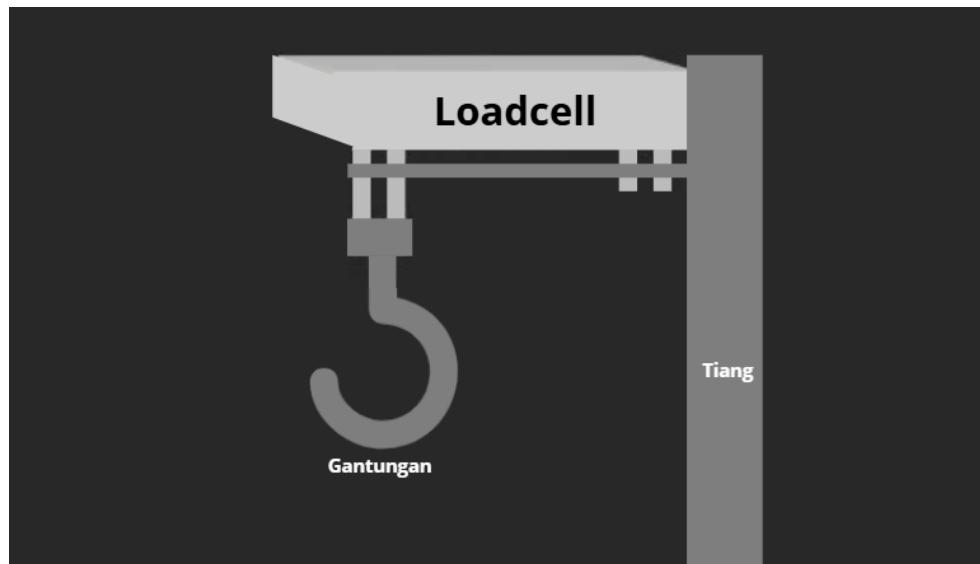
## 7. Motor servo

- Digunakan untuk mengatur tetesan infus sesuai TPM yang sudah diinput

Sistem ini bekerja dimulai dengan pengguna yang memasukkan data atau pengaturan jumlah tetesan cairan infus per menit melalui keypad. Data tersebut diproses oleh mikrokontroler ESP32 untuk mengatur seluruh sistem. Sensor photodiode mendeteksi jumlah tetesan cairan infus secara *real-time*, sedangkan sensor load cell, yang terhubung melalui modul HX711, mengukur berat cairan infus. Kedua data ini dikirimkan ke ESP32 untuk diproses lebih lanjut. Hasil pengukuran dan status sistem, seperti jumlah tetesan cairan per menit dan sisa berat cairan infus, ditampilkan pada layar LCD sebagai informasi bagi pengguna.

Berdasarkan data yang diterima, ESP32 mengendalikan motor servo untuk mengatur aliran cairan infus agar sesuai dengan pengaturan awal. Sistem ini dirancang untuk memastikan infus berjalan sesuai pengaturan, memberikan peringatan tepat waktu, dan meningkatkan efisiensi serta keamanan dalam penggunaan infus.

### 3.2.4 Perancangan Penempatan Loadcell



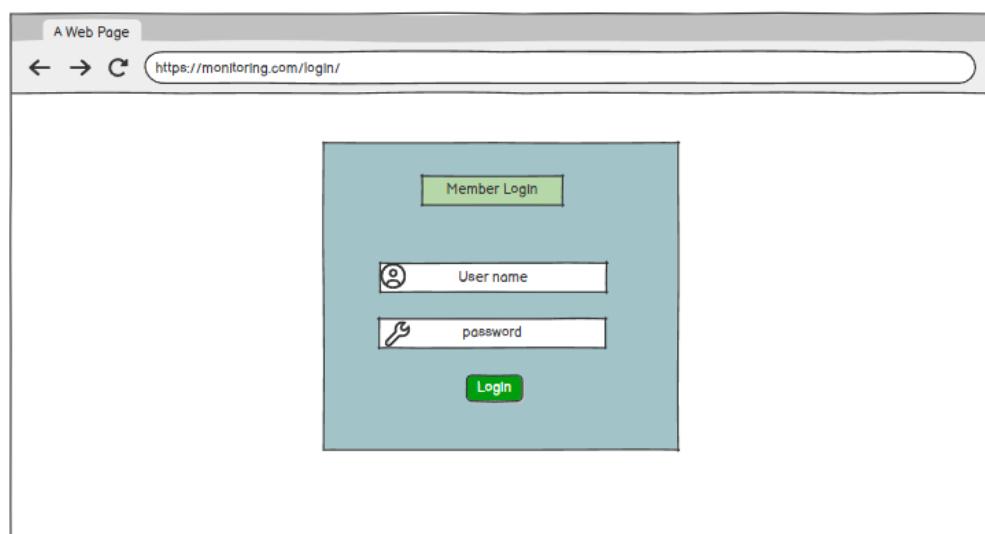
Gambar 3. 4 Perancangan Penempatan Loadcell

Desain di atas menunjukkan penempatan load cell untuk mengukur berat. Load cell dipasang di bagian atas struktur utama dan berfungsi sebagai sensor utama untuk mendekripsi perubahan berat. Sebuah gantungan yang terhubung langsung dengan load cell digunakan untuk menggantung benda, seperti kantong infus, sehingga beratnya dapat diukur dengan akurat. Tiang penyangga memastikan stabilitas struktur sekaligus melindungi load cell dari gangguan yang dapat mempengaruhi keakuratan pengukuran.

### 3.2.5 Desain Software

Desain *software* pada implementasi sistem otomatis dan monitoring infus berbasis IoT yang melibatkan desain *website* berfokus pada pembuatan antarmuka pengguna yang dapat memantau dan mengontrol sistem infus secara *real-time*. Berikut desain *hardware* yang dirancang dalam implementasi sistem otomatis dan monitoring infus :

#### 1. Tampilan Login



**Gambar 3. 5** Desain Tampilan Login

Halaman login adalah laman utama yang pertama kali ditemui saat mengakses web monitoring ini. Pada tahap ini, pengguna diwajibkan memasukkan username dan password yang valid sebelum diizinkan masuk ke dalam sistem. Proses ini bertujuan untuk memastikan bahwa hanya pihak yang berwenang yang dapat mengakses data di dalamnya.

Fitur login memiliki peran krusial dalam menjaga integritas dan keamanan data, terutama data pasien yang bersifat sensitif dan sangat berharga. Dengan penerapan sistem login yang baik, risiko akses tidak sah dan potensi kerusakan data dapat diminimalkan. Selain itu, autentikasi ini juga memungkinkan pemantauan aktivitas pengguna, termasuk siapa saja yang mengakses sistem dan waktu aksesnya, sehingga keamanan data pasien dapat tetap terjaga dengan optimal.

## 2. Desain Dashboard



**Gambar 3. 6** Desain Tampilan Dashboard

Tampilan *Dashboard* ini dirancang dengan tampilan awal yang dilengkapi navigasi intuitif, memudahkan pengguna untuk mengakses berbagai fitur yang tersedia. Di bagian utama, informasi pasien ditampilkan secara jelas, mencakup data penting seperti nomor kamar, tetesan infus per menit yang diperlukan, Tetes Per Menit (TPM) infus, serta volume infus yang tersedia.

Selain itu, halaman ini dilengkapi fitur pencarian yang memungkinkan pengguna menemukan data pasien tertentu dengan cepat dan efisien. Halaman utama ini dirancang untuk menyajikan informasi secara ringkas dan akurat, sehingga memudahkan tenaga medis dalam memantau kondisi pasien. Pada halaman ini, pengguna juga dapat menambahkan atau menghapus data pasien sesuai kebutuhan.

### 3. Desain Tampilan Data Riwayat Pasien

The screenshot shows a web application interface for managing patient history. At the top, there's a header bar with a 'Monitoring' button and navigation links for 'Home', 'Riwat Pasien', and 'Riwat Posi'. Below the header, a main title 'Riwayat Pasien' is centered. On the left, there are buttons for 'hapus' (delete) and a search bar with placeholder 'Q, search'. To the right of the search bar is a date input field 'dd/mm/yy'. Below these controls is a table displaying patient data. The table has columns for 'No', 'Ruang', 'Nomor Bed', 'Nama Pasien', and 'tanggal'. The data is as follows:

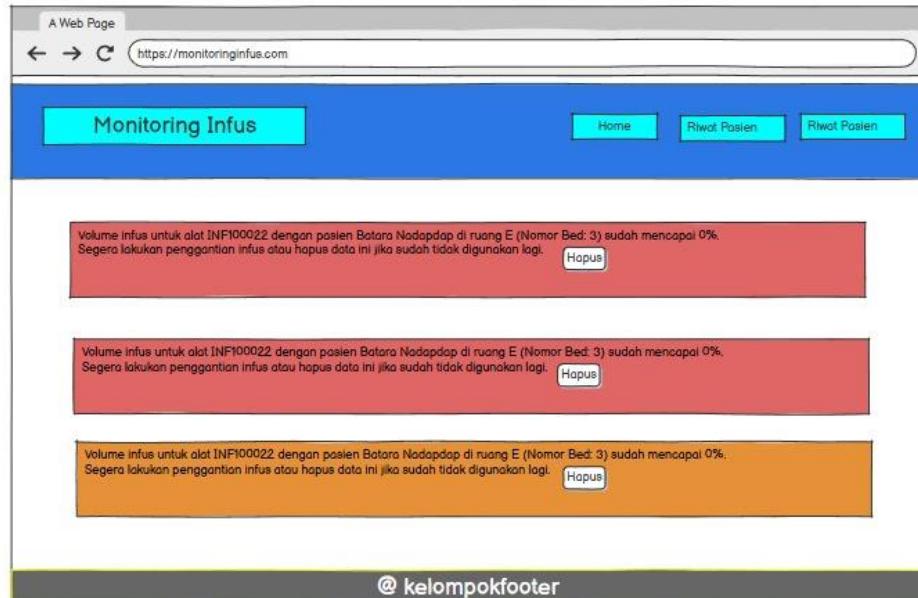
No	Ruang	Nomor Bed	Nama Pasien	tanggal
1	A1	01	Aldi	11/11/2024
2	A2	04	Sintong	12/11/2024
3	A1	03	Doli	16/11/2024
4	4	05	Refael	11/10/2024

At the bottom of the page, there's a footer bar with the text '@ kelompokfooter'.

**Gambar 3. 7** Desain Tampilan Data Riwayat Pasien

Halaman Riwayat pasien berisikan data pasien yang sudah pernah di rawat sebelumnya. Dimana disana akan ditampilkan ruangan , nomor bed , nama pasien dan tanggal. Data-data tersebut dapat diurutkan sesuai dengan tanggal baik mulai terbaru ataupun dari yang terlama. Disini juga memiliki fitur *search* (mesin pencari) sehingga mempermudah dalam mencari pasien yang ingin di cari. Terdapat pula fitur *hapus* (*delete*) yang memungkinkan penghapusan data secara permanen dari riwayat. Halaman ini dirancang untuk memberikan kemudahan dalam mengakses dan mengelola informasi riwayat pasien secara praktis dan terorganisir.

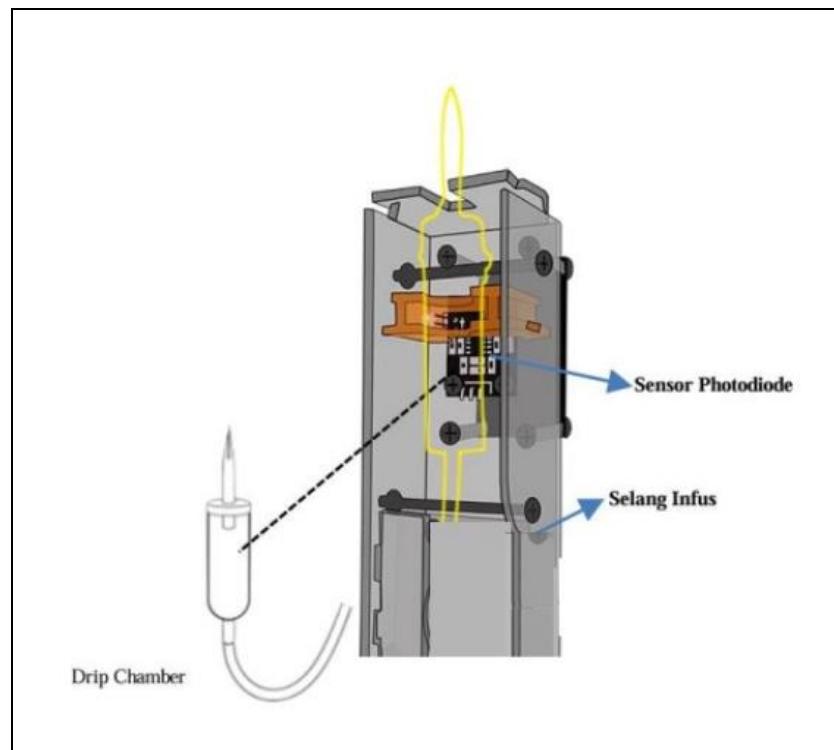
#### 4. Desain Tampilan Notifikasi



**Gambar 3. 8** Desain Tampilan Notifikasi

Tampilan notifikasi pada dashboard ini dirancang untuk memberikan informasi penting secara jelas dan langsung. Setiap notifikasi berisi pesan mengenai kondisi volume infus yang telah mencapai batas tertentu, seperti 0%, disertai identitas pasien, nomor alat infus, nomor bed, dan lokasi ruang. Pesan ini dilengkapi dengan peringatan agar pengguna segera mengganti infus atau menghapus data jika sudah tidak digunakan. Warna latar pada setiap notifikasi memberikan penekanan visual terhadap urgensi pesan. Selain itu, terdapat tombol Hapus di setiap notifikasi untuk mempermudah pengguna mengelola data yang sudah selesai ditangani.

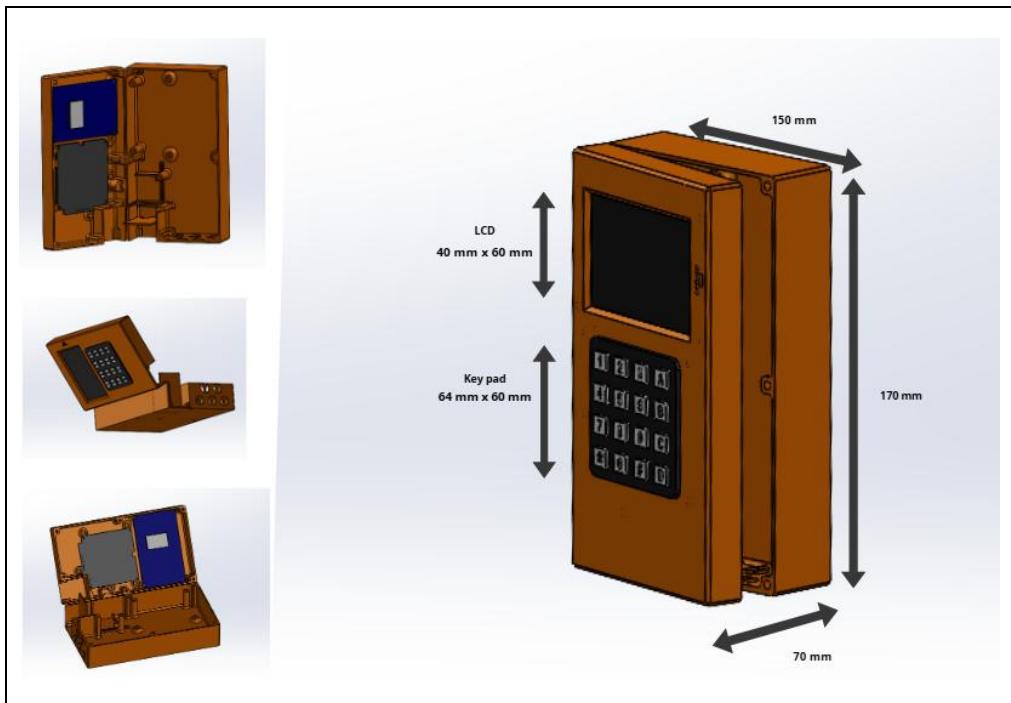
### 3.2.6 Desain Case Pengatur Tetesan



Gambar 3. 9 Desain Case Pengatur Tetesan

Gambar di atas memperlihatkan desain *case* pengatur tetesan pada sistem infus otomatis berbasis IoT. *Case* ini dirancang untuk melindungi dan menyusun seluruh komponen sistem secara rapi dan terintegrasi. Bagian dalam *case* dilengkapi dengan ruang untuk memasang sensor photodiode.

### 3.2.7 Desain Perancangan Case

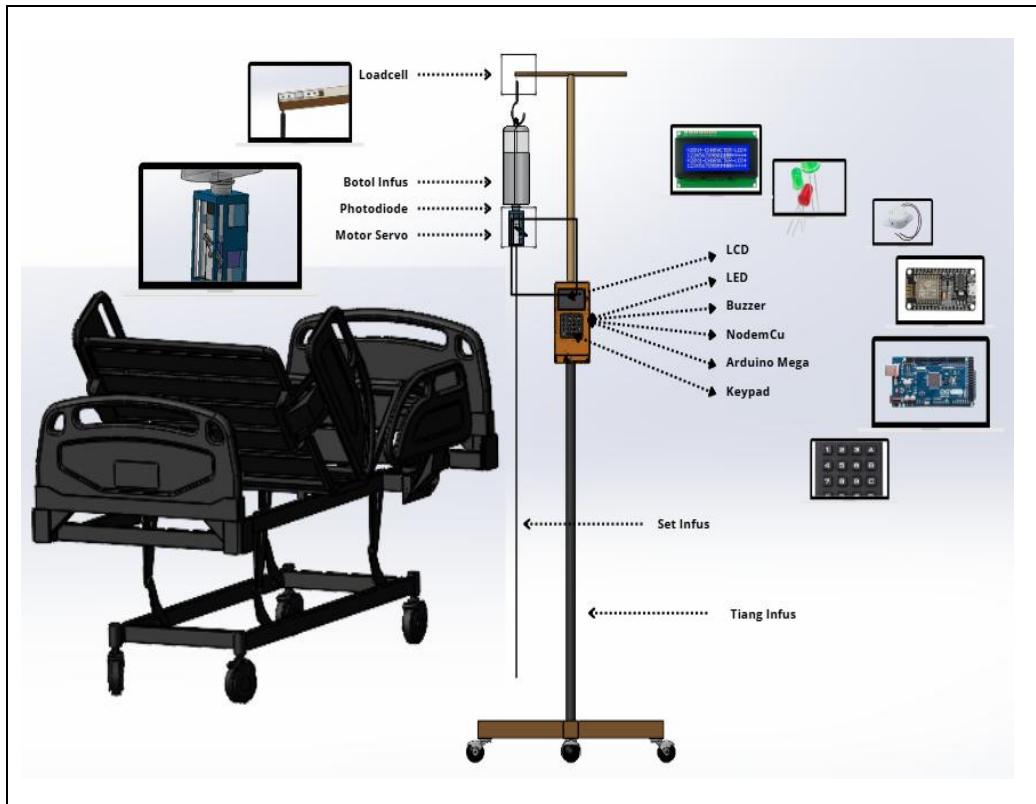


**Gambar 3. 10** Desain Case

1. Struktur Fisik: Case box ini memiliki desain kotak yang terbuat dari material yang kuat dan tahan lama, seperti logam atau plastik. Hal ini bertujuan untuk melindungi komponen internal dari kerusakan fisik, benturan, atau kondisi lingkungan yang tidak diinginkan.
2. Fungsi Perlindungan: Case box ini berfungsi sebagai tempat untuk menyimpan dan melindungi komponen-komponen elektronik, sensor, dan perangkat lain yang menyusun sistem infus otomatis. Hal ini penting untuk menjaga integritas dan keamanan komponen-komponen tersebut selama penggunaan.
3. Akses dan Kontrol: Case box ini juga dilengkapi dengan panel kontrol atau display yang memungkinkan pengguna untuk mengakses, memantau, dan mengendalikan sistem infus secara langsung. Desain ini memudahkan interaksi pengguna dengan sistem.

4. Portabilitas: Bentuk case box yang kompak dan mudah dibawa memungkinkan sistem infus otomatis ini untuk digunakan secara mobile atau dipindahkan ke lokasi yang berbeda sesuai kebutuhan.

### 3.2.8 Desain Fisik Sistem Infus Otomatis



Gambar 3. 11 Desain Fisik Sistem Infus

1. **Sensor photodiode:** Sensor photodioda terletak di bawah infus. Sensor ini berfungsi untuk menghitung jumlah tetesan infus yang mengalir, sehingga sistem dapat memantau dan mengatur laju aliran infus.
2. **Servo:** Servo juga terletak di bawah infus. Komponen ini bertugas untuk mengatur laju aliran infus berdasarkan data yang diterima dari sensor photodioda. Servo akan menyesuaikan bukaan katup atau mekanisme pengatur tetesan infus.
3. **Load Cell:** Load cell berada di atas tiang infus. Komponen ini berfungsi sebagai alat pengukur beban, yang dapat mendeteksi perubahan berat kantong infus. Data dari load cell dapat digunakan untuk memantau volume infus yang tersisa.

4. **Case Box:** Case box merupakan wadah atau tempat untuk menyimpan dan melindungi komponen-komponen elektronik, seperti mikrokontroler, dan modul komunikasi. Case box ini ditempatkan di tiang infus untuk memudahkan akses dan integrasi dengan komponen lainnya.

### 3.2.9 Kalibrasi Loadcell

Pengujian bertujuan untuk mengevaluasi tingkat kesalahan (*error*) dari pengukuran loadcell dengan membandingkan hasil pengukuran dari loadcell terhadap timbangan manual.

**Tabel 3. 3 Kalibrasi Loadcell**

No.	Timbangan Manual (gr)	Loadcell (gr)	Error (%)
1.	80	80	
2.	100	99	
3.	114	114	
4.	124	124	
5.	136	136	
6.	183	183	
7.	223	222	
8.	380	380	
9.	560	560	
10.	585	585	
Jumlah	2485	2483	0.08%

Tabel menunjukkan hasil pengujian pada 10 sampel beban dengan berbagai nilai berat. Kolom-kolom pada tabel meliputi:

1. **Timbangan Manual (gr):** Nilai berat referensi yang diperoleh dari timbangan manual.
2. **Loadcell (gr):** Nilai berat yang diukur oleh loadcell.

3. **Error (%)**: Persentase kesalahan antara timbangan manual dan loadcell, dihitung dengan rumus[42] :

$$\text{Error (\%)} = \left( \frac{\text{Nilai Timbangan Manual} - \text{Nilai Loadcell}}{\text{Nilai Timbangan Manual}} \right) \times 100\%$$

Berdasarkan hasil pengujian, terdapat 8 pengujian yang tidak memiliki error, yaitu pada nomor 1, 3, 4, 5, 6, 8, 9, dan 10. Hal ini menunjukkan bahwa pada sebagian besar pengujian, nilai yang diukur oleh loadcell sama persis dengan nilai timbangan manual, sehingga akurasinya mencapai 100%. Namun, terdapat 2 pengujian yang memiliki sedikit error, yaitu pada nomor 2 dan 7. Pada pengujian nomor 2, timbangan manual menunjukkan berat 100 gram, sedangkan loadcell mencatat 99 gram, menghasilkan error sebesar 1%. Sementara itu, pada pengujian nomor 7, timbangan manual menunjukkan berat 223 gram, dan loadcell mencatat 222 gram, dengan error sebesar 0,45%. Secara keseluruhan, rata-rata error dari semua pengujian hanya sebesar 0,08%, yang menunjukkan bahwa loadcell memiliki tingkat akurasi yang sangat tinggi.

#### Kode Program Kalibrasi Loadcell :

```

1. #include <EEPROM.h>
2. #include "HX711.h"
3.
4. const int LOADCELL_DOUT_PIN = 21;
5. const int LOADCELL_SCK_PIN = 22;
6. HX711 scale;
7.
8. float calibration_factor = 1095;
9. float weight_threshold = 0.5;
10.
11. float last_weight = 0.0;
12. const int buzzerPin = 5;

```

```
13.  
14. void setup() {  
15.   Serial.begin(115200);  
16.   scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);  
17.  
18.   Serial.println("Menjalankan sistem untuk menimbang berat dalam  
gram");  
19.  
20.   scale.set_scale();  
21.   scale.tare();  
22.   Serial.println("Tara selesai.");  
23. }  
24.  
25. void loop() {  
26.   if (scale.is_ready()) {  
27.     scale.set_scale(calibration_factor);  
28.     float weight = scale.get_units(10);  
29.  
30.     if (isnan(weight)) {  
31.       Serial.println("Kesalahan: Pembacaan berat menghasilkan  
NaN.");  
32.     weight = 0;  
33.   }  
34.  
35.   if (abs(weight) < weight_threshold) {
```

```
36.     weight = 0.0;  
37. }  
38.  
39. int rounded_weight = round(weight);  
40.  
41. if (rounded_weight != last_weight) {  
42.     String buzzerStatus = "Mati";  
43.  
44.     // Cek jika berat berada dalam rentang 2 - 100 gram  
45.     if (rounded_weight >= 2 && rounded_weight <= 100) {  
46.         digitalWrite(buzzerPin, HIGH); // Menyalakan buzzer  
47.         buzzerStatus = "Bunyi";  
48.     } else {  
49.         digitalWrite(buzzerPin, LOW); // Mematikan buzzer  
50.     }  
51.  
52.     Serial.print("Berat yang terdeteksi: ");  
53.     Serial.print(rounded_weight);  
54.     Serial.print(" gram - Buzzer: ");  
55.     Serial.println(buzzerStatus);  
56.  
57.     last_weight = rounded_weight;  
58. }  
59. } else {  
60.     Serial.println("HX711 tidak terdeteksi. Periksa koneksi.");  
61. }
```

```

62.

63. delay(1000);

64. }

```

### 3.2.10 Kalibrasi Motor Servo

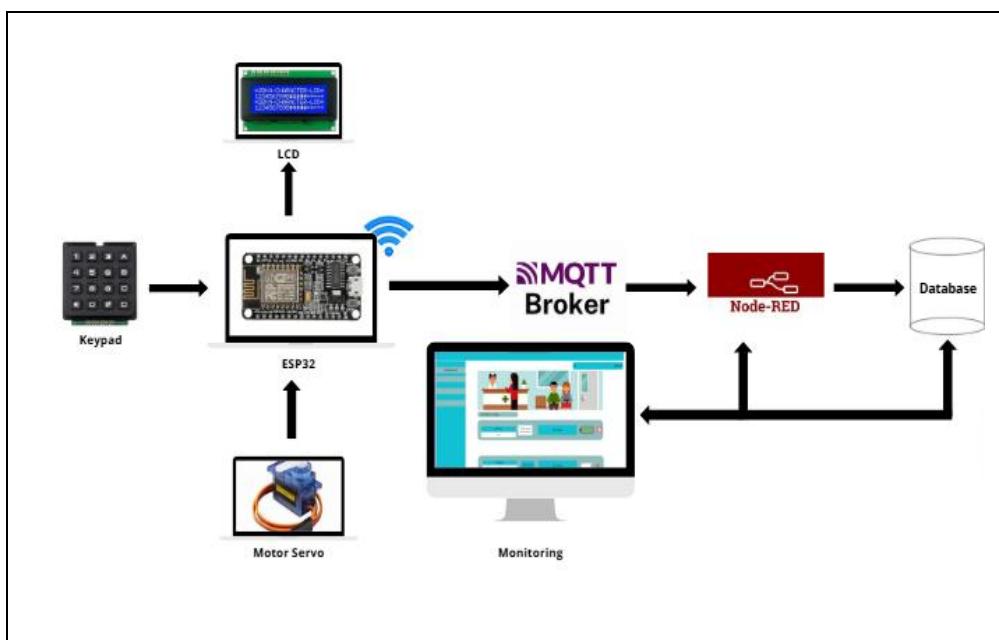
Kalibrasi motor servo dilakukan untuk menentukan hubungan antara sudut servo dengan jumlah tetesan cairan infus per menit (tpm). Proses ini bertujuan agar sistem dapat mengatur aliran cairan dengan akurasi tinggi sesuai kebutuhan. Kalibrasi dilakukan dengan mengatur sudut servo mulai dari  $19^\circ$  hingga  $31^\circ$  secara bertahap, dengan kenaikan  $1^\circ$  setiap kali pengujian. Pada setiap sudut, dilakukan lima kali pengujian untuk mencatat jumlah tetesan per menit.

**Tabel 3. 4** Kalibrasi Motor Servo

Pengujian Ke-	Input Sudut	Tetes per Menit				
		Menit ke-				
		1	2	3	4	5
1	$19^\circ$	58	57	57	57	58
2	$20^\circ$	56	55	55	55	54
3	$21^\circ$	53	53	53	53	52
4	$22^\circ$	51	52	51	52	51
5	$23^\circ$	42	42	43	43	43
6	$24^\circ$	39	39	39	39	39
7	$25^\circ$	32	34	33	34	34
8	$26^\circ$	27	28	28	28	28
9	$27^\circ$	25	25	25	26	25
10	$28^\circ$	21	20	20	20	20
11	$29^\circ$	19	19	20	19	19
12	$30^\circ$	17	17	17	17	17
13	$31^\circ$	15	16	15	15	15

Hasil kalibrasi menunjukkan bahwa jumlah tetesan cairan stabil pada setiap sudut, dengan variasi yang sangat kecil. Misalnya, pada sudut  $19^\circ$ , jumlah tetesan berkisar antara 57-58 tpm, sedangkan pada sudut  $30^\circ$ , jumlah tetesan konsisten di 17 tpm. Semakin besar sudut servo, semakin sedikit jumlah tetesan cairan yang keluar setiap menit. Selain itu, akurasi sistem terlihat dari variasi tetesan yang tidak melebihi  $\pm 2$  tetesan per menit pada setiap sudut. Dengan hasil ini, pengguna dapat dengan mudah menentukan sudut servo yang sesuai untuk mencapai target tetesan tertentu.

### 3.2.11 Skenario Pengujian Penginputan Data melalui Keypad



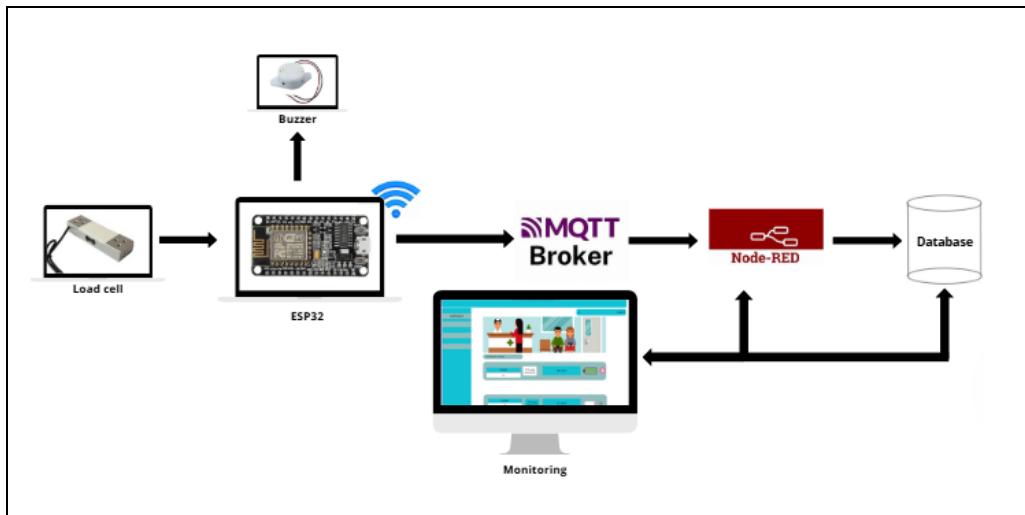
Gambar 3. 12 Skenario Pengujian Penginputan Data melalui Keypad

Skenario pengujian ini bertujuan untuk menguji kemampuan sistem dalam menerima input tetesan per menit dari keypad. Data input dari keypad diproses oleh ESP32, yang mengatur motor servo untuk menyesuaikan aliran tetesan infus sesuai dengan pengaturan tetesan per menit. Informasi ini kemudian dikirimkan ke broker MQTT melalui konektivitas Wi-Fi yang dimiliki ESP32. Data tersebut disimpan di database dan ditampilkan secara *real-time* pada antarmuka monitoring. Sistem ini memastikan bahwa input dari pengguna dapat diterima dengan baik, aliran infus dapat dikendalikan sesuai dengan tetesan per menit yang diinginkan, dan informasi monitoring tersedia secara akurat.

**Tabel 3. 5** Skenario Pengujian Penginputan Data

<b>Nama Butir Uji</b>	Pengujian Penginputan Data dari Keypad
<b>Tujuan</b>	Memastikan data dari keypad dapat diinputkan dan diteruskan ke sistem dengan benar.
<b>Deskripsi</b>	Pengujian sistem bertujuan untuk memverifikasi bahwa data yang diinputkan melalui keypad dapat diterima oleh ESP32, ditampilkan di LCD, dan diteruskan ke broker MQTT menggunakan koneksi Wi-Fi bawaan ESP32.
<b>Kondisi Awal</b>	Keypad terhubung dengan ESP32, LCD terpasang, dan sistem dalam keadaan aktif.
<b>Tanggal Pengujian</b>	-
<b>Penguji</b>	Mahasiswa
<b>Skenario Uji</b>	<ol style="list-style-type: none"> <li>1. Masukkan angka melalui keypad.</li> <li>2. Amati angka yang muncul pada LCD.</li> <li>3. Pastikan data yang dimasukkan melalui keypad diteruskan ke ESP32</li> <li>4. Periksa data yang diterima di MQTT Broker untuk memvalidasi bahwa input dikirimkan.</li> </ol>
<b>Kriteria Evaluasi Hasil</b>	<p>Data yang dimasukkan melalui keypad berhasil:</p> <ol style="list-style-type: none"> <li>1. Ditampilkan di LCD.</li> <li>2. Diteruskan ke ESP32 dan muncul di broker MQTT tanpa kesalahan.</li> </ol>

### 3.2.12 Skenario Pengujian Menghitung Volume Infus



Gambar 3. 13 Skenario Pengujian Menghitung Volume Infus

Skenario pengujian ini bertujuan untuk menguji kemampuan sistem dalam menghitung volume cairan infus yang telah digunakan, berdasarkan data yang diberikan oleh load cell. Setelah nilai volume infus diinput melalui sistem, load cell akan mendeteksi perubahan berat cairan infus yang telah mengalir.

Tabel 3. 6 Skenario Pengujian Menghitung Volume Infus

<b>Nama Butir Uji</b>	Pengujian Menghitung Volume Infus melalui Load Cell
<b>Tujuan</b>	Memastikan data berat dari Load Cell dapat digunakan untuk menghitung volume infus secara akurat.
<b>Deskripsi</b>	Sistem akan menghitung volume infus berdasarkan perubahan berat cairan yang terdeteksi oleh load cell.
<b>Kondisi Awal</b>	Sistem terhubung sepenuhnya, load cell terpasang dengan baik pada jalur infus, ESP32 dalam keadaan aktif.
<b>Tanggal Pengujian</b>	-
<b>Penguji</b>	Mahasiswa
<b>Skenario Uji</b>	
	1. Inputkan volume cairan infus awal ke dalam sistem.

<ol style="list-style-type: none"> <li>2. Nyalakan sistem dan pastikan load cell terpasang pada jalur infus.</li> <li>3. Sistem mulai mendeteksi perubahan berat cairan infus saat digunakan.</li> <li>4. Load cell mengirimkan data perubahan berat ke mikrokontroler (ESP32).</li> <li>5. ESP32 mengonversi data berat menjadi volume cairan infus yang telah digunakan.</li> <li>6. Data volume yang dihitung dikirim ke MQTT Broker dan disimpan di database.</li> <li>7. Volume cairan infus yang tersisa ditampilkan di antarmuka monitoring secara <i>real-time</i>.</li> <li>8. Jika volume cairan mencapai batas minimum, buzzer akan aktif sebagai peringatan.</li> </ol>	<p><b>Kriteria Evaluasi Hasil</b></p> <p>Volume cairan infus yang dihitung sesuai dengan perubahan berat yang terdeteksi oleh load cell, dan notifikasi buzzer berfungsi saat volume mencapai batas minimum.</p>
---	--

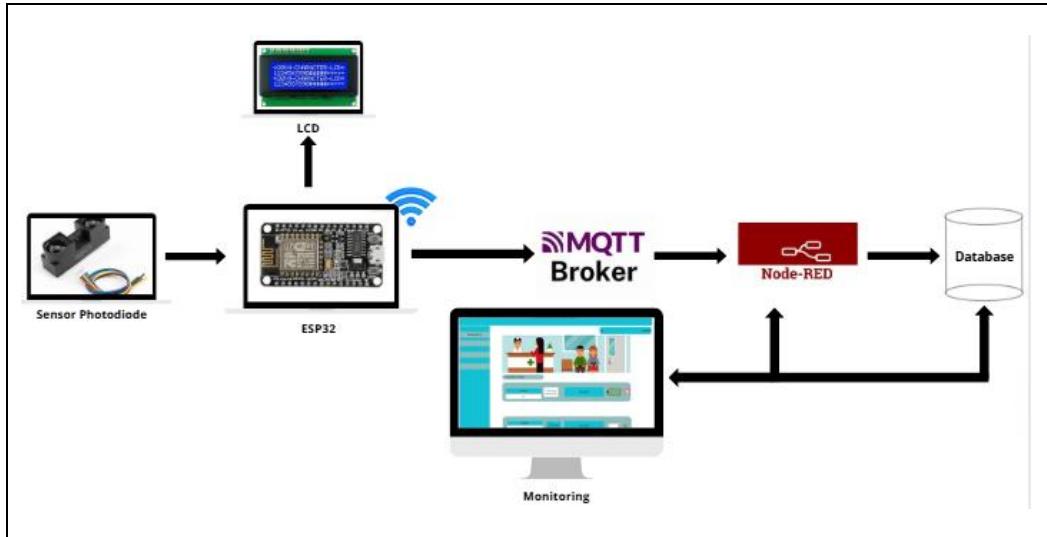
### 3.2.13 Skenario Pengujian Penginputan Data Pasien Melalui Website

Skenario pengujian ini dirancang untuk memastikan bahwa sistem mampu menerima, memproses, dan menyimpan data pasien yang diinput melalui *website*. Data yang diinput meliputi nama pasien, nomor bed, dan ruangan. Tujuan pengujian adalah untuk memastikan form input pada *website* bekerja dengan baik, data yang diinput tersimpan dengan benar di database, dan tidak terjadi kesalahan validasi. Pengujian dilakukan dengan memasukkan data pasien secara manual melalui form yang tersedia di *website*. Evaluasi hasil pengujian akan didasarkan pada keberhasilan penyimpanan data di database dan akurasi informasi yang ditampilkan kembali.

**Tabel 3. 7** Skenario Pengujian Penginputan Data Pasien melalui Website

<b>Nama Butir Uji</b>	Pengujian Penginputan Data Pasien Melalui <i>Website</i>
<b>Tujuan</b>	Memastikan data pasien dapat diinput dan disimpan di database dengan benar melalui <i>website</i> .
<b>Deskripsi</b>	Pengujian form input pada <i>website</i> untuk memasukkan data pasien berupa nama, nomor bed, dan ruangan, serta memastikan data tersimpan dan tampil sesuai.
<b>Kondisi Awal</b>	<i>Website</i> aktif dan terhubung ke <i>database</i> , form input pasien tersedia, dan pengguna telah login ke sistem.
<b>Tanggal Pengujian</b>	-
<b>Penguji</b>	Mahasiswa
<b>Skenario Uji</b>	<ol style="list-style-type: none"> <li>1. Buka halaman form input pasien pada <i>website</i>.</li> <li>2. Masukkan data pasien (contoh: Nama: Paian, Nomor Bed: 1, Ruangan: A).</li> <li>3. Klik tombol Tambah Pasien.</li> <li>4. Periksa apakah data pasien berhasil tersimpan di <i>database</i>.</li> <li>5. Verifikasi data dengan menampilkan kembali data yang telah diinput di Data Pasien</li> </ol>
<b>Kriteria Evaluasi Hasil</b>	<ol style="list-style-type: none"> <li>1. Data pasien tersimpan di <i>database</i> tanpa <i>error</i>.</li> <li>2. Informasi yang ditampilkan sesuai dengan data yang diinput.</li> </ol>

### 3.2.14 Skenario Pengujian Menghitung Tetesan Infus



**Gambar 3. 14** Skenario Pengujian Menghitung Tetesan Infus

Skenario pengujian ini bertujuan untuk menguji kemampuan sistem dalam menghitung tetesan infus menggunakan sensor photodiode. Sensor photodiode mendeteksi kecepatan tetesan, data diproses oleh ESP32, ditampilkan pada LCD, dan dikirimkan ke ESP32 untuk diteruskan ke MQTT Broker. Data ini disimpan di database dan ditampilkan secara *real-time* pada antarmuka monitoring. Pengguna dapat mengatur kecepatan tetesan melalui antarmuka, dan sistem akan menyesuaikan serta memantau keakuratan kecepatan infus sesuai pengaturan tersebut.

**Tabel 3. 8** Skenario Pengujian Menghitung Tetesan Infus

<b>Nama Butir Uji</b>	Pengujian Menghitung Tetesan Infus Menggunakan Sensor Photodiode
<b>Tujuan</b>	Memastikan penghitungan tetesan infus berjalan akurat menggunakan sensor photodiode.
<b>Deskripsi</b>	Penggunaan sensor photodiode untuk menghitung kecepatan tetesan infus sesuai nilai yang diinginkan.

<b>Kondisi Awal</b>	Sistem terhubung sepenuhnya, sensor photodiode telah terpasang, ESP32 dalam keadaan aktif, serta jalur infus terpasang.
<b>Tanggal Pengujian</b>	-
<b>Penguji</b>	Mahasiswa
<b>Skenario Uji</b>	<ol style="list-style-type: none"> <li>1. Pasang infus pada alat dan nyalakan sistem.</li> <li>2. Atur tetesan sesuai nilai hasil TPM</li> <li>3. Sensor Photodiode menghitung jumlah tetesan infus setiap menit</li> <li>4. Hasil penghitungan sensor photodiode ditampilkan ke LCD secara <i>real time</i></li> </ol>
<b>Kriteria Evaluasi Hasil</b>	Jumlah tetesan pada photodiode sesuai dengan TPM yang diinput

### 3.2.15 Skenario Pengujian Notifikasi melalui Buzzer

Dalam skenario ini, fokus pengujian adalah fitur notifikasi melalui buzzer, yang diaktifkan saat volume cairan infus tersisa 100ml. Tujuan pengujian adalah untuk memastikan bahwa sistem dapat mendeteksi kondisi kritis dan memberikan peringatan secara *real-time* kepada pengguna. Pengujian dilakukan dengan simulasi pengurangan cairan infus secara bertahap hingga mencapai batas volume 100ml. Hasil pengujian akan dievaluasi berdasarkan respons sistem, yaitu apakah buzzer menyala pada saat yang tepat.

**Tabel 3. 9** Skenario Pengujian Notifikasi melalui Buzzer

<b>Nama Butir Uji</b>	Pengujian Notifikasi Melalui Buzzer
<b>Tujuan</b>	Memastikan buzzer memberikan notifikasi saat volume cairan infus tersisa 100ml.
<b>Deskripsi</b>	Pengujian sistem yang memantau volume cairan infus dan mengaktifkan buzzer sebagai notifikasi jika volume cairan mencapai ambang batas 100ml.

<b>Kondisi Awal</b>	Sistem terhubung sepenuhnya, sensor photodiode telah terpasang, ESP32 dalam keadaan aktif, serta jalur infus terpasang.
<b>Tanggal Pengujian</b>	-
<b>Penguji</b>	Mahasiswa
<b>Skenario Uji</b>	<ol style="list-style-type: none"> <li>1. Pasang kantong infus pada alat.</li> <li>2. Nyalakan sistem dan pastikan load cell aktif.</li> <li>3. Biarkan sistem membaca berat awal cairan infus.</li> <li>4. Kurangi berat cairan secara bertahap hingga sistem mendeteksi volume cairan tersisa 100ml.</li> <li>5. Perhatikan apakah buzzer aktif ketika volume cairan mencapai 100ml.</li> </ol>
<b>Kriteria Evaluasi Hasil</b>	Buzzer menyala dengan jelas ketika volume cairan infus tersisa 100ml dan berhenti menyala saat volume cairan di atas ambang batas.

## **BAB IV**

### **IMPLEMENTASI DAN HASIL PENGUJIAN**

Pada bab ini menjelaskan tahap implementasi dan pengujian dari Node-RED. Proses implementasi dilakukan dengan berbagai aktivitas instalasi *software* yang penting dalam pembuatan proyek ini. Pada bab ini juga dijelaskan tahap pengujian yang disesuaikan dengan skenario pengujian yang telah dibuat sebelumnya.

#### **4.1 Instalasi**

Pada bagian ini, akan dijelaskan tentang langkah-langkah yang dilakukan untuk menginstal berbagai aplikasi yang diperlukan dalam proses implementasi yang akan dilakukan. Berikut akan dijelaskan secara detail tentang proses instalasi yang akan dilakukan.

##### **4.1.1 Instalasi MQTT**

MQTT (*Message Queuing Telemetry Transport*) merupakan protokol komunikasi yang ringan dan efisien, sering digunakan dalam sistem IoT. Proses instalasinya dimulai dengan memilih broker MQTT, seperti Mosquitto, yang dapat diunduh melalui situs resmi atau manajer paket.

###### **1. Langkah 1 : Instalasi Mosquitto**

Berikut adalah perintah untuk menginstal aplikasi mosquitto pada raspberry :

```
#apt-get update  
  
#apt-get upgrade  
  
#apt-get dist-upgrade  
  
#apt-get install mosquitto mosquitto-clients python-mosquitto
```

###### **2. Langkah 2: Konfigurasi Mosquito**

```
#nano /etc/mosquitto/conf.d/mosquitto.conf
```

Kemudian isi file dengan konfigurasi dibawah ini

```

user mosquitto
max_queued_messages 200
message_size_limit 0
allow_zero_length_clientid true
allow_duplicate_messages false

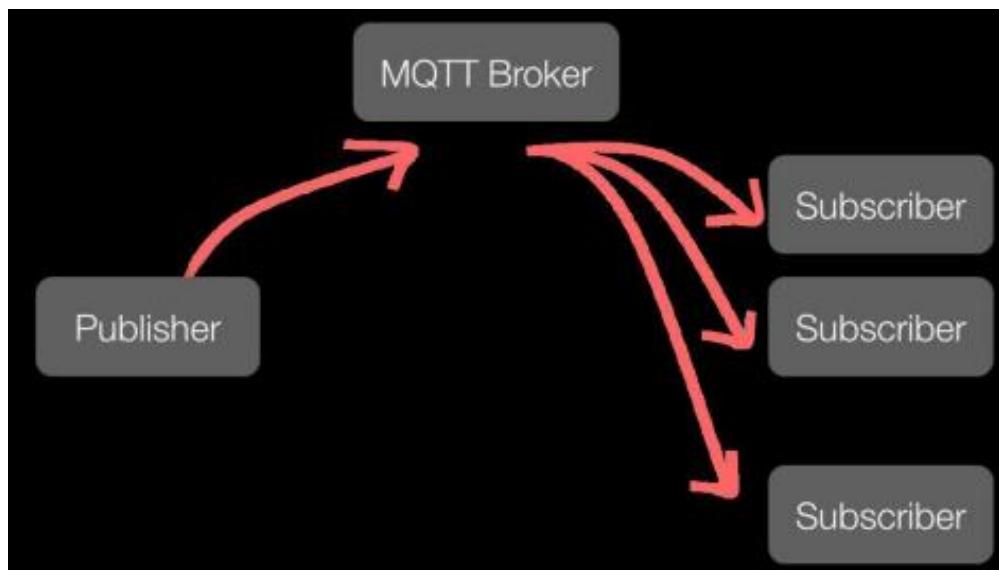
listener 1883
autosave_interval 900
autosave_on_changes false
persistence true
persistence_file mosquitto.db
allow_anonymous true
password_file /etc/mosquitto/passwd

```

### 3. Langkah 3 : Restart Service Mosquito

```
#systemctl restart mosquito
```

Ada komponen penting pada MQTT yaitu *publisher* dan *subscriber*. *Publisher* berperan untuk mengirimkan pesan kepada semua *subscriber* sehingga *subscriber* akan menerima pesan dari *subscriber* sesuai dengan channel/topik yang terbuka pada *subscriber* tersebut.



**Gambar 4. 1** Arsitektur Dasar Protokol MQTT

#### 4.1.2 Instalasi Node-RED

Node-RED adalah sebuah *tool* berbasis *browser* untuk membuat aplikasi *Internet of Things* (IoT) yang mana lingkungan pemrograman visualnya mempermudah penggunanya untuk membuat aplikasi sebagai “*flow*”. Flow ini terbentuk dari node-node yang saling berhubungan di mana tiap node melakukan tugas tertentu. Instalasi Node-RED memerlukan Node.js sebagai prasyarat. Langkah pertama adalah mengunduh dan memasang Node.js dari situs resmi <https://nodejs.org>, disarankan menggunakan versi LTS (*Long Term Support*) untuk stabilitas.

Gunakan perintah berikut untuk memastikan **Node.js** sudah terinstal:

```
node -version
npm -version
```

Berikut adalah langkah-langkah instalasi Node-RED :

1. Jalankan perintah di terminal :

```
sudo npm install -g --unsafe-perm node-red
```

2. Setelah instalasi selesai, verifikasi dengan :

```
node-red -v
```

3. Jalankan Node-RED dengan perintah :

```
node-red
```

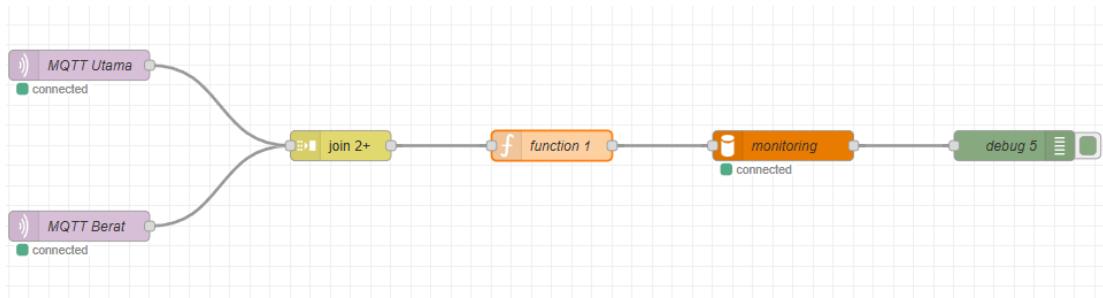
#### 4.2 Implementasi

Pada sub bagian ini menjelaskan proses implementasi dari setiap perangkat lunak dan perangkat keras yang membangun sistem otomatis dan monitoring infus berbasis IoT.

##### 4.2.1 Implementasi Node-RED

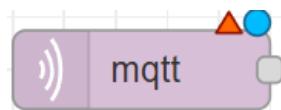
Bagian ini menjelaskan proses implementasi Node-RED dalam pengembangan proyek Tugas Akhir. Implementasi ini melibatkan pembuatan sistem IoT yang menjadi perangkat utama sebagai platform untuk mengaplikasikan Node-RED. Proyek ini disusun menggunakan beberapa node yang saling terhubung, membentuk alur kerja yang terintegrasi.

Berikut adalah gambaran pengiriman data pada Node-RED.



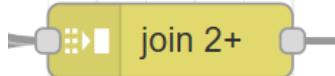
Node-node yang Menyusun projek ini yaitu :

### 1. MQTT



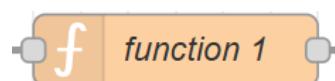
MQTT berfungsi sebagai jalur utama untuk menerima data dari perangkat IoT melalui protokol MQTT. Node ini bertindak sebagai penghubung antara broker MQTT dan alur sistem pada Node-RED. Dalam proyek ini, MQTT digunakan untuk meneruskan data dari alat ke sistem secara *real-time* agar dapat diproses lebih lanjut.

### 2. Join 2+



Node "Join" digunakan untuk menggabungkan beberapa pesan menjadi satu payload yang dapat diproses lebih lanjut. Dalam konfigurasi "Join 2+", node ini dirancang untuk mengumpulkan setidaknya dua pesan atau lebih sebelum menghasilkan keluaran. Node ini berguna ketika data dari berbagai sumber perlu digabungkan menjadi satu entitas, seperti array atau objek JSON, untuk analisis atau pengolahan lebih lanjut. Penggabungan pesan ini dapat diatur berdasarkan jumlah pesan tertentu, waktu tunggu, atau kriteria lainnya sesuai kebutuhan. Node "Join" sering digunakan dalam alur yang memerlukan sinkronisasi data dari beberapa sumber atau node input.

### 3. Function 1



Function berfungsi untuk melakukan pengolahan data yang diterima. Operasi yang dapat dilakukan meliputi mengubah format data, melakukan perhitungan matematika, menyaring data berdasarkan kondisi tertentu, menambahkan atau memodifikasi properti pesan, dan menggabungkan data menggunakan konteks. Dalam proyek ini, Function digunakan sebagai penyaring data untuk memastikan hanya data yang sesuai dengan kriteria tertentu yang diteruskan ke proses berikutnya, sehingga alur data menjadi lebih terstruktur dan efisien.

#### 4. MySQL



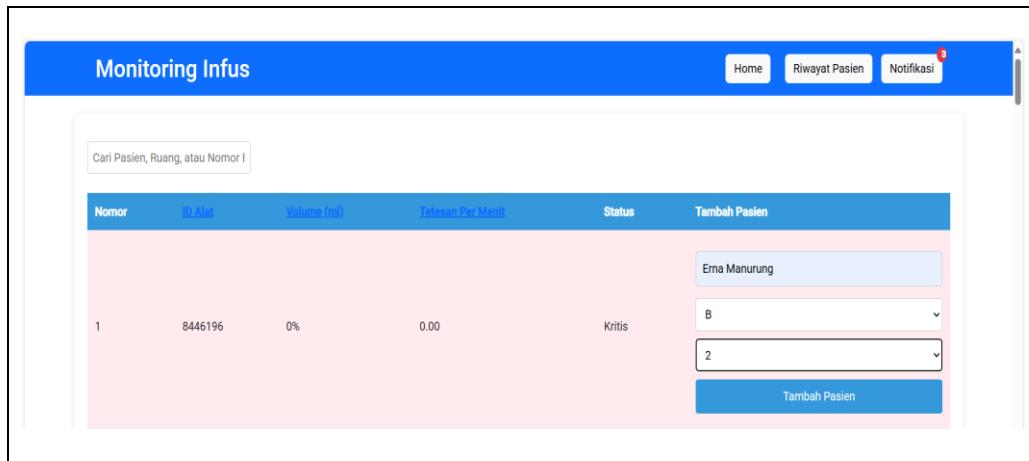
MySQL berfungsi sebagai penghubung antara sistem dan database. Node ini digunakan untuk mengirimkan data yang telah diolah oleh *Function* ke dalam database MySQL. Dalam proyek ini, node MySQL memastikan data yang relevan tersimpan dengan baik di database sehingga dapat diakses atau digunakan untuk kebutuhan monitoring dan analisis lebih lanjut.

#### 5. Debug



Debug dalam Node-RED berfungsi untuk menampilkan data pada tab debug di editor Node-RED. Node ini digunakan untuk memantau apakah data berhasil terkirim, diolah, atau diterima pada setiap tahap alur sistem. Dalam proyek kami, Debug membantu memastikan bahwa aliran data berjalan sesuai dengan yang diharapkan dan memudahkan proses troubleshooting jika terjadi kesalahan.

#### 4.2.2 Implementasi Inputan Data Pasien melalui Website



The screenshot shows a web application titled "Monitoring Infus". At the top, there is a blue header bar with the title and three buttons: "Home", "Riwayat Pasien", and "Notifikasi". Below the header is a search bar labeled "Cari Pasien, Ruang, atau Nomor I". The main area contains a table with columns: "Nomor", "ID Alert", "Volume (ml)", "Tahanan Per Menit", "Status", and "Tambah Pasien". A single row is visible, showing data for patient number 1, ID alert 8446196, 0% volume, 0.00 tahanan per menit, and status Kritis. To the right of this table is a form for adding a new patient, with fields for "Nama Pasien" (Erna Manurung), "Status" (dropdown menu showing 'B'), and "Ruangan" (dropdown menu showing '2'). A blue "Tambah Pasien" button is at the bottom of this form.

**Gambar 4. 2** Implementasi Inputan Data Pasien melalui Website

Sistem yang dibangun menyediakan fitur untuk input data pasien melalui *website*, yang bertujuan untuk mempermudah tenaga medis dalam mencatat informasi pasien dengan lebih efisien. Data yang diinput meliputi nama pasien, nomor bed, dan ruangan. Proses input data dilakukan melalui formulir yang terdapat pada halaman *website*. Formulir ini dilengkapi dengan kolom isian untuk nama pasien, nomor bed, dan ruangan, yang masing masing dilengkapi validasi sederhana untuk memastikan format data yang dimasukkan sesuai dengan ketentuan. Setelah data diisi, pengguna dapat mengklik tombol Tambah pasien untuk mengirimkan data ke server, yang kemudian akan diproses oleh sistem.

Data yang berhasil diproses selanjutnya disimpan di dalam database menggunakan integrasi Node-RED dengan node MySQL. Dengan demikian, data pasien yang telah disimpan dapat diakses secara *real-time* oleh sistem lain yang memerlukan informasi tersebut.

#### 4.2.3 Implementasi Inputan TPM melalui Keypad



**Gambar 4. 3** Inputan TPM melalui Keypad

Pada sistem ini, inputan Tetesan Per Menit (TPM) dilakukan melalui sebuah keypad yang terhubung langsung dengan mikrokontroler ESP32. Keypad ini memungkinkan tenaga medis untuk mengatur jumlah tetesan infus sesuai dengan kebutuhan pasien, berdasarkan perhitungan yang sudah ditentukan sebelumnya. Dengan menggunakan keypad, pengguna dapat memasukkan nilai TPM yang diinginkan dengan menekan tombol-tombol yang ada pada keypad.

Setelah data TPM diinput melalui keypad, nilai yang dimasukkan akan langsung diproses oleh mikrokontroler dan ditampilkan pada layar LCD sebagai bentuk verifikasi bagi tenaga medis. Hal ini memastikan bahwa input TPM yang dimasukkan telah berhasil diterima dan diproses dengan benar. Data yang ditampilkan pada LCD akan menunjukkan nilai TPM yang baru, yang selanjutnya digunakan untuk mengatur aliran infus sesuai dengan pengaturan yang diinginkan.

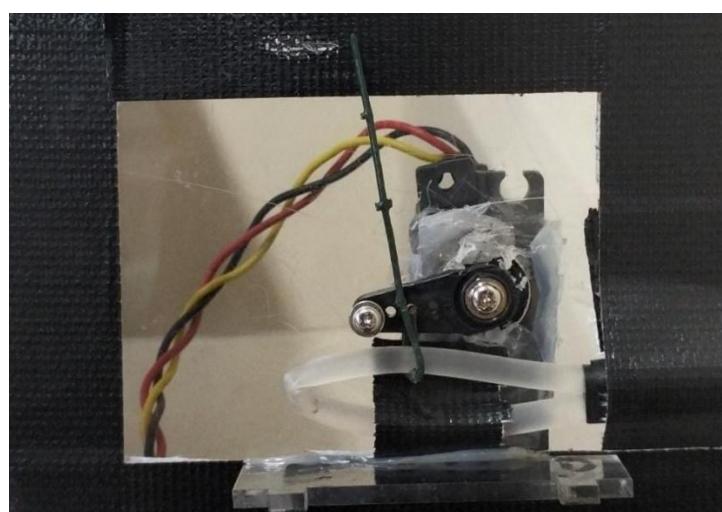
#### 4.2.4 Implementasi Penghitungan Volume Infus melalui Loadcell

Pada sistem ini, kantong infus digantungkan pada load cell yang berfungsi untuk mengukur berat cairan infus dalam kantong. Setiap kali cairan infus digunakan, berat kantong infus akan berkurang, dan load cell akan mendeteksi perubahan berat

tersebut. Perubahan berat yang terdeteksi langsung berhubungan dengan volume cairan infus yang telah keluar, dengan satuan pengukuran dalam gram.

Data yang dihasilkan oleh load cell ini kemudian diteruskan ke mikrokontroler ESP32, yang memproses nilai yang diterima dan mengirimkan informasi volume infus yang tersisa ke *website* untuk pemantauan secara *real-time*. Dengan sistem ini, tenaga medis dapat memonitor sisa cairan infus dengan lebih akurat, sehingga dapat mencegah terjadinya kehabisan cairan infus tanpa terdeteksi.

#### 4.2.5 Implementasi Pengaturan Tetesan melalui Servo Motor



**Gambar 4. 4** Implementasi Pengaturan Tetesan melalui Servo Motor

Pada sistem ini, pengaturan tetesan dilakukan dengan menggunakan servo motor yang berfungsi untuk mengatur aliran cairan infus melalui jalur infus. Servo motor dikendalikan berdasarkan hasil perhitungan Tetesan per Menit (TPM) yang dihitung sebelumnya. TPM ini mengatur jumlah tetesan yang harus dikeluarkan per menit sesuai dengan kebutuhan pasien. Servo motor akan bergerak untuk membuka atau menutup jalur infus dengan tepat sesuai dengan nilai TPM yang telah diatur.

Proses ini dimulai dengan penginputan nilai TPM melalui keypad, yang kemudian diproses oleh mikrokontroler ESP32. Berdasarkan nilai TPM yang diinput, mikrokontroler mengirimkan sinyal kontrol untuk menggerakkan servo motor. Servo motor akan membuka katup infus dengan kecepatan tertentu sehingga cairan

infus dapat mengalir sesuai dengan jumlah tetesan yang diinginkan. Dengan cara ini, sistem dapat memastikan bahwa jumlah cairan yang diberikan kepada pasien tetap sesuai dengan yang telah dihitung dan diatur.

#### 4.2.6 Implementasi Penghitungan Tetesan melalui Sensor Photodiode



**Gambar 4. 5** Implementasi Perhitungan Tetesan melalui Sensor Photodiode

Pada sistem ini, sensor photodiode digunakan untuk menghitung jumlah tetesan cairan infus yang keluar dari kantong infus. Sensor photodiode bekerja dengan mendeteksi perubahan cahaya yang dihasilkan oleh tetesan infus yang jatuh. Ketika tetesan cairan infus melewati area sensor, cahaya yang dipantulkan oleh tetesan akan terdeteksi oleh photodiode, menghasilkan sinyal listrik yang digunakan untuk menghitung jumlah tetesan.

Setiap tetesan yang melewati sensor photodiode akan menghasilkan pulsa sinyal yang terhitung oleh mikrokontroler ESP32. Mikrokontroler kemudian memproses sinyal ini untuk menghitung jumlah tetesan dalam satuan waktu, biasanya per menit. Hasil penghitungan ini sangat penting untuk memastikan bahwa aliran cairan infus sesuai dengan nilai yang diinginkan, yang sebelumnya telah dihitung berdasarkan TPM. Data yang dihitung oleh sensor photodiode ini akan dikirim ke sistem untuk pemantauan lebih lanjut, dan juga dapat ditampilkan pada *website*

untuk memudahkan tenaga medis dalam memantau pengaliran infus secara *real-time*.

### 4.3 Hasil Pengujian

Dalam sub bab ini, dijelaskan mengenai pengujian implementasi yang bertujuan untuk memastikan bahwa sistem yang dibangun berfungsi dengan baik. Tahap ini dilakukan untuk mengevaluasi apakah alat tersebut sudah sesuai dengan kebutuhan yang dirancang pada sub bab sebelumnya.

#### 4.3.1 Tes Script Butir-Uji-1

**Tabel 4. 1** Pengujian Penginputan Data melalui Keypad

<b>No. Fungsi</b>	TS-01
<b>Nama Butir Uji</b>	Pengujian Penginputan Data melalui Keypad
<b>Tujuan Pengujian</b>	Menguji keakuratan input data Kebutuhan Cairan (V), Faktor Tetes (FT), dan Waktu (T) melalui keypad, serta menghitung dan menampilkan hasil TPM pada LCD.
<b>Hasil yang diharapkan</b>	Nilai Kebutuhan Cairan (V), Faktor Tetes (FT), dan Waktu (T) dapat dimasukkan dengan benar melalui keypad dan ditampilkan pada LCD. Hasil perhitungan TPM juga ditampilkan dengan benar setelah input lengkap.
<b>Tanggal Pengujian</b>	06 Januari 2025
<b>Penguji</b>	Paian Manalu, Erna Manurung, Batara Nadapdap
<b>Skenario Uji</b>	
<ol style="list-style-type: none"> <li>1. Pengguna memasukkan nilai Kebutuhan Cairan (V) menggunakan keypad.</li> <li>2. Pengguna memasukkan nilai Faktor Tetes (FT).</li> <li>3. Pengguna memasukkan nilai Waktu (T).</li> </ol>	

<p>4. Setelah semua input diterima, tombol "#" ditekan untuk menghitung dan menampilkan TPM pada LCD.</p>			
<b>Kriteria Evaluasi Hasil</b>			
<ul style="list-style-type: none"> <li>- Input data Kebutuhan Cairan (V), Faktor Tetes (FT), dan Waktu (T) dapat diterima dengan benar.</li> <li>- Tombol "*" menghapus input yang salah.</li> <li>- Tombol "#" mengonfirmasi input dan menampilkan TPM pada LCD sesuai dengan perhitungan.</li> <li>- Sistem tidak mengalami kesalahan atau <i>crash</i> selama pengujian.</li> </ul>			
<b>Kasus dan Hasil Uji</b>			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kebutuhan Cairan (V) = 500 ml, Faktor Tetes (FT) = 20, Waktu (T) = 10 jam	LCD menampilkan Nilai TPM sesuai dengan tetesan yang diinput melalui keypad Hasil perhitungan: $500 \times 20 / (10 \times 60)$ $= 16,67(17 \text{ Tetesan})$	Nilai Kebutuhan Cairan, Faktor Tetes, dan Waktu dimasukkan dengan benar, tombol "#" menghitung TPM dan menampilkannya di LCD.	Sistem bekerja sesuai harapan, hasil TPM yang dihitung ditampilkan dengan tepat.
<b>Hasil</b>			



**Gambar 4. 6 Penginputan Nilai TPM melalui Keypad**



**Gambar 4. 7 Hasil Inputan Keypad**

Pada gambar diatas memperlihatkan tahapan proses penginputan data serta hasil perhitungan Tetesan Per Menit (TPM) yang dirancang menggunakan keypad dan LCD. Pada Gambar 4.6, ditampilkan proses pemasukan data oleh pengguna yang terdiri dari volume cairan infus (dalam mililiter), faktor tetes, dan waktu pemberian infus (dalam jam) yang diinput secara bertahap melalui keypad numerik. Setelah seluruh data dimasukkan, tombol "#" ditekan untuk mengonfirmasi dan memulai proses perhitungan. Gambar 4.7 menunjukkan tampilan hasil yang muncul pada

LCD, berupa informasi lengkap mengenai volume, faktor tetes, waktu infus, serta nilai TPM yang dihitung secara otomatis oleh sistem. Berdasarkan tampilan tersebut, dapat disimpulkan bahwa alat berfungsi dengan baik dalam menerima masukan, melakukan perhitungan, dan menampilkan hasil sesuai dengan parameter yang diberikan.

#### 4.3.2 Tes Script Butir-Uji-2

**Tabel 4. 2 Pengujian Menghitung Volume Infus**

<b>No. Fungsi</b>	TS-02
<b>Nama Butir Uji</b>	Pengujian Menghitung Volume Infus
<b>Tujuan Pengujian</b>	Menguji keakuratan perhitungan volume infus berdasarkan input Kebutuhan Cairan (V) dan perubahan volume yang dihitung oleh load cell. Pastikan sistem dapat menghitung perubahan volume infus dengan akurat.
<b>Hasil yang diharapkan</b>	Volume infus dihitung dengan akurat oleh load cell, dan perubahan volume infus terdeteksi dengan benar.
<b>Tanggal Pengujian</b>	06 Januari 2025
<b>Penguji</b>	Paian Manalu, Erna Manurung, Batara Nadapdap
<b>Skenario Uji</b>	
<ol style="list-style-type: none"> <li>1. Pengguna memasukkan nilai Kebutuhan Cairan (V) menggunakan keypad.</li> <li>2. Sistem mengukur volume infus menggunakan load cell.</li> <li>3. Sistem menghitung perubahan volume infus berdasarkan data yang diterima.</li> </ol>	
<b>Kriteria Evaluasi Hasil</b>	
<ul style="list-style-type: none"> <li>- Input data Kebutuhan Cairan (V) diterima dengan benar melalui keypad.</li> <li>- Sistem berhasil menghitung volume infus dengan akurat menggunakan load cell.</li> </ul>	

<ul style="list-style-type: none"> <li>- Perubahan volume infus terdeteksi dan dihitung dengan tepat oleh load cell.</li> </ul>			
<b>Kasus dan Hasil Uji</b>			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Kebutuhan Cairan (V) = 500 ml	Volume infus dihitung secara akurat oleh load cell dan perubahan volume tercatat dengan tepat.	Volume infus dihitung dengan benar oleh load cell, dan perubahan volume terdeteksi dengan akurat.	Sistem berfungsi dengan baik dalam menghitung volume infus menggunakan load cell dengan akurat.
<b>Hasil</b>			



**Gambar 4. 8 Pengujian Menghitung Volume Infus menggunakan Loadcell**

```
Data berhasil terkirim!
ID Alat: 1624116, Berat: 434 gram, Buzzer: Mati
Data berhasil terkirim!
ID Alat: 1624116, Berat: 434 gram, Buzzer: Mati
Data berhasil terkirim!
ID Alat: 1624116, Berat: 434 gram, Buzzer: Mati
Data berhasil terkirim!
ID Alat: 1624116, Berat: 434 gram, Buzzer: Mati
Data berhasil terkirim!
ID Alat: 1624116, Berat: 434 gram, Buzzer: Mati
Data berhasil terkirim!
```

**Gambar 4. 9 Data Volume Infus dengan Volume Infus 434 gram**

Pengelolaan Pasien								
Cari Pasien, Ruang, atau Nomor Bed								
Tambah Pasien								
Nama Pasien	Ruang	Nomor Bed	Volume (%)	Tetesan per Menit	Tetesan	Faktor	Status	Aksi
Barata Nadapdap	F	2	66%	0.00	0	Not Specified	<span style="background-color: green; color: white; padding: 2px;">Aman</span>	<span style="color: green;">Hapus / Edit</span>

**Gambar 4. 10 Tampilan Website dengan Volume Infus 66%**

Pengujian ini dilakukan untuk mengukur volume infus menggunakan sensor loadcell. Sensor mendeteksi berat total kantong infus dan mengubahnya menjadi sinyal digital. Berdasarkan gambar 38, berat terukur adalah 434 gram. Data dikirim secara berkala dengan status "Data berhasil terkirim!". Berat tersebut mencerminkan volume cairan infus yang tersisa. Karena buzzer dalam kondisi "Mati", artinya infus belum mencapai batas minimum. Hasil pengujian menunjukkan sistem berfungsi dengan baik dan data berhasil dikirim tanpa gangguan.

#### 4.3.3 Tes Script Butir-Uji-3

**Tabel 4. 3 Pengujian Penginputan Data Pasien Melalui Website**

<b>No. Fungsi</b>	TS-03
<b>Nama Butir Uji</b>	Pengujian Penginputan Data Pasien Melalui <i>Website</i>
<b>Tujuan Pengujian</b>	Menguji penginputan data pasien melalui <i>website</i> dengan memastikan nama pasien, ruangan , dan nomor bed dapat dimasukkan dengan benar dan data tersimpan dengan baik di database.
<b>Hasil yang diharapkan</b>	Nama pasien, nomor bed, dan ruangan dapat diinput dengan benar melalui <i>website</i> dan disimpan dengan tepat di database.
<b>Tanggal Pengujian</b>	06 Januari 2025
<b>Penguji</b>	Paian Manalu, Erna Manurung, Batara Nadapdap
<b>Skenario Uji</b>	
<ol style="list-style-type: none"> <li>1. <i>User</i> mengakses <i>website</i> dan memasukkan nama pasien, nomor bed, dan ruangan.</li> <li>2. <i>User</i> mengklik tombol "Tambah Pasien" untuk mengirimkan data.</li> </ol>	

<p>3. Data yang dimasukkan (nama pasien, ruangan dan ruangan) disimpan dalam database.</p>			
<b>Kriteria Evaluasi Hasil</b>			
<ul style="list-style-type: none"> <li>- Input data pasien (nama, nomor bed, ruangan) diterima dengan benar.</li> <li>- Data berhasil disimpan di database tanpa error.</li> </ul>			
<b>Kasus dan Hasil Uji</b>			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama Pasien = "Paian Manalu", Ruangan = "B" Nomor Bed = "1",	Data nama pasien, ruangan, dan nomor bed berhasil disimpan dengan benar di database.	Data berhasil diinput dan disimpan tanpa <i>error</i> .	Penginputan data pasien melalui <i>website</i> berjalan sesuai harapan dan berhasil menyimpan data ke <i>database</i> .
<b>Hasil</b>			

Paian manalu

B

1

Tambah Pasien

**Gambar 4. 11** Input Data Pasien

Monitoring Infus

Data Riwayat Pasien

No	Nama Pasien	Ruang	Nomor Bed	Tanggal Hapus
1	Paian manalu	B	1	2025-01-07 10:43:45

**Gambar 4. 12** Tampilan Data Pasien yang Sudah di Input

Server: 127.0.0.1 » Database: monitor » Table: delete\_pasien

	Browse	Structure	SQL	Search	Insert	Export	Import	Privileges	Operations	
<input type="checkbox"/> Show all	Number of rows: 25	Filter rows: Search this table	Sort by key: None							
Extra options										
		id_pasien	id_alat	nama_pasien	ruang	nomor_bed	tanggal			
<input type="checkbox"/>		Edit	Copy	Delete	11	INF100022	Paian manalu	B	1	2025-01-07 10:43:45

**Gambar 4. 13** Tampilan Database Pasien yang di Input

Pengujian penginputan data pasien melalui website menunjukkan bahwa sistem berhasil menerima dan menyimpan informasi dengan baik. Pada pengujian ini, data yang dimasukkan meliputi nama pasien "Paian Manalu", ruangan "B", dan nomor bed "1". Hasil yang diharapkan adalah seluruh data tersebut tersimpan dengan benar di dalam database tanpa terjadi kesalahan. Berdasarkan pengamatan, proses input

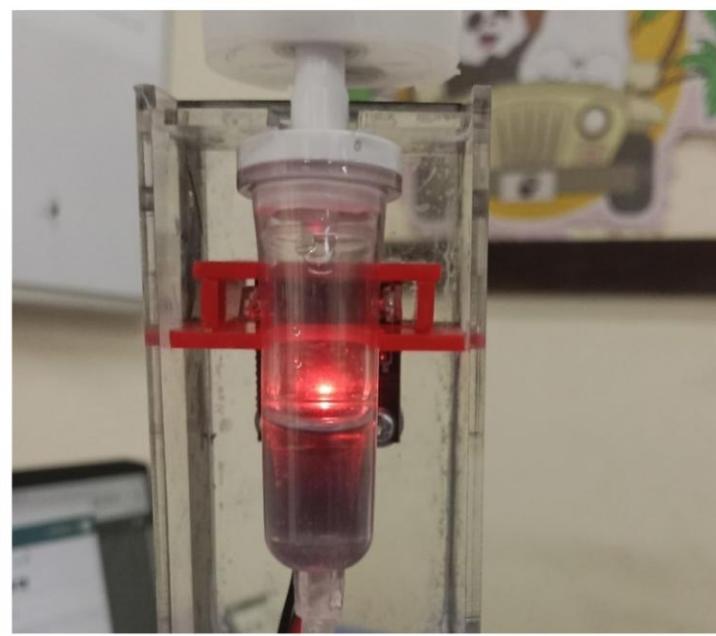
berjalan lancar dan data berhasil disimpan tanpa munculnya error. Hal ini menunjukkan bahwa fitur penginputan data pasien pada sistem telah berfungsi sesuai dengan yang dirancang. Proses penyimpanan data juga telah terverifikasi melalui tampilan data yang sesuai di antarmuka pengguna. Dengan demikian, dapat disimpulkan bahwa penginputan data pasien melalui website berjalan dengan baik dan sesuai harapan.

#### 4.3.4 Tes Script Butir-Uji-4

**Tabel 4. 4 Pengujian Menghitung Tetesan Infus**

<b>No. Fungsi</b>	TS-04
<b>Nama Butir Uji</b>	Pengujian Menghitung Tetesan Infus
<b>Tujuan Pengujian</b>	Menguji keakuratan penghitungan tetesan infus menggunakan sensor photodiode, memastikan bahwa sensor dapat mendeteksi tetesan infus dengan tepat dan hasil penghitungan tetesan sesuai dengan nilai TPM yang telah diinput.
<b>Hasil yang diharapkan</b>	Sensor photodiode dapat menghitung tetesan infus dengan akurat dan jumlah tetesan yang terdeteksi sesuai dengan hasil perhitungan TPM yang telah diinput sebelumnya.
<b>Tanggal Pengujian</b>	06 Januari 2025
<b>Penguji</b>	Paian Manalu, Erna Manurung, Batara Nadapdap
<b>Skenario Uji</b>	<ol style="list-style-type: none"> <li>1. Pengguna menyiapkan sistem dengan sensor photodiode terpasang pada aliran infus.</li> <li>2. Pengguna memulai aliran infus dan sensor photodiode mulai mendeteksi tetesan.</li> <li>3. Sensor mengirim data tetesan yang terdeteksi ke sistem.</li> <li>4. Sistem menghitung jumlah tetesan yang terdeteksi oleh sensor dan mencocokkannya dengan nilai TPM yang telah diinput.</li> </ol>

<b>Kriteria Evaluasi Hasil</b>			
<b>Kasus dan Hasil Uji</b>			
<b>Data Masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Aliran infus dimulai dengan kecepatan standar dan TPM yang telah dihitung sebelumnya.	Sensor photodiode dapat mendeteksi jumlah tetesan sesuai dengan TPM yang telah diinput.	Sensor berhasil mendeteksi tetesan dan jumlahnya sesuai dengan hasil perhitungan TPM.	Sensor photodiode berfungsi dengan baik dalam menghitung tetesan infus yang sesuai dengan nilai TPM.
<b>Hasil</b>			



**Gambar 4. 14** Penghitungan Tetesan Infus Menggunakan Photodiode

Vol: 500.00 mL  
Factor: 20.00  
Time: 1.00 hr  
Tetesan per menit: 20

A digital display screen showing the following information:

- Vol: 500.00 mL
- Factor: 20.00
- Time: 1.00 hr
- Tetesan per menit: 20

The text is displayed in white against a dark blue background.

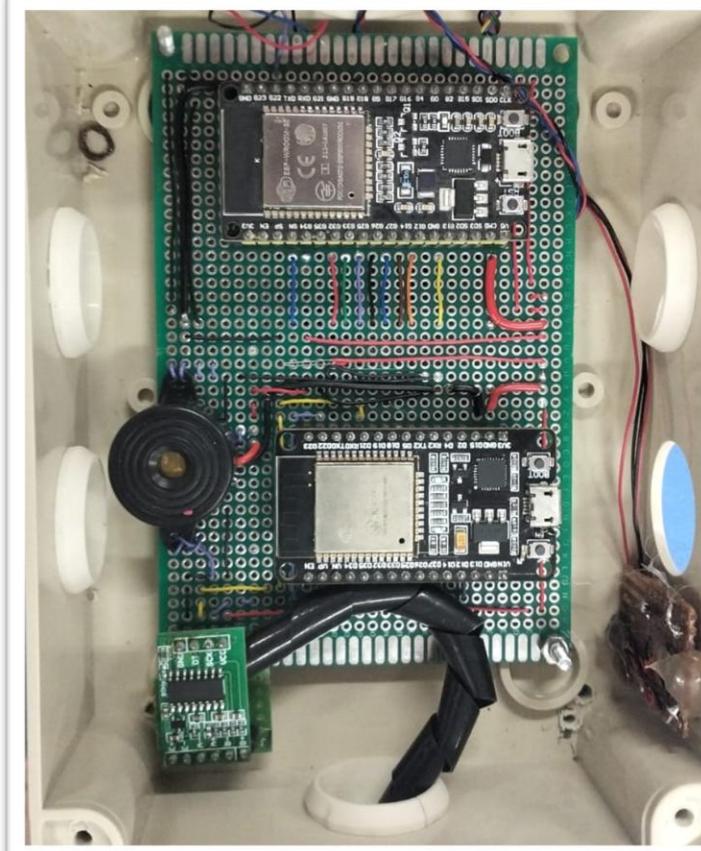
**Gambar 4. 15** Tampilan Nilai TPM Sesuai Deteksi Tetesan Cairan

### 4.3.5 Tes Script Butir-Uji-5

Tabel 4. 5 Pengujian Notifikasi melalui Buzzer

<b>No. Fungsi</b>	TS-05		
<b>Nama Butir Uji</b>	Pengujian Notifikasi melalui Buzzer		
<b>Tujuan Pengujian</b>	Menguji fungsi buzzer yang memberikan notifikasi suara dengan interval tertentu sesuai dengan kondisi berat cairan infus.		
<b>Hasil yang diharapkan</b>	<ul style="list-style-type: none"> <li>• Buzzer berbunyi setiap 1 menit jika berat cairan infus berada antara 135–145 ml.</li> <li>• Buzzer berbunyi setiap 30 detik jika berat cairan infus berada di bawah 135 ml.</li> </ul>		
<b>Tanggal Pengujian</b>	06 Januari 2025		
<b>Penguji</b>	Paian Manalu, Erna Manurung, Batara Nadapdap		
<b>Skenario Uji</b>			
<ol style="list-style-type: none"> <li>1. Sistem sedang memantau volume infus yang masuk melalui load cell.</li> <li>2. Setelah berat cairan infus terdeteksi :           <ul style="list-style-type: none"> <li>• Jika berat antara 135–145 ml, buzzer harus berbunyi setiap 1 menit.</li> <li>• Jika berat di bawah 135 ml, buzzer harus berbunyi setiap 30 detik.</li> </ul> </li> </ol>			
<b>Kriteria Evaluasi Hasil</b>			
<ul style="list-style-type: none"> <li>– Buzzer berbunyi sesuai interval yang ditentukan berdasarkan kondisi berat cairan infus.</li> <li>– Tidak ada notifikasi buzzer jika berat cairan infus melebihi 145 ml.</li> </ul>			
<b>Kasus dan Hasil Uji</b>			
<b>Data Masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>

Berat cairan infus > 145 ml	Tidak ada notifikasi buzzer.	Tidak ada notifikasi buzzer.	Fungsi buzzer berjalan sesuai dengan harapan pada kondisi berat cairan di atas 145 ml.
Berat cairan infus 135–145 ml	Buzzer berbunyi setiap 1 menit.	Buzzer berbunyi setiap 1 menit ketika berat cairan infus berada antara 135–145 ml.	Fungsi buzzer berjalan sesuai dengan harapan pada kondisi berat cairan 135–145 ml.
Berat cairan infus < 135 ml	Buzzer berbunyi setiap 30 detik.	Buzzer berbunyi setiap 30 detik ketika berat cairan infus di bawah 135 ml.	Fungsi buzzer berjalan sesuai dengan harapan pada kondisi berat cairan di bawah 135 ml.
<b>Hasil</b>			



**Gambar 4. 16 Rangkaian Sistem Buzzer**

```
ID Alat: 1624116, Berat: 297 gram, Buzzer: Mati  
Data berhasil terkirim!  
ID Alat: 1624116, Berat: 297 gram, Buzzer: Mati  
Data berhasil terkirim!  
ID Alat: 1624116, Berat: 297 gram, Buzzer: Mati  
Data berhasil terkirim!  
ID Alat: 1624116, Berat: 297 gram, Buzzer: Mati  
Data berhasil terkirim!
```

**Gambar 4. 17 Data Volume Infus dengan Kondisi > 145**

Monitoring Infus								
<a href="#">Home</a> <a href="#">Riwayat Pasien</a> <a href="#">Notifikasi</a> <span style="float: right;">1</span>								
<input type="text" value="Cari Pasien, Ruang, atau Nomor Bed"/>								
<a href="#">Tambah Pasien</a>								
Nama Pasien	Ruang	Nomor Bed	Volume (%)	Tetesan per Menit	Tetesan	Faktor	Status	Aksi
Batara Nadapdap	F	2	37%	17.00	0	Makro	<span style="background-color: green; color: white; padding: 2px 5px;">Aman</span>	<a href="#">Hapus</a> / <a href="#">Edit</a>

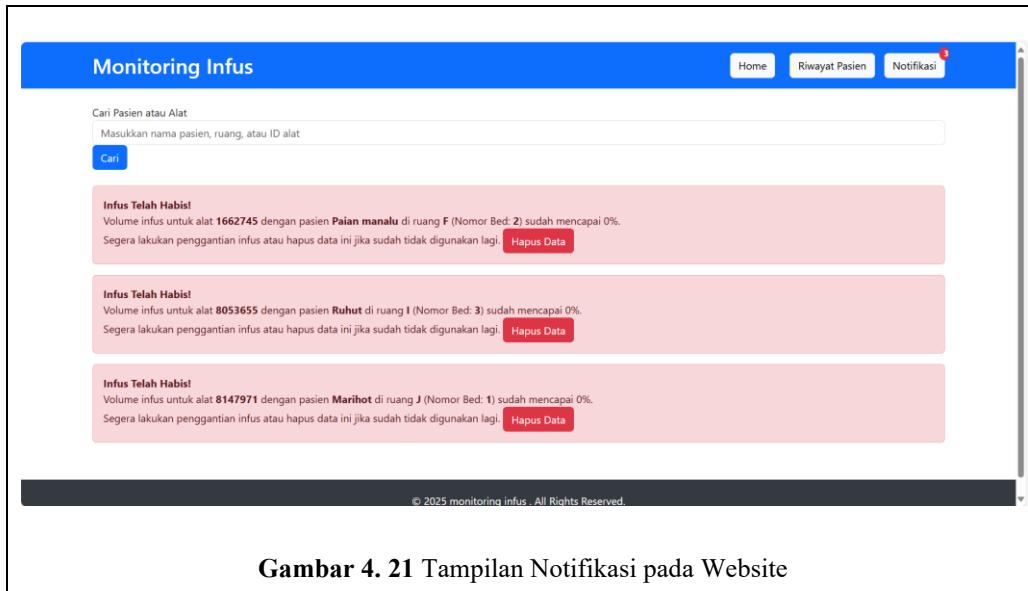
**Gambar 4. 18** Tampilan Website dengan Kondisi Volume Infus 37%

```
Output Serial Monitor X
Message (Ctrl + Enter to send message to 'DOIT ESP32 DEVKIT V1' or
Data berhasil terkirim!
ID Alat: 1624116, Berat: 63 gram, Buzzer: Mati
Data berhasil terkirim!
ID Alat: 1624116, Berat: 63 gram, Buzzer: Mati
Data berhasil terkirim!
ID Alat: 1624116, Berat: 63 gram, Buzzer: Mati
Data berhasil terkirim!
ID Alat: 1624116, Berat: 63 gram, Buzzer: Mati
Data berhasil terkirim!
ID Alat: 1624116, Berat: 63 gram, Buzzer: Mati
Data berhasil terkirim!
ID Alat: 1624116, Berat: 63 gram, Buzzer: Mati
Data berhasil terkirim!
ID Alat: 1624116, Berat: 63 gram, Buzzer: Mati
Data berhasil terkirim!
ID Alat: 1624116, Berat: 63 gram, Buzzer: Bunyi <135
Data berhasil terkirim!
ID Alat: 1624116, Berat: 63 gram, Buzzer: Mati
```

**Gambar 4. 19** Data Volume Infus dengan Kondisi < 135 gram

Monitoring Infus								
<a href="#">Home</a> <a href="#">Riwayat Pasien</a> <a href="#">Notifikasi</a> <span style="float: right;">1</span>								
<input type="text" value="Cari Pasien, Ruang, atau Nomor Bed"/>								
<a href="#">Tambah Pasien</a>								
Nama Pasien	Ruang	Nomor Bed	Volume (%)	Tetesan per Menit	Tetesan	Faktor	Status	Aksi
Batara Nadapdap	F	2	0%	19.00	0	Makro	<span style="background-color: red; color: white; padding: 2px 5px;">Kritis</span>	<a href="#">Hapus</a> / <a href="#">Edit</a>

**Gambar 4. 20** Tampilan Website dengan Kondisi Volume Infus Habis



**Gambar 4. 21** Tampilan Notifikasi pada Website

**Tabel 4. 6** Hasil Pengujian Volume Cairan Infus terhadap Aktivasi Notifikasi

No.	Volume Cairan Infus (ml)	Status Sistem	Interval Aktivasi Buzzer	Notifikasi Web	Deskripsi
1	500	Normal	Tidak Aktif	Tidak	Volume penuh; sistem berjalan dalam kondisi normal.
2	400	Normal	Tidak Aktif	Tidak	Volume masih cukup; alarm tidak diaktifkan
3	300	Normal	Tidak Aktif	Tidak	Tidak ada notifikasi yang dikirimkan
4	145	Peringatan Dini	1 menit	Ya	Volume cairan mulai menurun; notifikasi aktif.
5	140	Peringatan Dini	1 menit	Ya	Sistem meminta penggantian cairan infu

No.	Volume Cairan Infus (ml)	Status Sistem	Interval Aktivasi Buzzer	Notifikasi Web	Deskripsi
6	135	Peringatan Dini	1 menit	Ya	Batas atas zona kritis.
7	130	Peringatan Kritis	30 detik	Ya	Volume rendah; penggantian segera diperlukan.
8	100	Peringatan Kritis	30 detik	Ya	Notifikasi terus aktif.
9	50	Peringatan Kritis	30 detik	Ya	Risiko kehabisan cairan infus.
10	10	Peringatan Kritis	30 detik	Ya	Kondisi kritis; cairan infus hampir habis.

#### 4.3.6 Hasil Pengujian Durasi Habisnya Infus

Pengujian ini dilakukan untuk mengetahui estimasi durasi waktu yang dibutuhkan agar cairan infus dengan volume 500 ml dapat habis, berdasarkan variasi kecepatan tetesan per menit (tpm). Pengujian dilakukan secara sistematis dengan menggunakan set infus makro yang memiliki drip factor sebesar 20 tetes/ml. Artinya, setiap 1 ml cairan infus setara dengan 20 tetes.

Tujuan dari pengujian ini adalah untuk menyediakan data empiris yang dapat digunakan dalam pengaturan kecepatan infus di lingkungan medis, sehingga tenaga kesehatan dapat mengatur waktu pemberian cairan dengan lebih akurat dan aman.

Berdasarkan perhitungan, jumlah total tetesan untuk infus 500 ml dengan faktor tetes 20 tetes/ml adalah:

$$\text{Total tetesan} = 500 \text{ ml} \times 20 \text{ tetes/ml} = 10.000 \text{ tetes}$$

Dari hasil pengujian yang disimulasikan dengan berbagai tingkat kecepatan tetesan, diperoleh estimasi durasi habisnya infus seperti ditunjukkan pada tabel berikut:

**Tabel 4. 7 Tabel Hasil Pengujian Durasi Habisnya Infus**

No.	Tetesan per Menit (TPM)	Volume Infus (ml)	Faktor Tetes (tetes/ml)	Total Tetesan	Estimasi Waktu Habis (menit)	Estimasi Waktu Habis (jam:menit)
1.	20 tpm	500 ml	20 tetes/ml	10.000 tetes	500 menit	8 jam 20 menit
2.	30 tpm	500 ml	20 tetes/ml	10.000 tetes	333 menit	5 jam 33 menit
3.	40 tpm	500 ml	20 tetes/ml	10.000 tetes	250 menit	4 jam 10 menit
4.	60 tpm	500 ml	20 tetes/ml	10.000 tetes	167 menit	2 jam 47 menit
5.	80 tpm	500 ml	20 tetes/ml	10.000 tetes	125 menit	2 jam 5 menit
6.	100 tpm	500 ml	20 tetes/ml	10.000 tetes	100 menit	1 jam 40 menit
7.	120 tpm	500 ml	20 tetes/ml	10.000 tetes	83 menit	1 jam 23 menit

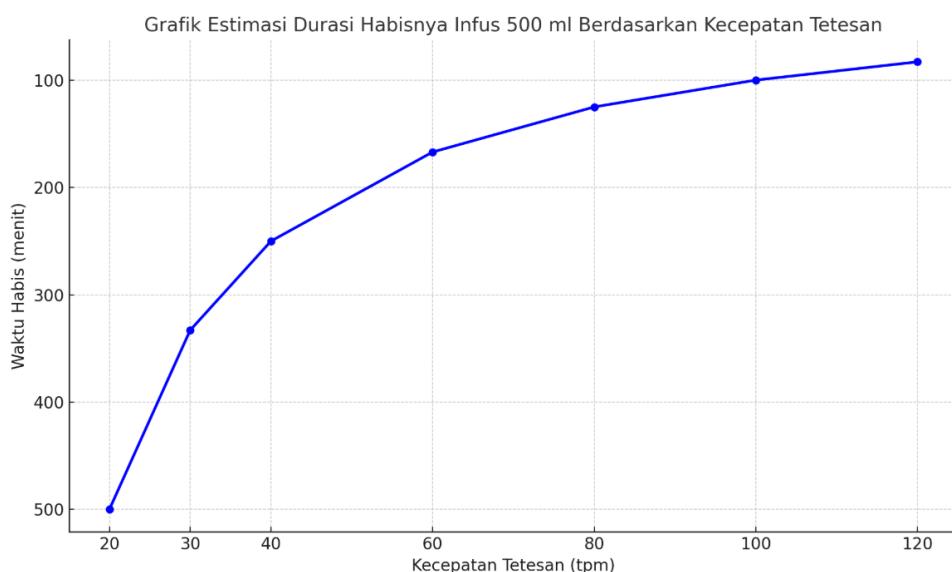
Keterangan :

- **Tpm (tetes per menit)** = jumlah tetesan yang diatur melalui regulator infus.
- **Drip factor** (faktor tetes) tergantung jenis set infus:
  - **Makro:** 20 tetes/ml (umum)
  - **Mikro:** 60 tetes/ml (untuk pediatri atau presisi tinggi)

- **Total tetesan** = volume (ml) × faktor tetes =  $500 \times 20 = 10.000$  tetes
- Estimasi waktu dihitung:

$$\text{Waktu (menit)} = \frac{\text{Total Tetesan}}{\text{Tetesan per Menit (TPM)}}$$

Dari tabel di atas dapat dilihat bahwa semakin tinggi kecepatan tetesan per menit, maka waktu yang dibutuhkan untuk menghabiskan cairan infus menjadi semakin singkat. Sebaliknya, pada kecepatan tetesan yang lebih rendah, durasi pemberian infus menjadi lebih lama.



**Gambar 4. 22** Grafik Estimasi Durasi Habisnya Infus

Grafik diatas menunjukkan hubungan antara kecepatan tetesan (tpm) dan durasi habisnya infus (menit). Grafik menunjukkan bahwa semakin tinggi kecepatan tetesan, semakin cepat infus 500 ml akan habis. Kurva menurun ini menggambarkan hubungan berbanding terbalik antara tpm dan waktu.

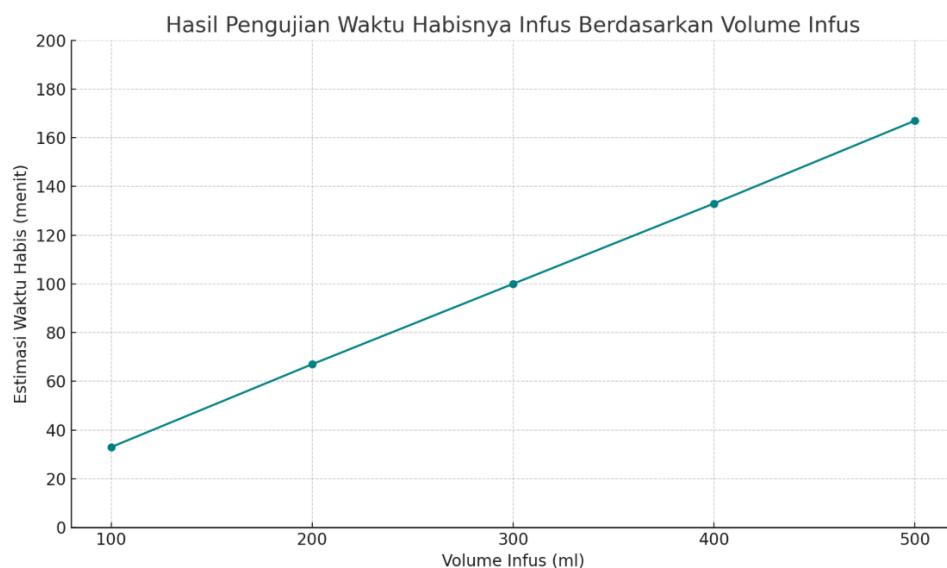
#### 4.3.7 Hasil Pengujian Waktu Habisnya Infus Berdasarkan Volume Infus

Pengujian ini bertujuan untuk mengetahui durasi waktu yang dibutuhkan hingga cairan infus habis, berdasarkan volume infus yang berbeda-beda, dengan kecepatan tetesan dan faktor tetes yang tetap. Kecepatan tetesan yang digunakan dalam

simulasi ini adalah 60 tetes per menit (TPM), dan set infus yang digunakan memiliki faktor tetes 20 tetes/ml

**Tabel 4. 8** Hasil Pengujian Waktu Infus Berdasarkan Volume Infus

No.	Tetesan per Menit (TPM)	Volume Infus (ml)	Faktor Tetes (tetes/ml)	Total Tetesan	Estimasi Waktu Habis (menit)	Estimasi Waktu Habis (jam:menit)
1.	60 tpm	100 ml	20 tetes/ml	2.000 tetes	33 menit	0 jam 33 menit
2.	60 tpm	200 ml	20 tetes/ml	4.000 tetes	67 menit	1 jam 7 menit
3.	60 tpm	300 ml	20 tetes/ml	6.000 tetes	100 menit	1 jam 40 menit
4.	60 tpm	400 ml	20 tetes/ml	8.000 tetes	133 menit	2 jam 13 menit
5.	60 tpm	500 ml	20 tetes/ml	10.000 tetes	167 menit	2 jam 47 menit



**Gambar 4. 23** Hasil Pengujian Waktu Habisnya Infus Berdasarkan Volume Infus

Grafik di atas menunjukkan hubungan yang linear antara volume infus (dalam ml) dan estimasi waktu habisnya infus (dalam menit). Semakin besar volume cairan yang digunakan, semakin lama waktu yang dibutuhkan agar infus habis, dengan asumsi kecepatan tetesan tetap.

## **BAB V**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan berisikan mengenai kesimpulan yang didapat setelah menyelesaikan proyek Tugas Akhir serta saran mengenai proyek Tugas Akhir agar dapat menjadi perbaikan untuk pengembangan kedepannya.

#### **5.1 Kesimpulan**

Berdasarkan Implementasi sistem otomatis dan monitoring infus berbasis IoT, dapat disimpulkan :

1. **Otomatisasi Kontrol Cairan Infus:** Sistem yang dikembangkan berhasil mengotomatisasi pengaturan aliran cairan infus, meminimalkan kesalahan manual, dan memberikan kemudahan dalam pengaturan jarak jauh melalui *website*. Hal ini meningkatkan efisiensi dalam proses perawatan pasien, mengurangi ketergantungan pada tenaga medis, dan meningkatkan akurasi dalam pengaturan aliran infus.
2. **Monitoring Volume Infus:** Sistem mampu memantau volume cairan infus secara real-time dan memberikan notifikasi saat cairan hampir habis, yang memastikan penggantian infus tepat waktu. Dengan adanya fitur ini, perawatan pasien menjadi lebih terjamin dan mengurangi risiko keterlambatan penggantian infus yang dapat berdampak negatif pada pasien.

Secara keseluruhan, sistem otomatis dan monitoring infus berbasis IoT ini dapat meningkatkan kualitas perawatan pasien, efisiensi kerja tenaga medis, serta memberikan solusi yang lebih aman dalam proses pengelolaan infus.

#### **5.2 Saran**

Disarankan untuk mengembangkan sistem ini agar dapat digunakan dengan berbagai jenis kantong infus atau alat medis lainnya. Hal ini akan memperluas penerapan sistem dan memungkinkan penggunaannya dalam berbagai kondisi medis, meningkatkan fleksibilitas dan manfaat sistem di rumah sakit atau fasilitas kesehatan lainnya.

### 5.3 Estimasi Biaya Penelitian

**Tabel 5. 1** Estimasi Biaya

No	Nama Barang	Jumlah	Harga @unit	Total
1	Sensor Load Cell	1	Rp 60.500	Rp 60.500
2	Alat Pengatur Tetesan	1	Rp 250.000	Rp 250.000
3	ESP32	1	Rp 60.000	Rp 60.000
4	Motor Servo	1	Rp 12.500	Rp 12.500
5	Keypad	1	Rp 18.000	Rp 18.000
6	Buzzer	1	Rp 5.000	Rp 5.000
7	Modul I2C LCD	1	Rp 52.000	Rp 52.000
8	Modul HX711	1	Rp 35.000	Rp 35.000
9	PCB	1	Rp 23.000	Rp 23.000
10	Infus	5	Rp 15.000	Rp 75.000
11	Kabel Infus	2	Rp 15.000	Rp 30.000
12	Case	1	Rp 50.000	Rp 50.000
13	Tiang Infus	1	Rp 160.000	Rp 160.000
14	Adaptor	1	Rp 20.000	Rp 20.000
<b>Total Biaya</b>				Rp 851.000

## DAFTAR PUSTAKA

- [1] Siti Nur Khasanah, M. Maisyaroh, A. Nugraha, And M. Ulinnuha, “Pembuatan Alat Monitoring Infus Berbasis Nodemcu Esp8266,” *J. Ilm. Inform.*, Vol. 6, No. 2, Pp. 105–110, 2021, Doi: 10.35316/Jimi.V6i2.1472.
- [2] N. A. Kurniawan, “Alat Monitoring Cairan Infus Berbasis Iot (Internet Of Things),” Pp. 1–74, 2024.
- [3] D. Retno, M. W. Sari, And P. W. Ciptadi, “Pengembangan Sistem Kontrol Dan Monitoring Jumlah Tetesan Infus Pada Pasien Menggunakan Android,” *Semin. Nas. Din. Inform.*, Pp. 150–154, 2021.
- [4] T. D. Hendrawati And R. A. Ruswandi, “Sistem Pemantauan Tetesan Cairan Infus Berbasis Internet Of Things,” Vol. 1, No. 1, Pp. 25–32, 2021.
- [5] S. Youda And J. Sardi, “Rancang Bangun Kontrol Kecepatan Cairan Infus Berbasis Arduino Uno,” *Ranah Res. J. Multidiscip. Res. Dev.*, Vol. 5, No. 1, Pp. 23–32, 2022, Doi: 10.38035/Rrj.V5i1.576.
- [6] P. Zulfikr I, Faoji, “Sistem Monitoring : Pengertian, Jenis, Dan Tujuan.” [Online]. Available: <Https://Testindo.Co.Id/Sistem-Monitoring-System-Pengertian/>
- [7] S. Purwiyanti, H. Fitriwan, S. R. Sulistiyanti, And D. Rahmat, “Sistem Nirkabel Otomatis Untuk Memantau Tinggi Cairan Infus Dan Darah Pada Selang Infus,” *Semin. Nas. Tek. Elektro*, No. 1, 2023.
- [8] N. Kirana, “Memahami Konsep Sistem Monitoring Berbasis Iot Dalam Dunia Bisnis.” [Online]. Available: <Https://Synapsis.Id/Konsep-Sistem-Monitoring-Berbasis-Iot.Html>
- [9] S. Madakam, R. Ramaswamy, And S. Tripathi, “Internet Of Things (Iot): A Literature Review,” *J. Comput. Commun.*, Vol. 03, No. 05, Pp. 164–173, 2015, Doi: 10.4236/Jcc.2015.35021.
- [10] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, And K. S. Kwak, “The Internet Of Things For Health Care: A Comprehensive Survey,” *Ieee Access*, Vol. 3, Pp. 678–708, 2015, Doi: 10.1109/Access.2015.2437951.
- [11] Y. E. K. Harlin, “Rancang Bangun Emergency Respons Cairan Infus Berbasis Internet Of Things Dan Tetes Infus Otomatis,” P. 12, 2019.

- [12] T. A. Utami, “Kebutuhan Nutrisi Dan Cairan Modul Pembelajaran Keperawatan Anak,” *Stik-Sintcarolus.Ac.Id*, Pp. 1–32, 2020, [Online]. Available: [Http://Repository.Stik-Sintcarolus.Ac.Id/1152/1/Kebutuhan\\_Nutrisi\\_Dan\\_Cairan.Pdf](Http://Repository.Stik-Sintcarolus.Ac.Id/1152/1/Kebutuhan_Nutrisi_Dan_Cairan.Pdf)
- [13] “Rumus Tetesan Infus, Faktor Tetes Makro-Mikro.” [Online]. Available: <Https://Www.Detik.Com/Edu/Detikpedia/D-7205131/Rumus-Tetesan-Infus-Faktor-Tetes-Makro-Mikro-Dan-Contoh-Soalnya>
- [14] “Konversi Dalam Bentuk Persentase”, [Online]. Available: [Https://Www.Gramedia.Com/Best-Seller/Cara-Menghitung-Persen/?Srsltid=Afmboopv9kxt\\_Xfwqmw98stkr-Zztx9kcknu5c7rsatooygfkvhk1ho](Https://Www.Gramedia.Com/Best-Seller/Cara-Menghitung-Persen/?Srsltid=Afmboopv9kxt_Xfwqmw98stkr-Zztx9kcknu5c7rsatooygfkvhk1ho)
- [15] “Esp 32.” [Online]. Available: [Https://M.Media-Amazon.Com/Images/I/61jiwy9v5fl.\\_Ac\\_Sx300\\_Sy300\\_Ql70\\_Fmwebp\\_.Jpg](Https://M.Media-Amazon.Com/Images/I/61jiwy9v5fl._Ac_Sx300_Sy300_Ql70_Fmwebp_.Jpg)
- [16] “Sensor Loadcell.”
- [17] O. O. E. Ajibola, O. O. Sunday, And D. O. Eyehorua, “Development Of Automated Intravenous Blood Infusion Monitoring System Using Load Cell Sensor,” *J. Appl. Sci. Environ. Manag.*, Vol. 22, No. 10, P. 1557, 2018, Doi: 10.4314/Jasem.V22i10.04.
- [18] “Modul Hx711.” [Online]. Available: [Https://Tse1.Mm.Bing.Net/Th?Id=Oip.U5lxm\\_Dgjhmb2rqnj9gt7qhaeo&Pid=Api&P=0&H=220](Https://Tse1.Mm.Bing.Net/Th?Id=Oip.U5lxm_Dgjhmb2rqnj9gt7qhaeo&Pid=Api&P=0&H=220)
- [19] R. G. Riyansyah, D. Wahiddin, And ..., “Smart Monitoring Alat Infus Pasien Berbasis Internet Of Things (Iot) Menggunakan Mikrokontroler Esp32,” ... *Student J.* ..., Vol. Ii, Pp. 142–148, 2021, [Online]. Available: <Http://Journal.Ubpkarawang.Ac.Id/Mahasiswa/Index.Php/Ssj/Article/Download/237/167>
- [20] B. Wijayanto, A. Hermawan, And L. Marlinda, “Automated Infusion Monitoring Device Using Arduino-Based Iot (Internet Of Things),” *J. Comput. Networks, Archit. High Perform. Comput.*, Vol. 5, No. 2, Pp. 590–598, 2023, Doi: 10.47709/Cnahpc.V5i2.2594.

- [21] “Photodiode.” [Online]. Available: <Https://Techiesms.Com/Wp-Content/Uploads/2023/02/Ir--1024x1024.Png>
- [22] R. M. , A. M. , Uray Ristian, “—Sistem Monitoring Dan Peringatan Pada Volume Cairan Intravena (Infus) Pasien Menggunakan Arduino Berbasis Website,” *Coding J. Komput. Dan Apl.*, Vol. 7, No. 03, 2019, Doi: 10.26418/Coding.V7i03.37170.
- [23] “Keypad.” [Online]. Available: <Https://Tse3.Mm.Bing.Net/Th?Id=Oip.Embinlhgnc7a9ie219ml6qhajq&Pid=Api&P=0&H=220>
- [24] “Motor Servo.” [Online]. Available: <Https://Tse2.Mm.Bing.Net/Th?Id=Oip.Lswr0gg8owosmtnrf64cdqhaha&Pid=Api&P=0&H=220>
- [25] “Buzzer.” [Online]. Available: [Https://Tse1.Mm.Bing.Net/Th?Id=Oip.Dhbab9wbq19p\\_Wticlt1dahaha&Pid=Api&P=0&H=220](Https://Tse1.Mm.Bing.Net/Th?Id=Oip.Dhbab9wbq19p_Wticlt1dahaha&Pid=Api&P=0&H=220)
- [26] “Modul I2c Lcd.” [Online]. Available: <Https://Tse1.Mm.Bing.Net/Th?Id=Oip.Hvhavhfafiz6ic6cxnouuqaaaa&Pid=Api&P=0&H=220>
- [27] “Printed Circuit Web.” [Online]. Available: [Https://Toko.Bali-Electro.Com/1097-Large\\_Default/Lubang-Paket-Pcb-Hijau.Jpg](Https://Toko.Bali-Electro.Com/1097-Large_Default/Lubang-Paket-Pcb-Hijau.Jpg)
- [28] “Arduino Ide.” [Online]. Available: [Https://Content.Arduino.Cc/Assets/Arduino\\_Logo\\_1200x630-01.Png](Https://Content.Arduino.Cc/Assets/Arduino_Logo_1200x630-01.Png)
- [29] A. B. Pratomo And R. S. Perdana, “Arduviz, A Visual Programming Ide For Arduino,” *Proc. 2017 Int. Conf. Data Softw. Eng. Icodse 2017*, Vol. 2018-Janua, Pp. 1–6, 2017, Doi: 10.1109/Icodse.2017.8285871.
- [30] “Mysql.” [Online]. Available: <Https://Tse3.Mm.Bing.Net/Th?Id=Oip.Wsus-9qxz6ayyvi8lj5lwhagb&Pid=Api&P=0&H=220>
- [31] Dewaweb, “Apa Itu Mysql? Cara Kerja, Fungsi, Kelebihan & Kekurangannya.” [Online]. Available: [Https://Www.Dewaweb.Com/Blog/Apa-Itu-Mysql/#Apa\\_Itu\\_Mysql](Https://Www.Dewaweb.Com/Blog/Apa-Itu-Mysql/#Apa_Itu_Mysql)
- [32] “Vscode.” [Online]. Available:

- Https://Tse1.Mm.Bing.Net/Th?Id=Oip.Uugf-  
Abd871n9wfmjcvnqwhaek&Pid=Api&P=0&H=220
- [33] P. Priya, “What Is Visual Studio Code? Practical Guide(2023).” [Online]. Available: <Https://Www.Educba.Com/What-Is-Visual-Studio-Code/>
- [34] “Mqtt.” [Online]. Available: <Https://Mqtt.Org/Mqtt-Specification/>
- [35] “Mosquitto.” [Online]. Available: <Https://Projects.Eclipse.Org/Projects/Iot.Mosquitto>
- [36] “Node-Red.” [Online]. Available: <Https://Nodered.Org/About/Resources/Media/Node-Red-Icon-2.Png>
- [37] P. A. Rosyady, A. S. S. Sukarjiana, N. U. Habibah, N. Ihsana, A. R. C. Baswara, And W. R. Dinata, “Monitoring Cairan Infus Menggunakan Load Cell Berbasis Internet Of Things (Iot),” *Techné J. Ilm. Elektrotek.*, Vol. 22, No. 1, Pp. 97–110, 2023, Doi: 10.31358/Techne.V22i1.345.
- [38] F. M. Yoga, A. Shahib Al Banna, A. Thariq Saputra, J. Teknik Elektro, And P. H. Negeri Banjarmasin Jl Brigjen Hasan Basri, “Sistem Kontrol Dan Monitoring Infus Berbasis Nodemcu,” *Semin. Nas. Terap. Ris. Inov. Ke-6 Isas Publ. Ser. Eng. Sci.*, Vol. 6, No. 1, Pp. 372–378, 2020.
- [39] O. : Septian And P. Aji, “Alat Monitoring Tetesan Infus...(Septian Prastyo Aji) 78 Infusing Monitoring Tools Using Web Online Based Esp8266 With Arduino Ide Programming,” Vol. 7, No. 1, Pp. 78–86, 2018.
- [40] S. A. Kadiran, E. Supriyanto, And M. Y. Maghribi, “Sistem Monitoring Dan Controlling Cairan Infus Berbasis Website,” *J. Ris. Rekayasa Elektro*, Vol. 5, No. 1, P. 57, 2023, Doi: 10.30595/Jrre.V5i1.17743.
- [41] M. Rakadytia Akbar And A. Stefanie, “Sistem Monitoring Infus Menggunakan Hx 711 Untuk Membantu Perawat,” *Jimpis J. Ilm. Mhs. Pendidik. Sej.*, Vol. 8, No. 3, 2023, [Online]. Available: <Http://Jim.Unsyiah.Ac.Id/Sejarah/Mm>
- [42] W. Wahyudi, A. Rahman, And M. Nawawi, “Perbandingan Nilai Ukur Sensor Load Cell Pada Alat Penyortir Buah Otomatis Terhadap Timbangan Manual,” *Elkomika J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, Vol. 5, No. 2, P. 207, 2018, Doi: 10.26760/Elkomika.V5i2.207.

## LAMPIRAN

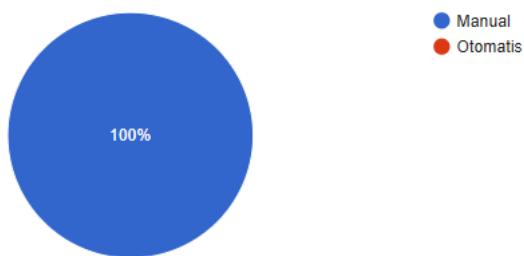
Berikut sistem infus yang digunakan oleh RSUD Porsea

### Sistem Infus yang Digunakan

Sistem infus apa yang saat ini digunakan di rumah sakit Anda?

17 jawaban

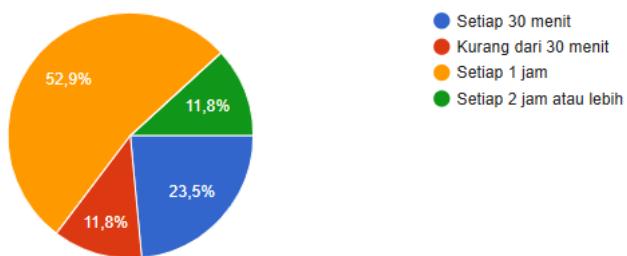
 Salin diagram



Seberapa sering perawat mengecek kondisi infus secara manual?

17 jawaban

 Salin diagram



Berikut respon perawat terhadap tantangan dalam sistem infus manual yang diperoleh melalui google form yang ditujukan kepada perawat RSUD Porsea.

#### Tantangan dalam Sistem Infus Manual

Apa saja tantangan yang dihadapi perawat saat memantau infus manual?  
(Pilih semua yang relevan)

17 jawaban

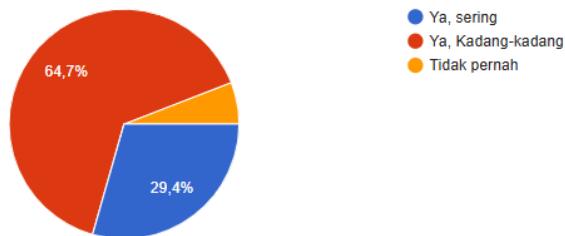
 Salin diagram



Apakah pernah terjadi keterlambatan penggantian infus karena perawat terlambat menyadari infus hampir habis?

17 jawaban

 Salin diagram



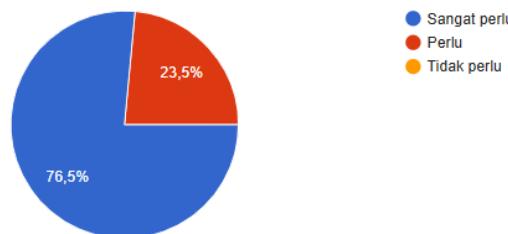
Berikut kebutuhan dan harapan perawat dalam penerapan sistem infus otomatis.

### Kebutuhan dan Harapan

Menurut Anda, apakah rumah sakit perlu menerapkan sistem infus otomatis yang dapat membantu mengatur tetesan cairan infus?

17 jawaban

 Salin diagram



### Apa harapan Anda dari sistem infus otomatis?

17 jawaban

 Salin diagram



Berikut adalah perbandingan waktu yang dibutuhkan oleh perawat untuk mengatur tetesan infus pada satu pasien secara manual dan menggunakan sistem infus otomatis. Data ini diperoleh melalui Google Form yang ditujukan kepada perawat di RS Awal Bros, RS Paru, RS Murni Teguh, dan RS Adam Malik.

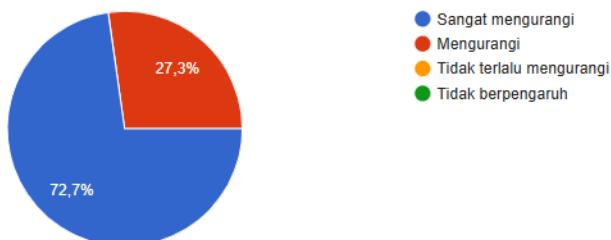
Berapa lama waktu yang Anda butuhkan untuk mengatur tetesan infus secara manual pada satu pasien?	Berapa lama waktu yang Anda butuhkan untuk mengatur pump infus pada satu pasien?
1 menit	20 detik
30 Detik	30 Detik
3 menit	1 menit
1 menit	Kurang dari 1 menit
Kalau betul-betul pas lebih kurang 20 detik,	10 detik
1 Menit	20 Detik
10 menit	5 menit
20 detik	10 detik
1menit	Lebih kurang 20 detik
2 menit	1 menit
15 detik	Kurang dari 10 detik
1- 3 menit	1 menit
15 detik	5 detik
Kurang lebih 1-3 menit	Kurang lebih 1 menit
2 menit	1 menit

Berikut adalah data mengenai Kebutuhan Notifikasi dan Pengaruh Laju Cairan Infus pada Sistem Monitoring Infus Berbasis IoT yang diperoleh melalui google form.

Apakah Anda merasa notifikasi otomatis dapat mengurangi beban kerja perawat?

 Salin diagram

11 jawaban

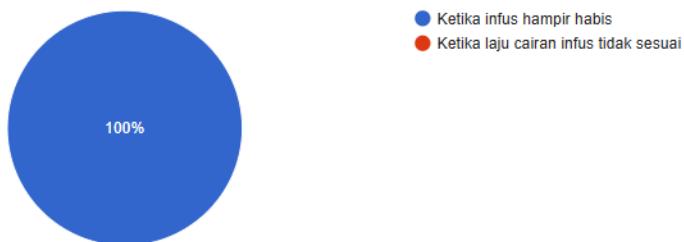


- Sangat mengurangi
- Mengurangi
- Tidak terlalu mengurangi
- Tidak berpengaruh

Dalam situasi apa notifikasi otomatis paling dibutuhkan menurut Anda?

 Salin diagram

11 jawaban



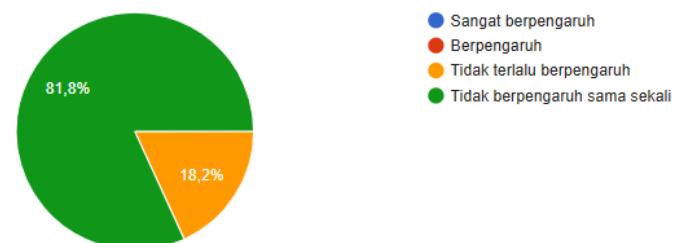
- Ketika infus hampir habis
- Ketika laju cairan infus tidak sesuai

### Laju Cairan Infus dan Dampaknya

Apakah menurut pengalaman Anda, laju cairan infus berpengaruh terhadap darah naik ke dalam selang infus?

 Salin diagram

11 jawaban

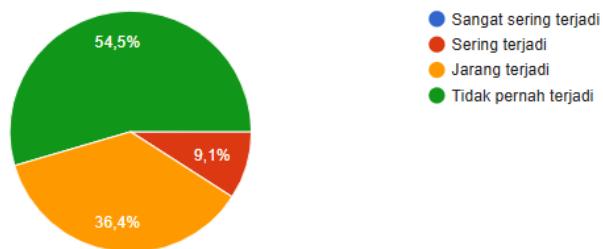


- Sangat berpengaruh
- Berpengaruh
- Tidak terlalu berpengaruh
- Tidak berpengaruh sama sekali

1	Faktor apa saja yang paling memengaruhi risiko terjadinya darah naik ke dalam selang infus?	<input type="button" value="▼"/>
2	Ketika pasien melakukan pergerakan contohnya pasien ingin ke kamar mandi namun infusnya tidak dimatikan	
3	Saat pasien ingin ke kamar mandi sementara infus nya tidak dimatikan terlebih dahulu	
4	Pergerakan pasien	
5	Ketika pasien mengubah posisi tubuh seperti bangun dari posisi tidur ke duduk, atau bergerak tiba-tiba, aliran cairan infus bisa terhambat sementara, menciptakan kondisi di mana tekanan darah lebih tinggi daripada tekanan cairan infus, sehingga darah dapat terdorong kembali ke selang.	
6	Menekuk lengan atau pergelangan tangan di mana kateter infus dipasang dapat menyebabkan Penyempitan atau lipatan selang infus	
7	Pasien melakukan pergerakan sehingga menyebabkan darah naik ke dalam selang infus	
8	Kelasakan pasien	
9	Pergerakan pasien	
10	Saat pasien tidak mematikan infus saat hendak bergerak ke kamar mandi	
11	Gerakan pasien yang terlalu lasak	
12	Pergerakan pasien yang berlebihan	

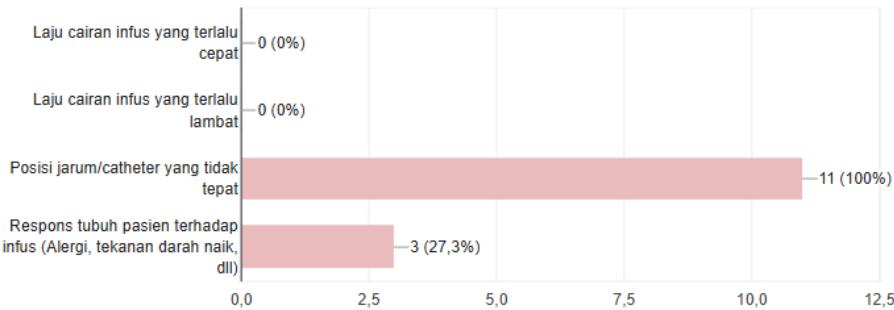
Apakah menurut pengalaman Anda, laju cairan infus yang terlalu cepat dapat menyebabkan pembengkakan di area infus?

11 jawaban



Faktor apa saja yang menurut Anda paling memengaruhi risiko pembengkakan di area infus? (Anda dapat memilih lebih dari satu)

11 jawaban

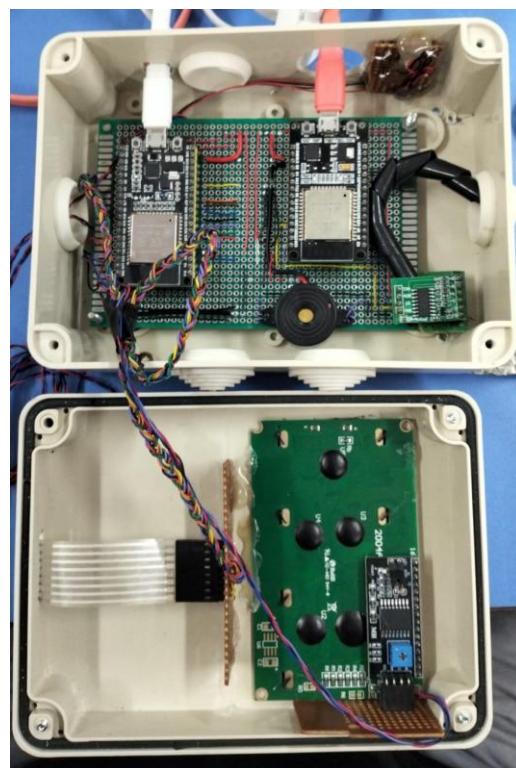


Implementasi hasil akhir alat :

a. Implementasi Alat Case Luar



b. Implementasi Alat Case Dalam



### Kode Program Kalibrasi Tetesan :

```
1 const int sensorPin = 34;

2 volatile bool sensorStatus = false; // Menyimpan status sensor

3 unsigned long lastDebounceTime = 0;

4 const unsigned long debounceDelay = 50;

5 int detectionCount = 0; // Menyimpan jumlah deteksi sensor

6 unsigned long startTime = 0; // Menyimpan waktu deteksi pertama

7 unsigned long lastMinuteTime = 0; // Waktu saat deteksi menit terakhir

8 int dropletsPerMinute = 0; // Jumlah tetesan per menit

9 void IRAM_ATTR countDroplet() {

10 unsigned long currentTime = millis();

11 if (currentTime - lastDebounceTime > debounceDelay) {

12 sensorStatus = true; // Sensor terdeteksi (nilai 1)

13 lastDebounceTime = currentTime;

14 }

15 }

16 void setup() {

17 Serial.begin(115200); // Kecepatan baud yang lebih tinggi untuk komunikasi cepat

18 pinMode(sensorPin, INPUT_PULLUP);

19 attachInterrupt(digitalPinToInterrupt(sensorPin), countDroplet, FALLING);

20 Serial.println("Sensor tetesan infus siap...");

21 }

22 void loop() {

23 static unsigned long lastPrintTime = 0;
```

```
24 unsigned long currentTime = millis();  
  
25 if (currentTime - lastPrintTime >= 10) { // Setiap 10 ms, untuk  
    respons cepat  
  
26 lastPrintTime = currentTime;  
  
27 if (sensorStatus) {  
  
28 detectionCount++; // Meningkatkan penghitung setiap kali sensor  
    terdeteksi  
  
29 // Jika ini adalah deteksi pertama, simpan waktu deteksi pertama  
30 if (detectionCount == 1) {  
  
    a. startTime = currentTime; // Reset waktu deteksi pertama  
  
31 }  
  
32 // Hitung waktu relatif dari deteksi pertama dalam detik  
  
33 unsigned long elapsedTime = (currentTime - startTime) / 1000; //  
    Menghitung waktu dalam detik  
  
34 // Menampilkan tetesan dan waktu relatif dalam detik, dimulai  
    dari detik 1  
  
35 Serial.print("Tetesan ke-");  
  
36 Serial.print(detectionCount);  
  
37 Serial.print(" pada detik: ");  
  
38 Serial.println(elapsedTime + 1); // Tampilkan detik, dimulai  
    dari detik 1  
  
39 sensorStatus = false; // Reset status sensor setelah print  
40 }  
  
41 // Menghitung jumlah tetesan per menit  
42 if (currentTime - lastMinuteTime >= 60000) { // Setiap 60 detik
```

```

43 lastMinuteTime = currentTime; // Update waktu untuk satu menit
berikutnya

44 // Menampilkan jumlah tetesan per menit

45 Serial.print("Tetesan per menit: ");

46 Serial.println(dropletsPerMinute);

47 // Reset penghitung tetesan per menit

48 dropletsPerMinute = 0;

49 // Reset penghitung deteksi dan waktu awal agar pembacaan tetesan
dimulai lagi dari 1

50 detectionCount = 0;

51 startTime = currentTime; // Reset waktu deteksi pertama

52 }

53 // Meningkatkan jumlah tetesan per menit

54 dropletsPerMinute = detectionCount;

55 }

56 }

```

#### **Kode Program Kalibrasi Keypad untuk TPM :**

```

1 #include <Keypad.h>

2 // Definisi jumlah baris dan kolom pada keypad

3 const byte ROW_NUM = 4; // Jumlah baris

4 const byte COLUMN_NUM = 4; // Jumlah kolom

5 // Map tombol keypad

6 char keys[ROW_NUM][COLUMN_NUM] = {

7 {'1', '2', '3', 'A'},

8 {'4', '5', '6', 'B'},

```

```

9  {'7', '8', '9', 'C'},
10 {'*', '0', '#', 'D'}
11 };

12 // Definisi pin baris dan kolom pada ESP32

13 byte pin_rows[ROW_NUM] = {13, 12, 14, 27}; // Sesuaikan dengan
pin ESP32 Anda

14 byte pin_column[COLUMN_NUM] = {26, 25, 33, 32}; // Sesuaikan
dengan pin ESP32 Anda

15 // Inisialisasi keypad

16 Keypad keypad = Keypad(makeKeymap(keys), pin_rows, pin_column,
ROW_NUM, COLUMN_NUM);

17 // Variabel untuk menyimpan inputan

18 float volume = 0.0;
19 float faktor = 0.0;
20 float waktu = 0.0;

21 void setup() {
22 Serial.begin(115200); // Inisialisasi komunikasi serial
23 Serial.println("Keypad Input Program");
24 Serial.println("Tekan A untuk mulai input data.");
25 }

26 void loop() {
27 char key = keypad.getKey(); // Membaca tombol yang ditekan

28 // Menunggu sampai tombol A ditekan untuk mulai input data
29 if (key == 'A') {
30 // Membersihkan layar

```

```
31 Serial.flush();
32 Serial.println("\033[2J");
33 Serial.println("Masukkan data berikut:");
34 // Menunggu input volume
35 volume = getInput("Masukkan Volume (mL): "); // Fungsi untuk menerima input
36 // Menunggu input faktor
37 faktor = getInput("Masukkan Faktor: "); // Fungsi untuk menerima input
38 // Menunggu input waktu
39 waktu = getInput("Masukkan Waktu (jam): "); // Fungsi untuk menerima input
40 // Menghitung TPM
41 float tpm = (volume * faktor) / (waktu * 60);
42 // Membulatkan TPM ke nilai bulat
43 int tpmRounded = round(tpm);
44 // Menampilkan hasil perhitungan TPM
45 Serial.println("\nHasil perhitungan TPM:");
46 Serial.print("Volume: ");
47 Serial.println((int)volume); // Menampilkan volume sebagai angka bulat
48 Serial.print("Faktor: ");
49 Serial.println((int)faktor); // Menampilkan faktor sebagai angka bulat
50 Serial.print("Waktu (jam): ");
51 Serial.println((int)waktu); // Menampilkan waktu sebagai angka bulat
```

```
52 Serial.print("TPM (dibulatkan): ");

53 Serial.println(tpmRounded); // Menampilkan hasil bulat tanpa
desimal

54 delay(2000); // Tunggu 2 detik sebelum menerima input lagi
55 }
56 }

57 // Fungsi untuk menerima input angka dari keypad
58 float getInput(String prompt) {
59 String input = ""; // Menyimpan input
60 char key;

61 // Menampilkan prompt di Serial Monitor
62 Serial.println(prompt);

63 while (true) {
64 key = keypad.getKey(); // Membaca tombol yang ditekan

65 if (key) {
66 if (key == '#') { // Menekan '#' untuk selesai
a. break;

67 } else if (key == '') { // Menekan '' untuk menghapus input dan
reset
a. input = "";
b. Serial.println("Input dibersihkan. Mulai ulang dari
Volume.");
c. return -1; // Kembalikan -1 untuk menandakan reset dan
mulai lagi dari volume
68 } else {
a. input += key; // Menambahkan digit yang ditekan ke input
```

```
b. Serial.print(key);

69 }
70 }
71 }

72 // Mengubah input string menjadi angka float
73 return input.toFloat();
74 }
```

#### Kode Program Kalibrasi Keypad , TPM dan Servo :

```
1. #include <Keypad.h>

2. #include <ESP32Servo.h> // Library untuk mengontrol servo

3.

4. const byte ROW_NUM = 4;

5. const byte COLUMN_NUM = 4;

6.

7. char keys[ROW_NUM][COLUMN_NUM] = {
8.     {'1', '2', '3', 'A'},
9.     {'4', '5', '6', 'B'},
10.    {'7', '8', '9', 'C'},
11.    {'*', '0', '#', 'D'}
12.};

13.

14. byte pin_rows[ROW_NUM] = {13, 12, 14, 27};
15. byte pin_column[COLUMN_NUM] = {26, 25, 33, 32};

16.
```

```
17. Keypad keypad = Keypad(makeKeymap(keys), pin_rows, pin_column,
   ROW_NUM, COLUMN_NUM);

18.

19. float volume = 0.0, faktor = 0.0, waktu = 0.0;

20.

21. Servo myServo;

22. const int servoPin = 5;

23.

24. int TpM_values[] = {15, 17, 19, 21, 25, 28, 33, 39, 42, 51, 53,
   55, 58, 63, 66};

25. int angle_values[] = {31, 30, 29, 28, 27, 26, 25, 24, 23, 22,
   21, 20, 19, 18, 17};

26. int size = sizeof(TpM_values) / sizeof(TpM_values[0]);

27.

28. void setup() {

29.   Serial.begin(115200);

30.   myServo.attach(servoPin);

31.   myServo.write(0);

32.   Serial.println("Sistem siap. Tekan tombol untuk memulai.");

33. }

34.

35. void loop() {

36.   char key = keypad.getKey();

37.

38.   if (key) {

39.     switch (key) {
```

```
40.     case 'A':
41.         inputData();
42.         break;
43.     case 'B':
44.         moveServoTo(0);
45.         break;
46.     case '*':
47.         resetSystem();
48.         break;
49.     }
50. }
51. }
52.
53. void inputData() {
54.     volume = getInput("Masukkan Volume (mL): ");
55.     if (volume == -1) return;
56.
57.     faktor = getInput("Masukkan Faktor: ");
58.     if (faktor == -1) return;
59.
60.     waktu = getInput("Masukkan Waktu (jam): ");
61.     if (waktu == -1) return;
62.
63.     float tpm = (volume * faktor) / (waktu * 60);
64.     int tpmRounded = round(tpm);
```

```
65. Serial.print("TPM: ");
66. Serial.println(tpmRounded);
67.
68. int targetAngle = mapTpMToAngle(tpmRounded);
69. moveServoTo(targetAngle);
70. }
71.
72. float getInput(String prompt) {
73.   Serial.println(prompt);
74.   String input = "";
75.   char key;
76.
77.   while (true) {
78.     key = keypad.getKey();
79.     if (key) {
80.       if (key == '#') {
81.         break;
82.       } else if (key == '*') {
83.         Serial.println("Input dibersihkan.");
84.         return -1;
85.       } else {
86.         input += key;
87.         Serial.print(key);
88.       }

```

```
89.    }
90. }
91. return input.toFloat();
92. }
93.
94. void moveServoTo(int targetAngle) {
95.     int currentAngle = myServo.read();
96.     int step = (targetAngle > currentAngle) ? 1 : -1;
97.
98.     unsigned long prevTime = millis();
99.     while (currentAngle != targetAngle) {
100.         if (millis() - prevTime >= 10) {
101.             currentAngle += step;
102.             myServo.write(currentAngle);
103.             prevTime = millis();
104.         }
105.     }
106.     Serial.print("Servo ke sudut: ");
107.     Serial.println(targetAngle);
108. }
109.
110. int mapTpMToAngle(int tpm) {
111.     for (int i = 0; i < size; i++) {
112.         if (tpm == TpM_values[i]) return angle_values[i];
113.     }
}
```

```

114.     return 0;
115. }
116. void resetSystem() {
117.     Serial.println("Sistem direset.");
118.     volume = faktor = waktu = 0.0;
119.     myServo.write(0);
120. }
```

**Kode Program Pengujian Durasi Habisnya Satu Botol Infus Berdasarkan Jumlah Tetesan:**

```

1 const int sensorPin = 34;
2 volatile bool sensorStatus = false; // Menyimpan status sensor
3 unsigned long lastDebounceTime = 0;
4 const unsigned long debounceDelay = 50;
5 int detectionCount = 0; // Menyimpan jumlah deteksi sensor
6 unsigned long startTime = 0; // Menyimpan waktu deteksi pertama

7 void IRAM_ATTR countDroplet() {
8     unsigned long currentTime = millis();
9     if (currentTime - lastDebounceTime > debounceDelay) {
10         sensorStatus = true; // Sensor terdeteksi (nilai 1)
11         lastDebounceTime = currentTime;
12     }
13 }

14 void setup() {
15     Serial.begin(115200); // Kecepatan baud yang lebih tinggi untuk komunikasi cepat
```

```
16 pinMode(sensorPin, INPUT_PULLUP);

17 attachInterrupt(digitalPinToInterrupt(sensorPin),
    countDroplet, FALLING);

18 Serial.println("Sensor tetesan infus siap...");

19 }

20 void loop() {

21 static unsigned long lastPrintTime = 0;

22 unsigned long currentTime = millis();

23 if (currentTime - lastPrintTime >= 10) { // Setiap 10 ms,
    untuk respons cepat

24 lastPrintTime = currentTime;

25 if (sensorStatus) {

26 detectionCount++; // Meningkatkan penghitung setiap kali
    sensor terdeteksi

27 // Jika ini adalah deteksi pertama, simpan waktu deteksi
    pertama

28 if (detectionCount == 1) {

        a. startTime = currentTime; // Reset waktu deteksi
            pertama

29 }

30 // Hitung waktu relatif dari deteksi pertama dalam detik

31 unsigned long elapsedTime = (currentTime - startTime) /
    1000; // Menghitung waktu dalam detik

32 // Menampilkan deteksi dan waktu relatif dalam detik,
    dimulai dari detik 1

33 Serial.print("Deteksi ke-");

34 Serial.print(detectionCount);
```

```
35 Serial.print(" pada detik: ");
36 Serial.println(elapsedTime + 1); // Tampilkan detik,
dimulai dari detik 1

37 sensorStatus = false; // Reset status sensor setelah print
38 }
39 }
40 }
```

## Jadwal Penelitian

No	Kegiatan	Jadwal																
		Agustus			September			Oktober			November			Desember			Januari	
		Minggu Ke -																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1.	Pembagian Kelompok dan Dosen Pembimbing																	
2.	Penentuan Topik																	
3.	Survei dan Review Jurnal																	
4.	Menulis / Menyusun Bab 1																	
5.	Menulis / Menyusun Bab 2																	
6.	Seminar Proposal																	
7.	Revisi Proposal																	
8.	Menulis / Menyusun Bab 3																	
9.	Seminar TA 1																	
10.	Revisi Seminar TA 1																	
11.	Menulis / Menyusun Bab 4																	
12.	Menulis / Menyusun Bab 5																	
13.	Menulis / Menyusun Bab 6																	
14.	Sidang Akhir																	