

**IMPLEMENTASI PROTOKOL MQTT (*Message Queuing  
Telemetry Transport*) UNTUK MONITORING INFUS PASIEN  
SECARA TERPUSAT**

**SKRIPSI**

**KEMINATAN TEKNIK KOMPUTER**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Sutikno

NIM: 135150300111004



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018**

## PENGESAHAN

IMPLEMENTASI PROTOKOL MQTT (*Message Queuing Telemetry Transport*)  
UNTUK MONITORING INFUS PASIEN SECARA TERPUSAT

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Sutikno

NIM: 135150300111004

Skripsi ini telah diuji dan dinyatakan lulus pada

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Dahnial Syauqy, S.T., M.T., M.Sc.

NIK. 2016078704231002

Rakhmadhany Primananda, S.T, M.Kom

NIK: 2016098604061001

Mengetahui

Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D

NIP. 19710518 200312 1 001

## IDENTITAS TIM PENGUJI

**1. Penguji 1/ Ketua Majelis**

Wijaya Kurniawan, S.T., M.T

NIP: 19820125 201504 1 002

**2. Penguji 2**

Hurriyatul Fitriyah, S.T., M.Sc

NIP: 19851001 201504 2 003



## PERNYATAAN ORISINALITAS

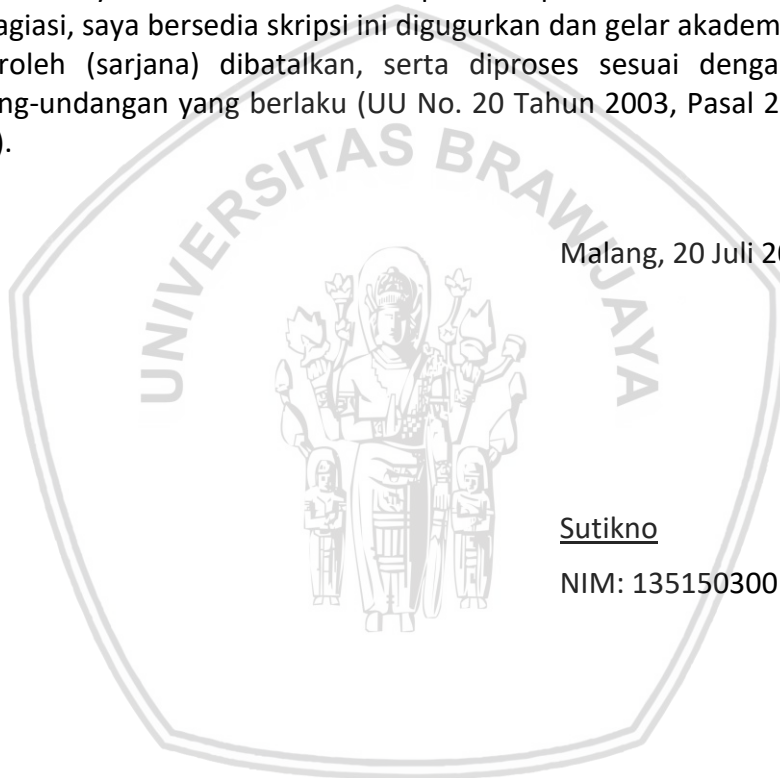
Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 20 Juli 2018

Sutikno

NIM: 135150300111004



## DAFTAR RIWAYAT HIDUP

### DATA PRIBADI

Nama : Sutikno  
Tempat/ Tgl Lahir : Madiun, 28 Juni 1994  
Jenis Kelamin : Laki-laki  
Agama : Islam  
Kewarganegaraan : Indonesia  
Status : Belum Menikah  
Alamat : RT. 07, RW. 01 Dsn. Bruwok, Desa Klumutan,  
Kecamatan Saradan, Kabupaten Madiun  
HP/WA : 082232014358  
Email : tikno.edu@gmail.com

### PENDIDIKAN FORMAL

2001 – 2007 SDN Klumutan 05  
2007 – 2010 SMPN 4 Saradan  
2010 – 2013 SMAN 2 Mejayan



## KATA PENGANTAR

Puji syukur peneliti panjatkan kepada Allah SWT, karena limpahan rahmat dan petunjuk-Nya peneliti dapat menyelesaikan penelitian dan laporan skripsi yang berjudul “IMPLEMENTASI PROTOKOL MQTT (*Message Queuing Telemetry Transport*) UNTUK MONITORING INFUS PASIEN SECARA TERPUSAT”.

Dalam penyusunan dan penulisan skripsi ini tidak lepas dari bantuan moral dan materiil yang diberikan dari berbagai pihak, maka peneliti mengucapkan banyak terima kasih kepada

1. Kedua orang tua dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian, dan kesabarannya memberikan semangat kepada peneliti, serta senantiasa tiada hentinya memberikan doa demi terselesaikannya skripsi ini.
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang.
4. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng. selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang.
5. Bapak Dahnia Syauqy, S.T., M.T., M.Sc. selaku dosen pembimbing I yang telah memberikan pengarahan dan bimbingan kepada peneliti, sehingga dapat menyelesaikan skripsi ini dengan baik.
6. Bapak Rakhmadhany Primananda, S.T, M.Kom selaku dosen pembimbing II yang telah memberikan pengarahan dan bimbingan kepada peneliti, sehingga dapat menyelesaikan skripsi ini dengan baik.
7. Seluruh civitas akademika Fakultas Ilmu Komputer Universitas Brawijaya dan terkhusus untuk teman-teman Teknik Komputer Angkatan 2013 yang telah banyak memberi bantuan dan dukungan selama peneliti menempuh studi di Teknik Komputer Universitas Brawijaya dan selama penyelesaian skripsi ini.
8. Teman-teman kontrakan “Piranha Atas” dan teman-teman yang tidak bisa disebutkan satu persatu, yang selalu membantu dan memberikan berbagai bantuan saat proses penulisan atau pengerjaan skripsi ini.

Peneliti menyadari bahwa dalam penyusunan skripsi ini masih terdapat banyak kekurangan yang perlu disempurnakan, sehingga saran dan kritik yang membangun sangat peneliti harapkan untuk ke depannya menjadi lebih baik lagi. Akhir kata peneliti berharap skripsi ini dapat membawa manfaat bagi semua pihak yang membutuhkannya.

Malang, 8 Mei 2018

Sutikno

## ABSTRAK

Jumlah pasien dengan petugas medis yang kurang seimbang menyebabkan persoalan baru. Salah satunya dalam memantau cairan infus pasien. Berdasarkan kasus yang sudah terjadi, diperlukan sebuah sistem untuk memonitoring cairan infus pasien ketika hampir habis sehingga petugas tidak terlambat dalam penggantian cairan infus kepada pasien. Untuk membantu mengatasi hal tersebut, dilakukan perancangan sistem untuk memonitoring infus pasien secara terpusat dengan menggunakan pengiriman MQTT (*Message Queuing Telemetry Transport*). Pembacaan tetesan dilakukan dengan menggunakan sensor photodiode dengan meletakkan sensor pada *chamber* infus. Pemrosesan data pada sensor node dilakukan dengan menggunakan NodeMCU dan pada server menggunakan PC/Laptop dengan memanfaatkan *websocket*. Proses menampilkan data dilakukan secara realtime pada antarmuka web. Hasil pengujian pada ketiga sensor node untuk mendeteksi tetesan menghasilkan nilai yang bervariasi yaitu 96%, 96%, dan 94.6%. Sedangkan untuk pengujian *delay*, diperoleh rata-rata *delay* yang terjadi adalah 454.6 ms.

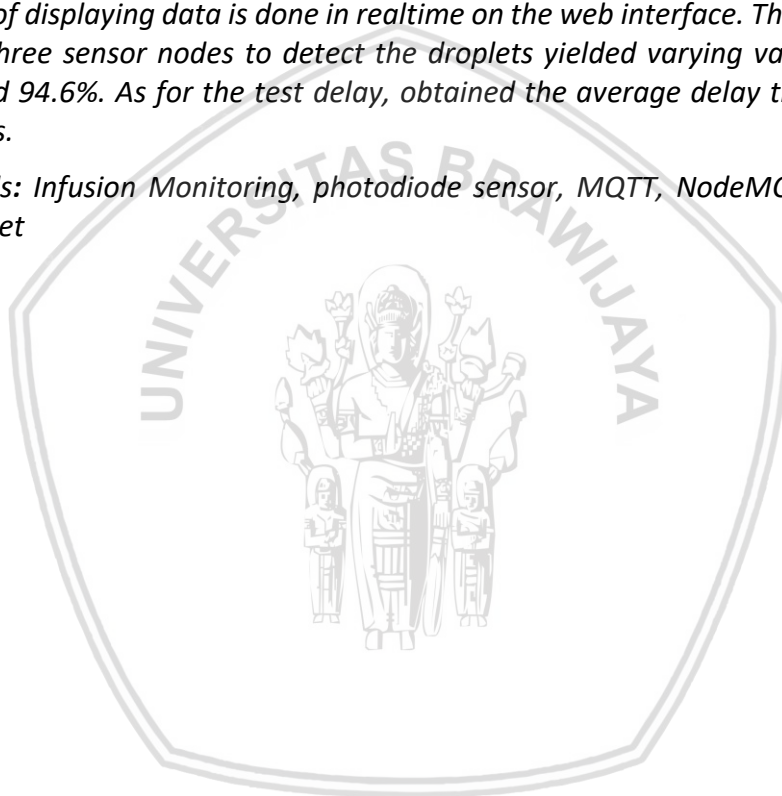
Kata Kunci: Monitoring infus, sensor photodiode, MQTT, NodeMCU, *realtime*, *websocket*



## ABSTRACT

Amount of patient and medical officers leads a new problems. One of them is about monitoring patient infusion fluid. Based on the case, a system is needed to monitor the patient's Infusion fluid when it is low, so that the officer is not late in replacing the intravenous fluids to the patient. To help overcome this problem, a system designed to monitor the patient's infusion centrally using MQTT (Message Queuing Telemetry Transport) delivery. The droplet reading is done by using a photodiode sensor by placing the sensor on the infusion chamber. Data processing on sensor node using NodeMCU and on server using PC / Laptop by utilizing websocket. The process of displaying data is done in realtime on the web interface. The test results on the three sensor nodes to detect the droplets yielded varying values of 96%, 96%, and 94.6%. As for the test delay, obtained the average delay that occurs is 454.6 ms.

**Keywords:** Infusion Monitoring, photodiode sensor, MQTT, NodeMCU, realtime, websocket





## DAFTAR ISI

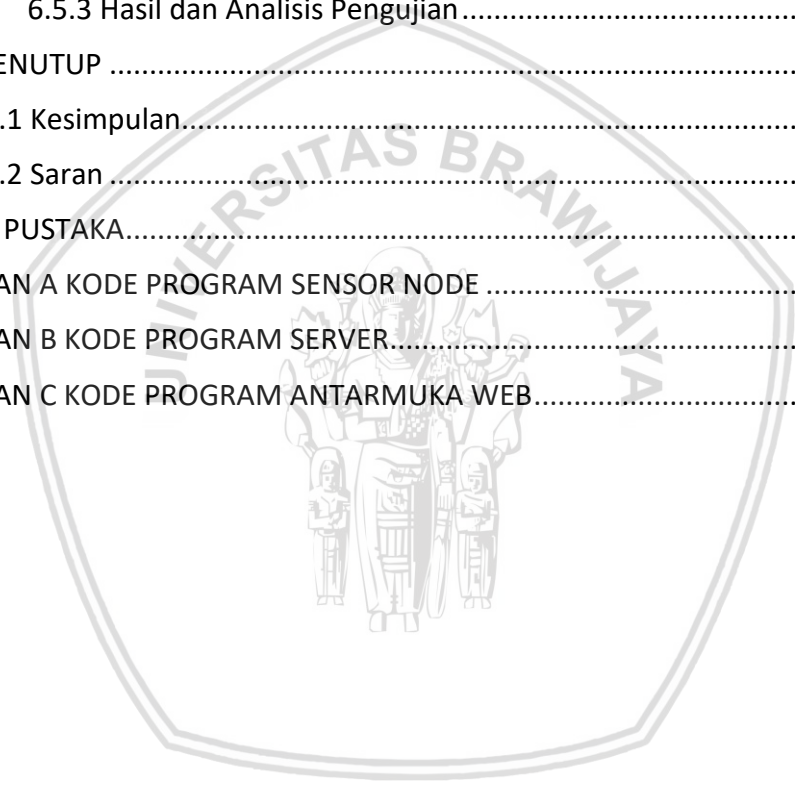
PENGESAHAN .....	ii
DAFTAR ISI .....	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah .....	2
1.3 Tujuan .....	2
1.4 Manfaat.....	2
1.5 Batasan masalah .....	3
1.6 Sistematika pembahasan .....	3
BAB 2 LANDASAN KEPUSTAKAAN .....	5
2.1 Kajian Pustaka .....	5
2.2 Dasar Teori.....	7
2.2.1 Infus.....	7
2.2.2 Sensor Photodiode.....	8
2.2.3 NodeMCU V3.....	10
2.2.4 Arduino IDE .....	11
2.2.5 Protokol MQTT ( <i>Message Queuing Telemetry Transport</i> ).....	12
2.2.6 <i>WebSocket</i> .....	13
BAB 3 METODOLOGI .....	15
3.1 Metode Penelitian .....	15
3.1.1 Studi Literatur .....	16
3.1.2 Analisis Kebutuhan Sistem .....	16
3.1.3 Kebutuhan <i>Hardware</i> .....	18
3.1.4 Kebutuhan <i>Software</i> .....	18
3.2 Perancangan Sistem.....	18
3.2.1 Perancangan <i>Hardware</i> .....	18
3.2.2 Perancangan <i>Software</i> .....	19
3.3 Implementasi .....	19



3.4 Pengujian Sistem.....	20
3.5 Analisis dan Pembahasan .....	20
3.6 Penarikan Kesimpulan .....	20
3.7 Pembuatan Laporan Skripsi .....	21
<b>BAB 4 REKAYASA KEBUTUHAN.....</b>	<b>22</b>
4.1 Gambaran Umum Sistem.....	22
4.2 Kebutuhan Sistem.....	22
4.2.1 Kebutuhan Fungsional.....	22
4.2.2 Kebutuhan <i>Hardware</i> .....	23
4.2.3 Kebutuhan <i>Software</i> .....	23
4.3 Batasan Desain Sistem.....	25
<b>BAB 5 PERANCANGAN DAN IMPLEMENTASI .....</b>	<b>26</b>
5.1 Perancangan Sistem.....	26
5.1.1 Perancangan <i>Prototype</i> Pada Node .....	26
5.1.2 Perancangan Perangkat Keras .....	27
5.1.3 Perancangan Perangkat Lunak.....	28
5.2 Implementasi Sistem .....	31
5.2.1 Implementasi <i>Prototype</i> Node Sensor .....	32
5.2.2 Implementasi Perangkat Keras .....	32
5.2.3 Implementasi Perangkat Lunak.....	33
<b>BAB 6 PENGUJIAN .....</b>	<b>39</b>
6.1 Pengujian Sensor Photodiode.....	39
6.1.1 Tujuan Pengujian.....	39
6.1.2 Prosedur Pengujian .....	39
6.1.3 Hasil dan Analisis Pengujian .....	40
6.2 Pengujian Pengiriman Data Sensor.....	42
6.2.1 Tujuan Pengujian.....	42
6.2.2 Prosedur Pengujian .....	42
6.2.3 Hasil dan Analisis Pengujian .....	42
6.3 Pengujian Delay Pengiriman Data Sensor.....	43
6.3.1 Tujuan Pengujian.....	43
6.3.2 Prosedur Pengujian .....	43

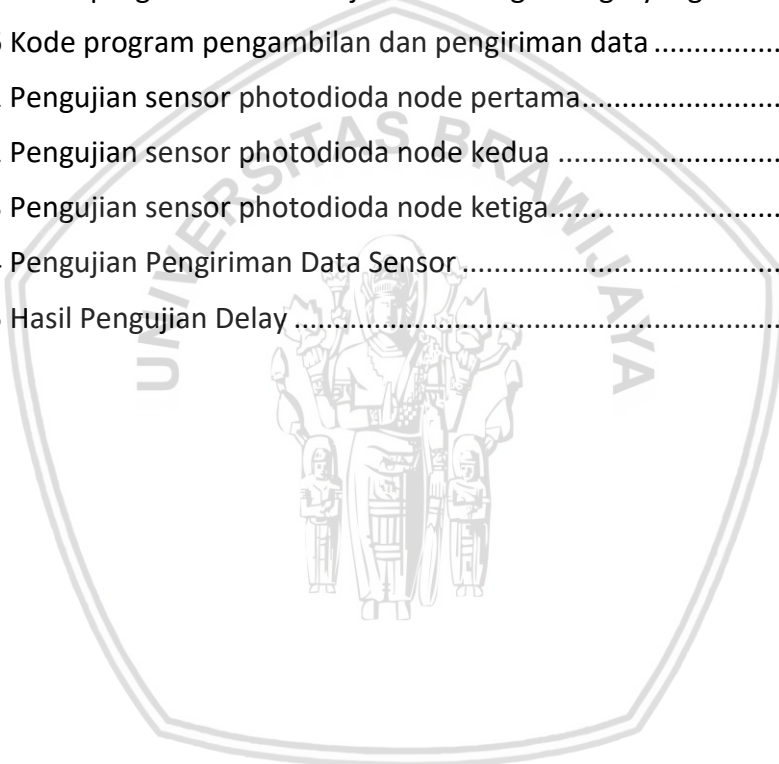


6.3.3 Hasil dan Analisis Pengujian .....	43
6.4 Pengujian Tampilan Pada Web Interface .....	44
6.4.1 Tujuan Pengujian.....	44
6.4.2 Prosedur Pengujian .....	44
6.4.3 Hasil dan Analisis Pengujian.....	45
6.5 Pengujian Sistem Peringatan Pada Web Interface .....	46
6.5.1 Tujuan Pengujian.....	46
6.5.2 Prosedur Pengujian .....	46
6.5.3 Hasil dan Analisis Pengujian.....	46
BAB 7 PENUTUP .....	47
7.1 Kesimpulan.....	47
7.2 Saran .....	47
DAFTAR PUSTAKA.....	49
LAMPIRAN A KODE PROGRAM SENSOR NODE .....	50
LAMPIRAN B KODE PROGRAM SERVER.....	53
LAMPIRAN C KODE PROGRAM ANTAR MUKA WEB.....	54



## DAFTAR TABEL

Tabel 2.1 Perbandingan kajian pustaka dengan penelitian yang diusulkan .....	6
Tabel 2.2 Spesifikasi NodeMCU V3 .....	10
Tabel 5.1 Pin yang digunakan pada NodeMCU .....	27
Tabel 5.2 Kode program untuk <i>import library</i> dan inialisasi variabel .....	34
Tabel 5.3 Kode program menghubungkan sensor node ke <i>access point</i> .....	34
Tabel 5.4 Kode program fungsi untuk menghubungkan sensor node ke <i>broker</i> .	35
Tabel 5. 5 Kode program untuk menjalankan fungsi-fungsi yang ada.....	36
Tabel 5.6 Kode program pengambilan dan pengiriman data .....	36
Tabel 6.1 Pengujian sensor photodiode node pertama.....	40
Tabel 6.2 Pengujian sensor photodiode node kedua .....	40
Tabel 6.3 Pengujian sensor photodiode node ketiga.....	41
Tabel 6.4 Pengujian Pengiriman Data Sensor.....	42
Tabel 6.5 Hasil Pengujian Delay .....	43



## DAFTAR GAMBAR

Gambar 2.1 Botol cairan infus 100 mL.....	7
Gambar 2.2 Infus set.....	8
Gambar 2.3 Photodiode.....	9
Gambar 2.4 Rangkaian sensor photodiode.....	9
Gambar 2.5 Pin pada NodeMCU V3.....	11
Gambar 2.6 Antarmuka Arduino IDE.....	12
Gambar 2.7 Ruang lingkup komunikasi pada MQTT.....	13
Gambar 3.1 Diagram Alir Penelitian.....	15
Gambar 3.2 Rekayasa Kebutuhan Sistem.....	17
Gambar 3.3 Diagram Blok Sensor Node.....	18
Gambar 3.4 Penerapan Sistem Pada Lingkungan.....	19
Gambar 5.1 Perancangan Prototype Sensor Node.....	26
Gambar 5.2 Skematik Sensor Node.....	27
Gambar 5.3 Flowchart untuk menghubungkan <i>hardware</i> dengan <i>access point</i> dan <i>broker</i> .....	28
Gambar 5.4 Diagram alir perancangan pengambilan dan pengiriman data.....	29
Gambar 5.5 Diagram blok perancangan MQTT.....	30
Gambar 5.6 Perancangan Web Interface.....	30
Gambar 5.7 Implementasi Prototype Sensor Node.....	32
Gambar 5.8 Sensor node pada <i>chamber</i> infus.....	32
Gambar 5.9 Implementasi Perangkat Keras.....	33
Gambar 5.10 Tabel untuk menampilkan data monitoring.....	38
Gambar 5.11 Diagram silinder sebagai indikator ketersediaan infus.....	38
Gambar 5.12 Tampilan web secara keseluruhan.....	38
Gambar 6.1 Hasil Pengujian Delay.....	44
Gambar 6.2 Hasil Pengujian Web Interface.....	45
Gambar 6.3 Pengujian Sistem Peringatan.....	46

## BAB 1 PENDAHULUAN

Pada bab ini membahas latar belakang penelitian, tujuan penelitian, manfaat penelitian dan batasan masalah yang digunakan dalam penelitian.

### 1.1 Latar belakang

Pada saat ini perkembangan ilmu kedokteran dan ilmu teknologi semakin canggih. Kondisi tersebut perlu diimbangi dengan adanya peningkatan pelayanan yang diberikan petugas medis kepada pasien rawat inap. Akan tetapi, pada masa sekarang, pelayanan yang diberikan masih tergolong cukup rendah. Sebagai contoh salah satu kasus yang terjadi di Indonesia terjadi di RSUD Depati Hamzah Pangkalpinang Bangka Belitung. Akibat petugas lalai mengganti infus yang sudah habis, seorang bayi berusia 4 hari dinyatakan meninggal dunia. (Fathurrakhman, 2009). Pemberian infus kepada pasien merupakan hal yang sangat penting didalam dunia medis. Alat infus adalah salah satu peralatan medis yang paling banyak digunakan. Dalam dunia kedokteran dan keperawatan, sebanyak 90% pasien di rumah sakit menerima berbagai pengobatan melalui infus. Pada proses pemberian cairan infus kepada pasien memerlukan keakuratan dengan tujuan tidak terjadi hal-hal yang tidak diinginkan kedepannya.

Beberapa klinik medis maupun Rumah Sakit, pemantauan infus masih dilakukan secara manual oleh petugas medis dengan cara mengecek setiap waktu pada ruang pasien atau melalui informasi keluarga pasien yang memberitahu petugas medis bahwa infus akan hampir habis. Hal tersebut tentu akan tidak efektif jika jumlah pasien rawat inap berjumlah banyak dengan jumlah tenaga medis yang terbatas.

Dari masalah di atas, pemantauan cairan infus secara terpusat sangat penting untuk diperhatikan karena dapat mempengaruhi kondisi pasien. Oleh karena itu, diperlukan sistem untuk monitoring infus pasien secara terpusat dengan tujuan mempermudah pekerjaan tenaga medis dalam memantau atau monitoring cairan infus pasien di saat cairan infus tersebut hampir habis kemudian memberikan pemberitahuan kepada petugas atau perawat rumah sakit.

Pada penelitian sebelumnya, telah diteliti sistem pemantauan infus secara terpusat dengan menggunakan sistem kabel menggunakan koneksi RS458 (Syahrul, 2010) . Dari penelitian tersebut, kelemahan yang ditemukan adalah masih menggunakan kabel sehingga untuk instalasi dan fleksibilitas sistem masih kurang. Selain itu, biaya untuk instalasi kabel juga dapat menjadi masalah baru yaitu mahalnya biaya instalasi kabel yang menjangkau seluruh ruangan pada suatu Rumah Sakit.

Pada penelitian diatas, untuk mendeteksi adanya tetesan adalah menggunakan sensor photodiode. Sensor ini memiliki fungsi spesifik salah satunya adalah mendeteksi adanya cahaya. Dari penelitan tersebut diperoleh sensor ini memiliki tingkat ketepatan pembacaan yang tinggi. Dengan demikian, pada

penelitian selanjutnya menggunakan sensor photodiode sebagai media untuk mendeteksi tetesan pada infus.

Protokol MQTT (*Message Queuing Telemetry Transport*) merupakan salah satu protokol yang sangat sederhana dan ringan. Dalam hal ini, arsitektur yang terdapat pada protokol tersebut adalah *publish/subscribe* yang dirancang secara terbuka. Untuk mentransmisikan data, protokol ini menggunakan jaringan wireless. Hanya dengan menggunakan server tunggal, Protokol MQTT dapat menangani ribuan *client* jarak jauh. Protokol MQTT meminimalkan *bandwidth* jaringan dan kebutuhan sumber daya perangkat ketika mencoba untuk menjamin kendalan dan pengiriman (Budioko, 2016). Fokus utama dari protokol MQTT adalah untuk meminimalisir jumlah bytes pada suatu jalur serta penggunaan daya yang rendah (MQTT.org, 2015).

Dari latar belakang diatas, peneliti bermaksud melakukan penelitian yang berjudul "Implementasi Protokol MQTT (*Message Queuing Telemetry Transport*) Untuk Monitoring Infus Pasien Secara Terpusat". Harapan dari penelitian ini adalah dapat membantu dan mempermudah tenaga medis rumah sakit untuk memonitoring infus pasien secara terpusat dengan harga yang terjangkau.

## 1.2 Rumusan masalah

Adapun rumusan masalah pada penelitian ini antara lain sebagai berikut:

1. Bagaimana penerapan sensor photodiode dalam membaca tetesan infus yang terdapat di ruangan pasien?
2. Bagaimana penerapan protokol MQTT dalam melakukan proses pengiriman data dari *node* ke *server* secara *realtime*?
3. Bagaimana menampilkan data yang dihasilkan oleh sensor photodiode dalam bentuk tabel dan diagram?

## 1.3 Tujuan

Adapun tujuan penelitian ini adalah sebagai berikut:

1. Menerapkan LED Inframerah dan photodiode untuk membaca tetesan infus yang terdapat di ruangan pasien.
2. Menerapkan protokol MQTT dalam melakukan proses pengiriman data dari *node* ke *server*.
3. Menampilkan data yang dihasilkan oleh LED Inframerah dan photodiode dalam bentuk *chart*.
4. Mengetahui performa sistem berdasarkan pengujian yang telah dilakukan.

## 1.4 Manfaat

Berikut ini adalah beberapa manfaat dari penelitian ini:

1. Mempermudah pekerjaan perawat Rumah Sakit dalam memonitoring cairan infus pasien.
2. Mengurangi resiko dalam keterlambatan penggantian cairan infus pada pasien.
3. Meningkatkan kualitas pelayanan Rumah Sakit.

### 1.5 Batasan masalah

Dalam menyusun penelitian ini, untuk menghindari pembahasan yang melebar peneliti menggunakan beberapa batasan masalah, antara lain:

1. Pada sistem digunakan 3 *node* sensor yang digunakan sebagai *client* dan satu buah PC yang telah terinstal MQTT *Broker* yang digunakan sebagai *server*.
2. Proses pengiriman data dari *node* ke *server* menggunakan protokol MQTT.
3. Sistem ini hanya untuk memantau ketersediaan infus pada ruang pasien melalui ruang petugas medis.
4. Jenis selang infus yang digunakan adalah selang dengan tetesan makro.
5. Cairan infus menggunakan cairan infus bening dengan volume 100 mL.
6. Lingkup sistem adalah jaringan lokal.

### 1.6 Sistematika pembahasan

Sistematika penulisan penelitian bertujuan untuk memberikan gambaran dan uraian dari penyusunan tugas akhir secara garis besar dengan tujuan supaya pembaca dapat memahami alur dari pembahasan penelitian.

#### **BAB I Pendahuluan**

Pada pendahuluan menjelaskan tentang latar belakang dari penelitian, rumusan masalah, batasan masalah, tujuan, manfaat dari penelitian, batasan masalah, dan sistematika pembahasan.

#### **BAB II Landasan Kepustakaan**

Pada landasan kepastakaan menguraikan tentang kajian pustaka dan dasar teori yang menjadi acuan atau dasar dari penelitian yang dilakukan.

#### **BAB III Metodologi Penelitian**

Pada bagian metodologi penelitian membahas tentang metode yang digunakan di dalam penelitian ini.

#### **BAB IV Rekayasa Kebutuhan**

Pada bagian rekayasa kebutuhan menjelaskan tentang berbagai persyaratan dan kebutuhan yang terdapat dalam sistem, seperti kebutuhan fungsional dan kebutuhan non fungsional.



## **BAB V Perancangan dan Implementasi**

Pada bagian perancangan dan implementasi menguraikan mengenai perancangan yang telah dibuat sebelumnya dan pengimplementasian terhadap rancangan tersebut.

## **BAB VI Pengujian dan Analisis**

Pada bagian pengujian dan analisis membahas tentang pengujian terhadap sistem dan melakukan analisa terhadap hasil pengujian yang telah dilakukan.

## **BAB VII Penutup**

Pada bagian penutup menguraikan kesimpulan dari hasil penelitian yang dilakukan, serta terdapat saran yang bertujuan untuk melakukan pengembangan sistem kedepannya.



## BAB 2 LANDASAN KEPUSTAKAAN

### 2.1 Kajian Pustaka

Kajian pustaka membahas tentang penelitian-penelitian yang sudah ada sebelumnya dan berhubungan dengan penelitian yang dilakukan saat ini.

Penelitian yang pertama merupakan penelitian dari Syahrul Hidayat yang berjudul “Sistem Pemantauan Infus Pasien Terpusat”, dimana pada penelitian tersebut merealisasikan suatu sistem pemantauan infus pasien secara terpusat. Sistem menggunakan 3 modul utama, yaitu modul sensor, modul *interface* dan modul mikrokontroler. Untuk mendeteksi adanya tetesan, menggunakan modul sensor photodiode yang terdiri dari 2 komponen utama yaitu photodiode dan led.. Modul antarmuka menggunakan komunikasi serial RS485 yang berguna untuk menghubungkan antara sensor-sensor optoelektronik dengan komputer dalam jarak yang relatif jauh. Modul mikrokontroler menggunakan AT89C51 keluaran Atmel yang merupakan turunan dari MC-51 (Syahrul, 2010).

Penelitian yang kedua merupakan penelitian dari Decy Nataliana yang berjudul “Alat Monitoring Infus Set pada Pasien Rawat Inap Berbasis Mikrokontroler ATmega 8535”, dimana pada penelitian tersebut membahas mengenai pendeteksian tetesan infus menggunakan sensor photodiode kemudian output dari sistem adalah memberikan keluaran berupa berapa jumlah tetesan permenitnya yang ditampilkan pada LCD dan memberikan peringatan apabila infus hampir habis dengan memberikan output suara menggunakan *buzzer* (Natalina, 2016).

Penelitian yang ketiga atau yang terakhir merupakan penelitian dari Totok Budioko yang berjudul “Sistem Monitoring Suhu Jarak Jauh Berbasis *Internet of Things* Menggunakan Protokol MQTT”, dimana pada penelitian tersebut protokol MQTT digunakan untuk sistem monitoring suhu jarak jauh. Implementasi sistem menggunakan sensor suhu LM35, Arduino UNO, dan modul ESP8266 (Budioko, 2016).

Berdasarkan beberapa penelitian diatas, maka peneliti tertarik untuk membuat sebuah penelitian yang berjudul “IMPLEMENTASI PROTOKOL MQTT (*MESSAGE QUEUING TELEMETRY TRANSPORT*) UNTUK MONITORING INFUS PASIEN SECARA TERPUSAT”, dimana sistem ini berguna untuk monitoring infus pasien secara terpusat dengan tujuan supaya mempermudah pekerjaan tenaga medis dalam memantau atau memonitoring cairan infus pasien di saat cairan infus tersebut hampir habis kemudian memberikan pemberitahuan kepada petugas atau perawat rumah sakit. Adapun pebandingan kajian pustaka dengan penelitian yang diusulkan terdapat pada Tabel 2.1 berikut.

Tabel 2.1 Perbandingan kajian pustaka dengan penelitian yang diusulkan

No.	Nama Penulis [Tahun], Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1.	(Syahrul, et. al., 2009), Sistem Pemantauan Infus Pasien Terpusat	Pada penelitian tersebut, untuk mendeteksi tetesan infus menggunakan sensor photodiode dan monitoring dilakukan secara terpusat.	Penelitian sebelumnya pengiriman data sensor menggunakan kabel RS232 dan tampilan menggunakan aplikasi <i>desktop</i> .	Pada penelitian selanjutnya, metode pengiriman dilakukan secara nirkabel sehingga akan meminimalisir biaya instalasi dengan mengurangi beban biaya pada kebutuhan kabel. Selain itu, penelitian selanjutnya menggunakan web sebagai sarana untuk menampilkan data yang diperoleh dari sensor.
2.	(Nataliana, 2016), Alat Monitoring Infus Set pada Pasien Rawat Inap Berbasis Mikrokontroler ATmega 8535	Pada penelitian sebelumnya, objek yang diteliti adalah infus pasien. Selain itu, peneliti juga menggunakan sensor photodiode sebagai sensor untuk mendeteksi tetesan.	Pada penelitian sebelumnya, peneliti menggunakan buzzer sebagai indikator peringatan ketika tidak adanya tetesan pada chamber infus.	Pada penelitian selanjutnya, peneliti akan menggunakan sistem peringatan yang diletakkan pada web interface dan memberikan suara peringatan.
3.	(Budioko, 2016), Sistem Monitoring Suhu Jarak Jauh Berbasis <i>Internet of Things</i>	Pada penelitian sebelumnya, peneliti menggunakan	Pada penelitian sebelumnya, objek yang diteliti adalah suhu.	Pada penelitian selanjutnya, peneliti menggunakan

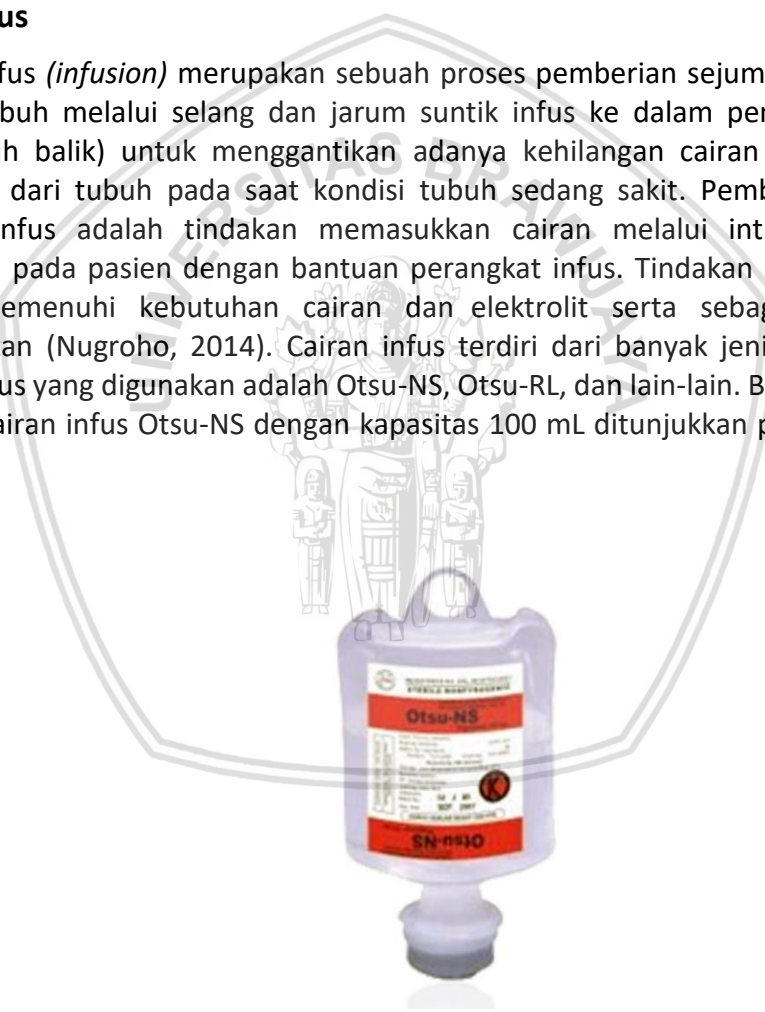
No.	Nama Penulis [Tahun], Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
	Menggunakan Protokol MQTT	metode MQTT sebagai metode pengiriman data sensor.		penelitian berupa infus pasien.

## 2.2 Dasar Teori

Pada sub bab ini akan menjelaskan referensi dasar teori sebagai pengetahuan mengenai komponen dan teknologi yang digunakan.

### 2.2.1 Infus

Infus (*infusion*) merupakan sebuah proses pemberian sejumlah cairan ke dalam tubuh melalui selang dan jarum suntik infus ke dalam pembuluh vena (pembuluh balik) untuk menggantikan adanya kehilangan cairan atau zat-zat makanan dari tubuh pada saat kondisi tubuh sedang sakit. Pemberian cairan melalui infus adalah tindakan memasukkan cairan melalui intravena yang dilakukan pada pasien dengan bantuan perangkat infus. Tindakan ini bertujuan untuk memenuhi kebutuhan cairan dan elektrolit serta sebagai tindakan pengobatan (Nugroho, 2014). Cairan infus terdiri dari banyak jenis. Jenis-jenis cairan infus yang digunakan adalah Otsu-NS, Otsu-RL, dan lain-lain. Berikut adalah contoh cairan infus Otsu-NS dengan kapasitas 100 mL ditunjukkan pada Gambar 2.1.



**Gambar 2.1 Botol cairan infus 100 mL**

Sumber: (medicalogy.com)

Selain cairan infus, kelengkapan infus yang digunakan adalah infus set yang digunakan untuk mengalirkan cairan infus ke tubuh pasien. Terdapat 2 jenis infus set yaitu sebagai berikut:

1. Infus set makro

Infus set makro merupakan infus set yang memiliki nilai faktor tetesan 20 drop/mL.

2. Infus set mikro

Infus set mikro merupakan jenis infus set yang memiliki nilai faktor tetesan 60 drop/mL.

Berikut ini adalah contoh gambar infus set ditunjukkan pada Gambar 2.2



**Gambar 2.2 Infus set**

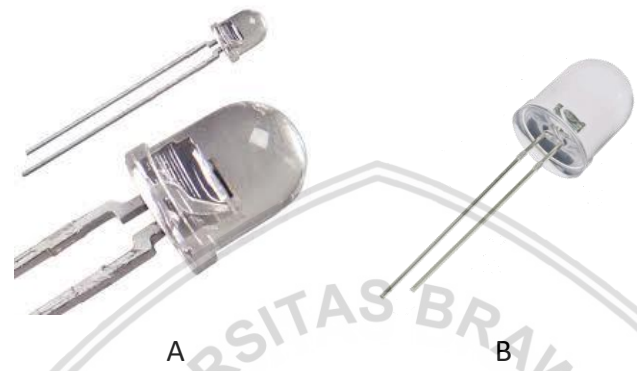
Sumber: (medicalogy.com)

### 2.2.2 Sensor Photodioda

Sensor photodioda adalah sensor yang digunakan untuk mendeteksi adanya suatu benda dengan membaca perubahan intensitas cahaya yang diperoleh. Sensor ini terdiri dari dua komponen utama yaitu pengirim (*transmitter*) dan penerima (*receiver*). Pada sensor ini, terdapat dua cara pemasangan antara *transmitter* dan *receiver*, yaitu diletakkan secara sejajar dan berhadapan. Perbedaan cara pemasangan tersebut disesuaikan dengan kebutuhan sistem.

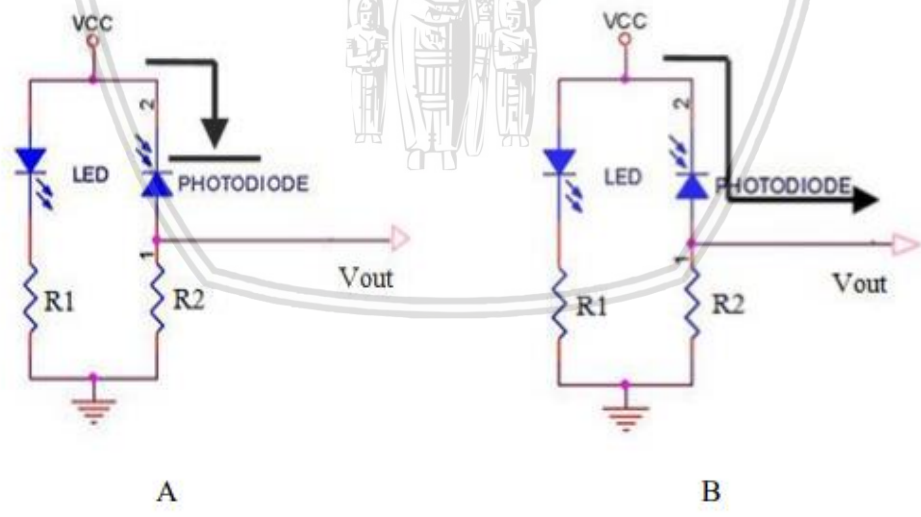
Sensor ini terdiri dari dua komponen utama yaitu photodioda dan LED. Photodioda adalah suatu jenis dioda yang resistansinya berubah-ubah kalau cahaya yang jatuh pada dioda berubah ubah intensitasnya. Dalam keadaan gelap, nilai tahanannya sangat besar hingga praktis tidak ada arus yang mengalir. Semakin kuat cahaya yang jatuh pada dioda maka makin kecil nilai tahanannya, sehingga arus yang mengalir semakin besar. Jika photodioda persambungan positif-negatif (p-n) bertegangan balik disinari, maka arus akan berubah secara linier dengan kenaikan fluks cahaya yang dikenakan pada persambungan tersebut. Photodioda terbuat dari bahan semikonduktor. Photodioda digunakan

sebagai komponen pendeteksi ada tidaknya cahaya maupun dapat digunakan untuk membentuk sebuah alat ukur akurat yang dapat mendeteksi intensitas cahaya dibawah  $1 \text{ pW/cm}^2$  sampai intensitas diatas  $10\text{mW/cm}^2$ . Photodiode mempunyai resistansi yang rendah pada kondisi forward bias, kita dapat memanfaatkan photodiode ini pada kondisi reverse bias dimana resistansi dari photodiode akan turun seiring dengan intensitas cahaya yang masuk (Ramadhan, 2013). Contoh photodiode ditunjukkan pada Gambar 2.3 A dan LED pada Gambar 2.3 B.



**Gambar 2.3 Photodiode**  
 Sumber : (edukasielektronika.com)

**2.2.2.1 Prinsip Kerja Photodiode**



**Gambar 2.4 Rangkaian sensor photodiode**  
 Sumber: (skpang.co.uk)

Berdasarkan Gambar 2.4 A merupakan rangkaian dasar dari sebuah sensor photodiode. Pada kondisi awal LED sebagai transmitter cahaya menyinari photodiode sebagai receiver, sehingga nilai resistansi pada sensor photodiode akan minimum dengan kata lain nilai  $V_{out}$  mendekati logika 0 (*low*). Sedangkan,

pada kondisi kedua pada gambar 2.4 B cahaya pada led terhalang oleh permukaan hitam sehingga photodiode tidak dapat menerima cahaya dari led maka nilai resistansi R1 maksimum, sehingga nilai  $V_{out}$  akan mendekati  $V_{cc}$  yang mempunyai logika 1 (*high*).

### 2.2.3 NodeMCU V3

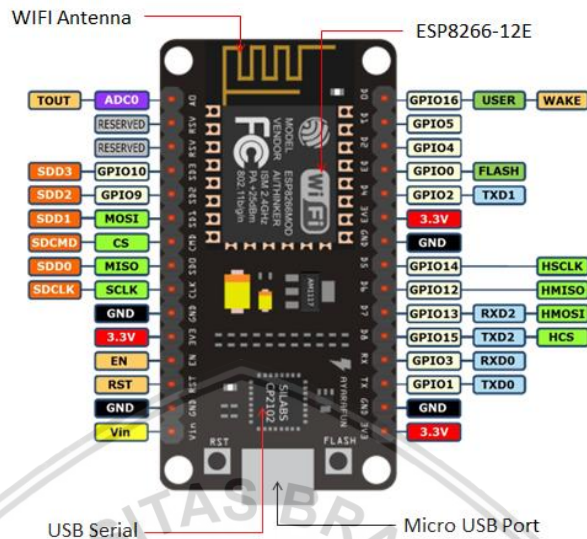
NodeMCU pada dasarnya adalah pengembangan dari ESP8266 dengan *firmware* berbasis Lua. Pada NodeMCU dilengkapi dengan *micro usb port* yang berfungsi untuk media transfer maupun power supply. Pada NodeMCU juga dilengkapi dengan tombol *reset* dan *flash*. Bahasa pemrograman yang dapat digunakan untuk memprogram NodeMCU adalah bahasa Lua yang merupakan package dari modul ESP8266. Bahasa *Lua* memiliki logika dan susunan pemrograman yang sama dengan Bahasa C, hanya berbeda pada *syntax*-nya. Apabila menggunakan bahasa *Lua*, maka memerlukan tool *Lua loader* maupun *Lua uploader*. NodeMCU juga didukung dengan software Arduino IDE untuk memprogramnya. Dengan memanfaatkan board yang dikembangkan oleh komunitas, Arduino IDE dengan mudah dapat terintegrasi dengan NodeMCU.

Tabel 2.2 Spesifikasi NodeMCU V3

<b>Mikrokontroler</b>	ESP8266
<b>Ukuran Board</b>	57 mm x 30 mm
<b>Tegangan Masukan</b>	3.3 – 5 Volt
<b>Pin GPIO</b>	13 Pin
<b>Kanal PWM</b>	1
<b>10 bit ADC Pin</b>	1 Pin
<b>Flash Memory</b>	4 MB
<b>Clock Speed</b>	40/26/24 MHz
<b>Wifi</b>	IEEE 802.11 b/g/n
<b>Frekuensi</b>	2.4 GHz – 22.5 Ghz
<b>USB Port</b>	<i>Micro</i> USB
<b>Card Reader</b>	Tidak ada
<b>USB to Serial Converter</b>	CP2102

NodeMCU merupakan komponen pengembang yang bersifat *open-source* dimana pada komponen tersebut terdiri dari dua komponen pembentuk utama yaitu modul *wifi* ESP8266-12E dan modul FTDI (*Future Technology Device International*) sehingga memudahkan pengguna dalam mengembangkan suatu sistem. NodeMCU bekerja pada tegangan 3,3 – 5 V dimana tegangan tersebut didapatkan dari konversi pada regulator yang terdapat pada *board*. Modul *wifi*

bekerja pada frekwensi 2,4 GHz – 2,5 GHz (Albert, 2015). Spesifikasi dan pin pada NodeMCU V3 ditunjukkan pada Tabel 2.2 dan Gambar 2.5.



**Gambar 2.5 Pin pada NodeMCU V3**

Sumber: (sparkfun.com)

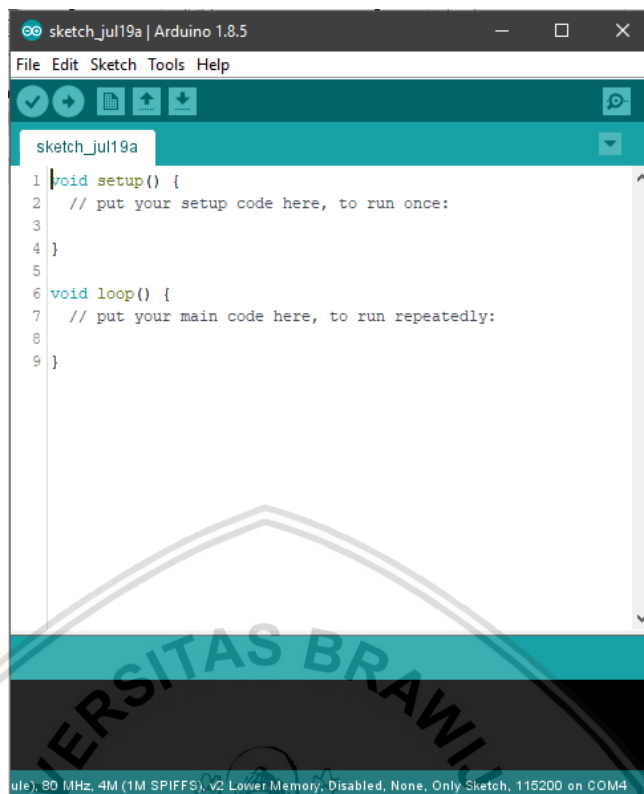
Gambar 2.5 merupakan antarmuka NodeMCU dan pin-pin yang terdapat pada NodeMCU. Selain pin-pin yang digunakan sebagai komunikasi dengan perangkat lain, pada NodeMCU terdapat 2 tombol yaitu tombol flash dan reset. Tombol flash digunakan untuk flash firmware dari NodeMCU, sedangkan tombol reset digunakan untuk menjalankan ulang program ketika NodeMCU dijalankan.

### 2.2.4 Arduino IDE

IDE merupakan singkatan dari *Integrated Development Environment*. IDE merupakan sebuah aplikasi yang membantu programmer dalam membuat sebuah aplikasi menggunakan bahasa pemrograman. Aplikasi IDE adalah tempat programmer menuliskan baris-baris kode dan menjalankan program. Untuk menuliskan kode program pada Arduino dapat menggunakan Arduino IDE.

Arduino IDE merupakan *tools* yang digunakan untuk memprogram board Arduino. Beberapa fitur yang terdapat pada Arduino IDE antara lain *text editor*, *compiler*, *uploader* dan *serial monitor*. Untuk fitur *text editor* merupakan fitur yang digunakan untuk menuliskan kode program. *Compiler* merupakan fitur yang digunakan untuk memastikan apakah kode program sudah sesuai dengan standar bahasa C Arduino. *Uploader* adalah fitur yang digunakan untuk upload program dari Arduino IDE menuju ke *board* Arduino. Serial monitor adalah fitur yang digunakan untuk menampilkan *output* kode program yang dijalankan dan untuk memberi masukan serial ke *board* Arduino. Gambar 2.6 merupakan tampilan antarmuka awal Arduino IDE saat dibuka. Pada Arduino IDE, terdapat 2 fungsi utama yaitu fungsi *setup()* dan fungsi *loop()*.





**Gambar 2.6 Antarmuka Arduino IDE**

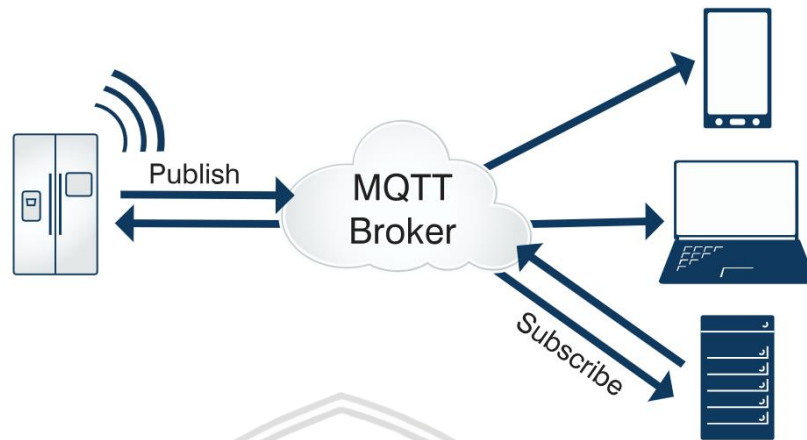
Fungsi *setup()* merupakan fungsi yang digunakan untuk konfigurasi *board* Arduino. Misalnya, untuk konfigurasi pin yang digunakan pada *board* Arduino, konfigurasi *baudrate*, dan lain sebagainya. Fungsi ini akan dijalankan pertama kali ketika Arduino sedang menjalankan program. Fungsi *loop()* merupakan fungsi yang akan dijalankan secara berulang-ulang selama program dijalankan. Kedua fungsi diatas memiliki tipe data *void*, yaitu tidak mengembalikan *return value* (James, 2015).

Selain digunakan untuk memprogram Arduino, Arduino IDE mendukung untuk digunakan untuk memprogram *board* NodeMCU. Untuk dapat mengintegrasikan Arduino IDE dengan *board* NodeMCU, menggunakan konfigurasi board external yang dapat dipasang melalui menu *Board Manager* pada Arduino IDE.

### **2.2.5 Protokol MQTT (*Message Queuing Telemetry Transport*)**

Message Queue Telemetry Transport (MQTT) merupakan sebuah protokol komunikasi data machine to machine (M2M) yang berada pada layer aplikasi, MQTT bersifat *lightweight* message artinya MQTT berkomunikasi dengan mengirimkan data pesan yang memiliki *header* berukuran kecil yaitu hanya sebesar 2bytes untuk setiap jenis data, sehingga dapat bekerja di dalam lingkungan yang terbatas sumber dayanya seperti kecilnya *bandwidth* dan terbatasnya sumber daya listrik, selain itu protokol ini juga menjamin terkirimnya semua pesan walaupun koneksi terputus sementara, protokol MQTT

menggunakan metode publish/subscribe untuk metode komunikasinya. Secara garis besar, komunikasi pada MQTT ditunjukkan pada Gambar 2.7.



**Gambar 2. 7 Ruang lingkup komunikasi pada MQTT**

Sumber: (Appcelerator.com)

#### **2.2.5.1 Publish dan Subscribe**

Publish/subscribe merupakan sebuah pola pertukaran pesan di dalam komunikasi jaringan dimana pengirim data disebut sebagai *publisher* dan penerima data disebut dengan sebagai *subscriber*, metode publish/subscribe memiliki beberapa kelebihan salah satunya yaitu loose coupling atau decouple dimana berarti antara *publisher* dan *subscriber* tidak saling mengetahui keberadaannya, terdapat 3 buah *decoupling* yaitu *time decoupling*, *space decoupling* dan *synchronization decoupling*, *time decoupling* adalah sebuah kondisi dimana *publisher* dan *subscriber* tidak harus saling aktif pada waktu yang sama, *space decoupling* adalah dimana *publisher* dan *subscriber* aktif di waktu yang sama akan tetapi antara *publisher* dan *subscriber* tidak saling mengetahui keberadaan dan identitas satu sama lain, dan yang terakhir adalah *synchronization decoupling* kondisi dimana pengaturan *event* baik itu penerimaan atau pengiriman pesan di sebuah node hingga tidak saling mengganggu satu sama lain (Rochman, 2017).

#### **2.2.5.2 Broker**

Broker merupakan komponen pada MQTT yang bertindak menangani adanya *publish* dan *subscribe* data. Broker berfungsi sebagai penghubung komunikasi antara publisher dan subscriber. Publisher mengirimkan data kepada broker dengan suatu topik, sedangkan subscriber meminta data kepada broker dengan suatu topik. Beberapa broker yang ada pada saat ini adalah Mosquitto, Hivemq, mosca, EMQ, dan lain sebagainya.

#### **2.2.6 WebSocket**

*WebSocket* adalah standar untuk komunikasi *realtime* pada Web dan aplikasi *mobile*. *WebSocket* dirancang untuk diterapkan di *browser* web dan *server* web, tetapi dapat digunakan oleh aplikasi *client* atau *server*. *WebSocket* adalah

protokol yang menyediakan saluran komunikasi *full-duplex* melalui koneksi TCP tunggal. Protokol *WebSocket* sudah di standarisasi oleh IETF sebagai RFC 6455 pada tahun 2011, dan API *WebSocket* di Web IDL sedang distandarisasi oleh W3C (Hardianto, 2015). Berikut adalah manfaat dari penggunaan *WebSocket*, antara lain:

1. *Websocket* memungkinkan server untuk mendorong data kepada klien yang sedang terhubung.
2. Mengurangi *traffic* atau lalu lintas jaringan yang tidak perlu dan *latency* menggunakan *full duplex* melalui koneksi tunggal (bukan dua).
3. *Streaming* melalui *proxy* dan *firewall*, mendukung komunikasi simultan hulu dan hilir.
4. Mendukung dengan pre-*WebSocket* dunia dengan cara beralih dari koneksi HTTP ke *WebSockets*.

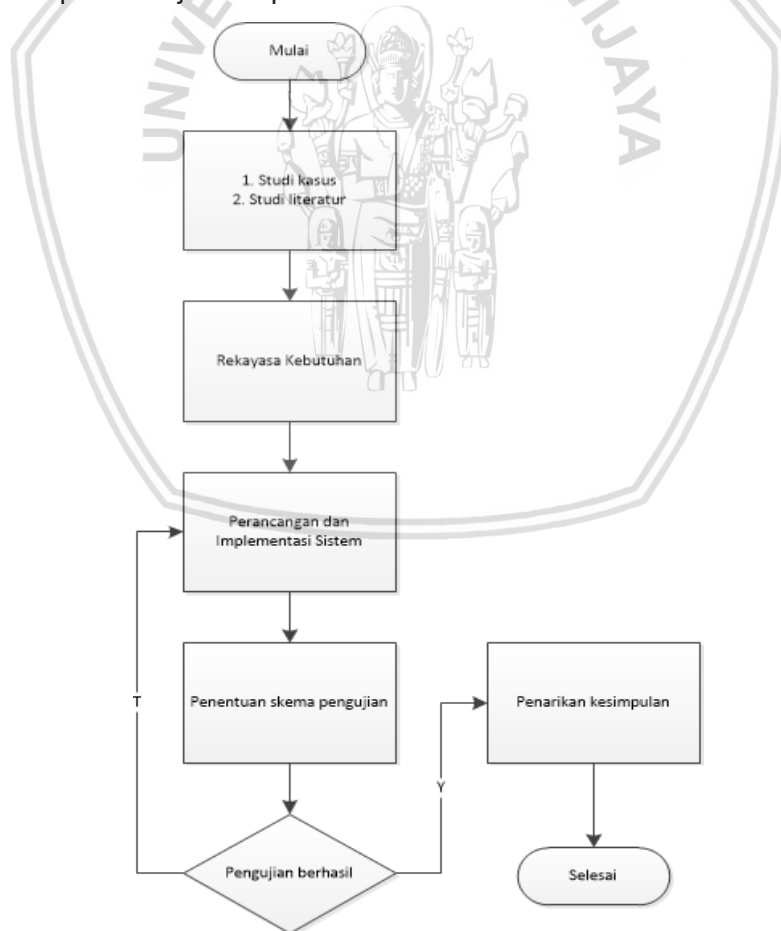


## BAB 3 METODOLOGI

Bab ini menjelaskan metode yang digunakan dalam melakukan penelitian tentang “Implementasi Protokol MQTT (*Message Queuing Telemetry Transport*) untuk Monitoring Infus Pasien dan Mengatur Kecepatan Tetesan Infus Terpusat” .

### 3.1 Metode Penelitian

Terdapat beberapa proses yang saling berkaitan untuk menciptakan suatu perancangan sistem yang terstruktur. Diawali dengan studi literatur terkait dengan tinjauan pustaka dari beberapa penelitian terdahulu dan dasar teori mengenai implementasi dari metode MQTT. Penelitian ini bersifat implementatif, pada awal penelitian dilakukan analisa kebutuhan yang menggambarkan keseluruhan sistem secara garis besar yang dibentuk melalui beberapa subsistem. Pada penelitian ini, proses perancangan dilakukan berdasarkan blok diagram sistem yang selanjutnya diimplementasikan dan dilakukan pengujian sistem berdasarkan analisa kebutuhan yang telah ditetapkan. Jika hasil yang didapatkan sesuai, maka didapatkan beberapa kesimpulan dari penelitian. Diagram alir penelitian dapat ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

### 3.1.1 Studi Literatur

Studi literatur dilakukan untuk mencari penelitian sebelumnya sebagai pendukung penelitian yang akan dilakukan sekaligus untuk menyusun teori yang digunakan dalam penelitian dan perancangan sistem. Teori yang diperlukan untuk merancang sistem yaitu teori protocol MQTT, pemrograman ESP8266, komunikasi *wireless*, pemrograman web dan prinsip kerja sensor photodiode.

1. Protokol MQTT (*Message Queuing Telemetry Transport*)

Pada sub bab ini akan dilakukan kajian terkait dengan penelitian terdahulu yang menggunakan metode MQTT sebagai protokol pengiriman data. Pada bagian ini juga dibahas mengenai bagaimana alur pengiriman data yang terjadi.

2. Modul komunikasi *wireless*

Pada bagian ini akan dibahas bagaimana penggunaan modul yang digunakan untuk komunikasi *wireless* agar dapat mengirimkan data.

3. Pemrograman Web

Pada bagian ini akan dilakukan pencarian literatur untuk mengetahui bagaimana penerapan pemrograman web untuk menampilkan data secara *real-time*.

4. Sensor photodiode

Pada bagian ini akan dilakukan pencarian literatur untuk mengetahui prinsip kerja sensor photodiode sehingga dapat membaca tetesan infus.

### 3.1.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem pada perancangan sistem bertujuan untuk mendefinisikan kebutuhan yang diperlukan pada sistem ini. Perancangan terdiri dari dua bagian yaitu perancangan perangkat keras (*Hardware*) dan perancangan perangkat lunak (*Software*). Kedua perancangan tersebut dianalisa sesuai fungsi dan cara kerjanya sehingga mempermudah dalam implementasi penelitian.

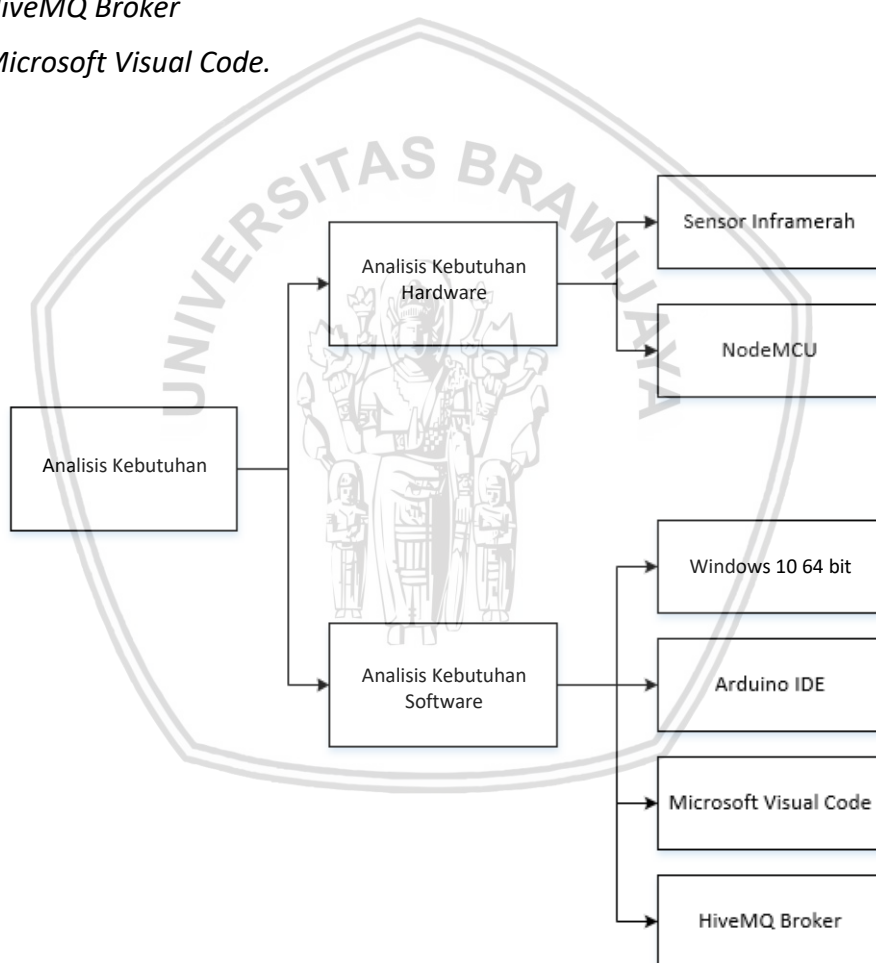
Pada perancangan perangkat keras sistem monitoring infus ini dibutuhkan 3 komponen utama yang digunakan untuk mendeteksi ada atau tidaknya tetesan pada selang infus yaitu sensor photodiode.

Kemudian untuk mengolah data dibutuhkan sebuah hardware yang dapat menerima dan mengolah data sensor. Pada sistem ini digunakan modul NodeMCU untuk mengolah data yang didapatkan dari sensor photodiode. NodeMCU adalah modul *wifi* yang dilengkapi dengan mikrokontroler sehingga selain untuk mengolah data, modul ini digunakan sebagai pengirim data secara *wireless*.

Sementara itu, untuk memprogram NodeMCU adalah menggunakan bahasa pemrograman Arduino dan untuk teks editor menggunakan Arduino IDE.

Berikut ini adalah kelengkapan alat yang diperlukan untuk membentuk sistem dan ditunjukkan pada Gambar 3.2:

1. 3 buah sensor photodioda.
2. 3 buah modul NodeMCU.
3. 3 buah kabel USB mikro.
4. Laptop/ Komputer.
5. Arduino IDE.
6. *HiveMQ Broker*
7. *Microsoft Visual Code*.



**Gambar 3.2** Rekayasa Kebutuhan Sistem

### 3.1.3 Kebutuhan *Hardware*

Berikut ini adalah kebutuhan *hardware* yang digunakan untuk membangun sistem:

1. Komputer/ Laptop.
2. Sensor photodiode.
3. Modul NodeMCU.
4. Kabel USB Mikro.
5. Sumber tegangan

### 3.1.4 Kebutuhan *Software*

Berikut ini adalah kebutuhan *software* yang digunakan untuk mendukung kinerja dari *hardware* yang digunakan sistem:

1. Arduino IDE.
2. HiveMQ Broker.
3. Microsoft Visual Code.
4. Windows 10 Pro 64-bit sebagai sistem operasi.
5. Browser

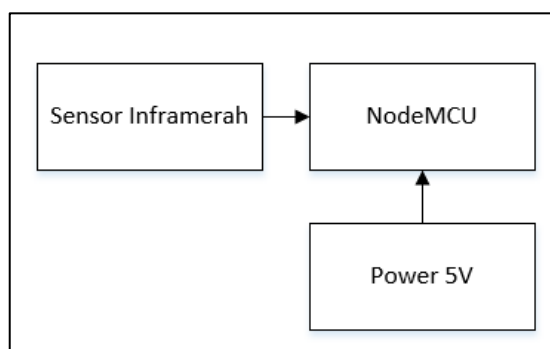
## 3.2 Perancangan Sistem

Pada tahap ini bertujuan untuk memudahkan dalam implementasi agar lebih terstruktur.

### 3.2.1 Perancangan *Hardware*

#### 1. Perancangan Sensor node

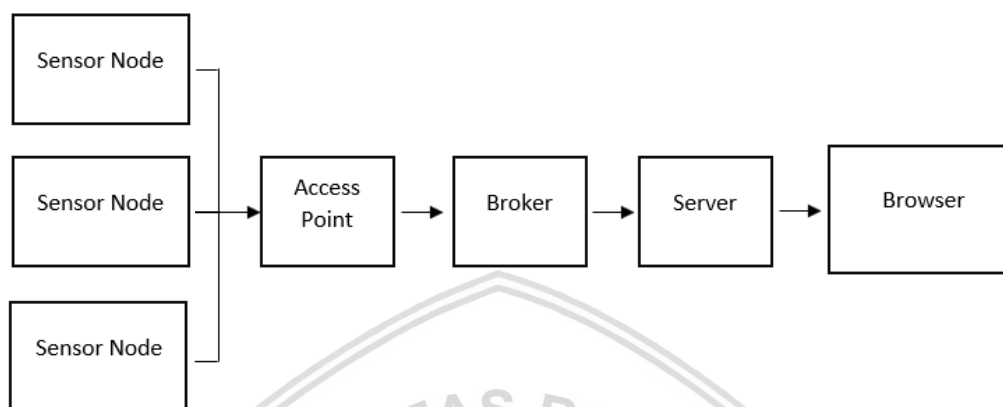
Sensor *node* merupakan sebuah *node* yang terdiri *input*, pemroses dan *output*. *Input* pada *node* adalah sensor-sensor yang digunakan sesuai dengan tujuan. Pemroses yaitu sebuah alat yang digunakan sebagai penerima *input* serta mengolah data. *Output* yaitu hasil *input* yang telah diproses yang siap dikirimkan dari *node* ke *server*.



Gambar 3.3 Diagram Blok Sensor Node

## 2. Perancangan Keseluruhan Sistem

Perancangan hardware akan dilakukan dengan membuat sensor node yang akan diletakkan pada setiap kamar pasien.



**Gambar 3.4 Penerapan Sistem Pada Lingkungan**

Berdasarkan gambar diatas perancangan yang dilakukan meliputi:

1. Pendeteksi tetesan infus pasien pada masing-masing ruangan.
2. Pengiriman data dari sensor node ke server.
3. Perangkat lunak komputer untuk monitoring dan memberikan peringatan infus habis pada pasien.

### 3.2.2 Perancangan *Software*

Perancangan software dilakukan dengan menggunakan bahasa pemrograman web dimana pada *user interface* akan ditampilkan data yang diperoleh dari sensor node dan fitur jika infus hampir habis maka akan menampilkan peringatan.

## 3.3 Implementasi

Implementasi merupakan tahap perancangan sistem hingga mendapatkan hasil dari penelitian yang diharapkan. Berikut ini adalah tahap-tahap implementasi dari sistem :

1. Implementasi pada sensor node dengan modul *wifi* NodeMCU yang dapat terkoneksi dengan jaringan *wifi* lokal, dapat menerima data sensor dan mengirimkan data ke server.
2. Implementasi pada protokol MQTT sebagai protokol pengiriman. Pada penelitian ini digunakan HiveMQ sebagai *broker*.
3. Implementasi server sebagai penyedia data secara *realtime* untuk diproses pada web browser.



4. Implementasi pada antarmuka halaman *browser* untuk menampilkan data yang diterima dari sensor yang disajikan dalam tabel dan diagram secara *realtime*.

Dari tahap-tahap diatas diharapkan mendapatkan hasil sebagai berikut :

1. Data sensor photodiode dapat diterima oleh modul NodeMCU.
2. Modul NodeMCU yang terhubung melalui jaringan *wifi* lokal dapat mem-*publish* data menuju ke *broker* HiveMQ.
3. Server node js melakukan *subscribe* data ke *broker* HiveMQ
4. Browser menampilkan data yang diterima yang disajikan pada tabel dan diagram secara *realtime*.

### 3.4 Pengujian Sistem

Pengujian dilakukan dengan beberapa tahap pengujian antara lain:

1. Pengujian sensor photodiode.
2. Pengujian pengiriman data sensor.
3. Pengujian delay pengiriman data sensor node.
4. Pengujian web *interface*.
5. Pengujian sistem peringatan pada web.

### 3.5 Analisis dan Pembahasan

Analisis dan pembahasan dilakukan setelah implementasi sistem dan pengujian sistem dilakukan. Pada tahap ini bertujuan untuk mendapatkan kesimpulan dari penelitian yang dilakukan dan menjadi parameter penelitian telah berjalan baik atau tidak.

### 3.6 Penarikan Kesimpulan

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi, pengujian dan analisis sistem telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis sistem yang dibuat. Kesimpulan diambil untuk mengetahui respon sistem ketika sedang proses monitoring, respon sistem pada saat infus hampir habis, respon sistem saat diatur kecepatan tetesan infus.

Tahap ini juga berisi saran untuk mengembangkan dan memperbaiki kesalahan pada penelitian "Implementasi Protokol MQTT (*Message Queuing Telemetry Transport*) Untuk Monitoring Infus Pasien Secara Terpusat". Memberikan ide kepada peneliti selanjutnya untuk menambahkan perangkat keras maupun perangkat lunak maupun pengembangan metode agar sistem lebih sempurna.

### 3.7 Pembuatan Laporan Skripsi

Tahap ini merupakan tahap akhir dari penelitian dan bertujuan untuk membuat dokumentasi dari penelitian yang dilakukan. Dokumentasi dibuat untuk mempermudah pengembangan penelitian ini kedepannya.



## BAB 4 REKAYASA KEBUTUHAN

Pada bab ini membahas mengenai kebutuhan sistem yang diperlukan untuk melaksanakan penelitian. Adapun kebutuhan yang diperlukan antara lain kebutuhan fungsional, kebutuhan perangkat keras dan kebutuhan perangkat lunak. Selain itu, pada bab ini membahas mengenai batasan desain sistem.

### 4.1 Gambaran Umum Sistem

Pada sistem monitoring infus pasien dan memberikan peringatan bahwa infus hampir habis ini terdapat *input* dan *output*. Dimana input pada sistem yaitu tetesan cairan pada chamber infus yang dideteksi dengan menggunakan sensor photodiode. Output yang dihasilkan dari sensor ini berupa data analog. output dari sistem ini adalah berupa diagram yang merepresentasikan ketersediaan infus pada ruang pasien dan peringatan kepada petugas apabila cairan infus hampir habis.

Pemrograman Arduino digunakan untuk melakukan pemrograman pada NodeMCU sebagai pemroses data. Pada NodeMCU dilakukan penerimaan data dari sensor yang kemudian diolah kemudian dikirimkan melalui jaringan *wifi* lokal menggunakan metode MQTT. Pada penelitian ini juga digunakan pemrograman web dimana untuk menampilkan data yang diterima dari node dalam bentuk *chart* yang merepresentasikan ketersediaan infus pada ruang pasien. Selain itu, ditampilkan peringatan ketika infus pada ruang pasien hampir habis.

### 4.2 Kebutuhan Sistem

Pada bagian ini dijelaskan penjelasan kebutuhan fungsional, kebutuhan *hardware*, dan kebutuhan *software* yang dianalisa sesuai dengan yang dibutuhkan sistem dengan harapan dapat mempermudah dalam melakukan desain sistem dan implementasi sistem.

#### 4.2.1 Kebutuhan Fungsional

Berikut adalah kebutuhan fungsional yang harus terpenuhi dalam penelitian ini, diantaranya :

1. **Fungsi sensor dapat mendeteksi tetesan infus**

Fungsi ini mengharuskan sistem dapat mendeteksi tetesan infus yang ada pada *chamber* yang akan digunakan sebagai *input* pada sistem. Fungsi ini menjadi prioritas utama karena sensor tidak mendeteksi tetesan infus, maka sistem tidak akan berfungsi.

2. **Fungsi node dapat mengirimkan (*publish*) counter ke *server broker***

Pada fungsi ini, data dari sensor menjadi acuan utama berjalannya fungsi ini karena jika tidak terdeteksi adanya tetesan infus, maka fungsi ini tidak berjalan.

### 3. Fungsi meminta (*subscribe*) data counter dari broker

Pada fungsi ini, sistem dapat meminta data dari broker. Data yang didapatkan adalah berupa data counter yang dikirimkan oleh node.

### 4. Fungsi menampilkan data dalam bentuk table dan *chart*

Pada fungsi ini data hasil monitoring tetesan infus ditampilkan pada browser.

#### 4.2.2 Kebutuhan *Hardware*

Adapun kebutuhan perangkat keras yang digunakan dalam proses implementasi sistem ini antara lain :

##### 1. Komputer/ Laptop

Perangkat ini digunakan sebagai media untuk memprogram NodeMCU, menjadi *server broker*, sebagai *subscriber* dan sebagai media untuk menampilkan data yang disajikan dengan *chart*. Adapun spesifikasi komputer yang digunakan adalah sebagai berikut:

- Model Perangkat : Acer Aspire E1-451G
- Processor : AMD A8
- RAM : DDR3 6GB

##### 2. Sensor photodiode

Sensor photodiode berfungsi untuk mendeteksi tetesan infus pada *chamber*.

##### 3. Modul NodeMCU

Modul NodeMCU merupakan sebuah alat pengembang yang didalamnya terdapat *chip* modul *wifi* ESP8266-12E dan FTDI (*Future Technology Device International*). Pada sistem ini, modul NodeMCU digunakan sebagai pemroses input dan pengirim data hasil pemrosesan secara *wireless*.

##### 4. Kabel USB *Micro*

Kabel USB Mikro berfungsi sebagai media penghubung antara NodeMCU dengan komputer. Kabel ini digunakan untuk mengirim kode program dari komputer ke NodeMCU.

##### 5. Adaptor/ *Powerbank*

Sumber tegangan yang digunakan oleh sensor node adalah menggunakan adaptor/ *powerbank* yang *output* dari sumber tegangan adalah 5 Volt.

#### 4.2.3 Kebutuhan *Software*

Adapun kebutuhan *software* yang digunakan adalah sebagai berikut :

##### 1. Windows 10 Pro 64 *bit*

Perangkat lunak ini digunakan sebagai sistem operasi pada komputer.

## 2. Google Chrome

Perangkat lunak ini digunakan untuk menampilkan hasil visual dari data sensor yang diterima oleh sistem.

## 3. Node JS

Perangkat ini digunakan sebagai penyedia data yang akan ditampilkan pada web browser. Server melakukan subscribe data ke broker kemudian

## 4. Arduino IDE

Perangkat lunak ini digunakan untuk menuliskan kode program dan *compile* dan *upload* ode ke NodeMCU. Perangkat ini juga dapat digunakan untuk menampilkan data sensor.

## 5. HiveMQ Broker

Perangkat lunak ini digunakan *broker* yaitu yang menjembatani antara *publisher* dengan *subscriber*.

## 6. Microsoft Visual Code

Perangkat lunak ini merupakan salah satu teks editor yang digunakan untuk menulis ataupun *edit* kode program.

## 7. Command Prompt

Perangkat lunak ini berfungsi untuk memberikan perintah kepada computer untuk menjalankan suatu aktivitas tertentu.

## 8. *Library*

Pada sistem ini, menggunakan beberapa *library* untuk memudahkan peneliti dalam menyajikan hasil penelitian. Beberapa *library* yang digunakan antara lain:

### a. ESP8266Wifi.h

*Library* ini digunakan pada sensor node yang berfungsi untuk menghubungkan Node MCU ke broker MQTT.

### b. PubSubClient.h

*Library* ini digunakan pada sensor node yang berfungsi untuk publish data sensor menuju ke broker MQTT.

### c. Socket io

*Library* ini digunakan pada server node js yang berfungsi untuk memperoleh data secara realtime.

### d. mqtt

*library* ini digunakan pada server node js yang berfungsi untuk menghubungkan server dengan broker dan melakukan subscribe data.

### 4.3 Batasan Desain Sistem

Dalam pembuatan Sistem Monitoring Infus ini digunakan beberapa batasan sehingga lingkup pembahasan, perancangan maupun implementasinya tidak terlalu meluas. Adapun batasan-batasan desain sistem pada penelitian ini adalah sebagai berikut.

1. Volume infus yang digunakan adalah 100 mL.
2. Jenis selang infus yang digunakan adalah selang infus makro yang memiliki nilai 20 drop/ mL. Artinya, dalam 100 mL infus akan menghasilkan 2000 tetes cairan infus.
3. Lingkup sistem adalah lokal jaringan.
4. Sistem peringatan infus habis hanya terdapat pada sisi software.



## BAB 5 PERANCANGAN DAN IMPLEMENTASI

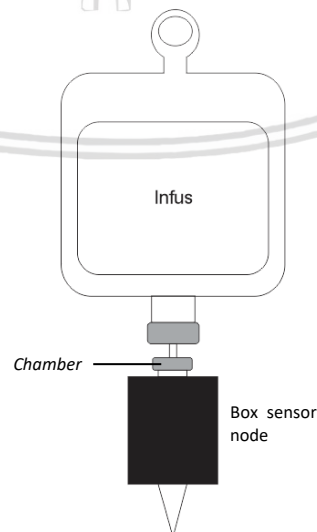
Pada bab ini berisi perancangan dan implementasi dari penelitian yang dilakukan yaitu “Implementasi Protokol MQTT (Message Queuing Telemetry Transport) Untuk Monitoring Infus Pasien Secara Terpusat”. Pada sub bab perancangan membahas perancangan perangkat prototype, perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*). Pada sub bab implementasi membahas implementasi prototype, implementasi perangkat keras (*hardware*) dan implementasi perangkat lunak (*software*).

### 5.1 Perancangan Sistem

Pada sub-bab ini akan dijelaskan mengenai cara perancangan sistem mulai dari perancangan prototype, perancangan perangkat keras hingga perancangan perangkat lunak.

#### 5.1.1 Perancangan *Prototype* Pada Node

Perancangan *prototype* dilakukan untuk menghasilkan sebuah sensor node yang dapat melakukan sensing tetesan infus. Perlu diperhatikan cara pemasangan sensor photodiode pada *prototype* dikarenakan sensor ini memiliki kepekaan yang tinggi terhadap cahaya. Apabila cahaya yang didapatkan memiliki intensitas cahaya yang terlalu tinggi, maka dapat mengganggu proses sensing. Posisi dari sensor photodiode diletakkan didalam box yang tidak tembus terhadap cahaya sehingga sensor berada dalam kondisi tempat yang gelap guna mendapatkan hasil pembacaan yang maksimal. Kemudian pada perangkat node terdapat modul NodeMCU yang diletakkan didalam box. NodeMCU berfungsi sebagai media untuk menerima data dari sensor, mengolah data sensor dan sebagai transmisi pengiriman data menuju *server*.

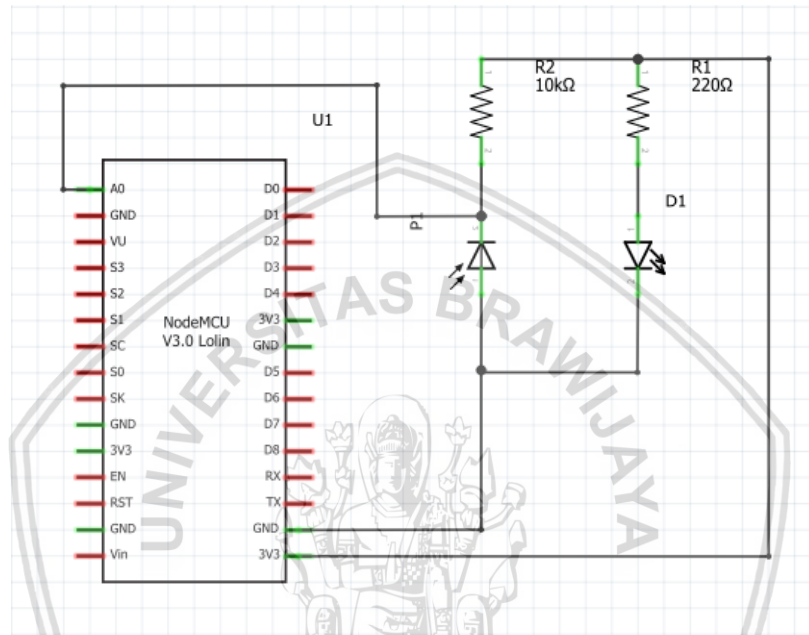


Gambar 5.1 Perancangan Prototype Sensor Node



### 5.1.2 Perancangan Perangkat Keras

Perancangan perangkat keras ini dilakukan berdasarkan analisis kebutuhan perangkat keras yang digunakan oleh sistem beserta spesifikasinya sehingga didapatkan sistem yang dapat bekerja sesuai dengan yang diharapkan peneliti. Pada perancangan ini, dijabarkan skematik dari perangkat keras yang digunakan dalam membangun sistem. Perangkat keras yang digunakan antara lain sensor photodiode, LED, resistor dan modul NodeMCU. Diagram skematik dari perancangan perangkat keras pada penelitian terdapat pada Gambar 5.2.



Gambar 5.2 Skematik Sensor Node

Pada diagram skematik yang terdapat pada Gambar 5.2, dapat dilihat bahwa terdapat sensor photodiode dan LED yang terhubung dengan sumber tegangan yang berasal dari NodeMCU. Kaki anoda pada photodiode terhubung dengan ground, sedangkan kaki katoda terhubung dengan resistor sebesar 10 kΩ. Adapun pin yang digunakan pada NodeMCU terdapat pada Table 5.1

Tabel 5.1 Pin yang digunakan pada NodeMCU

Pin Arduino	Pin Sensor Photodiode
VCC	VCC
GND	GND
A0	Data

Pemasangan sensor photodiode dilakukan secara berhadapan-hadapan dengan LED, serta diberikan jarak untuk peletakan *chamber* infus. Peletakan tersebut merupakan cara untuk mendapatkan data sensing, ketika tidak ada



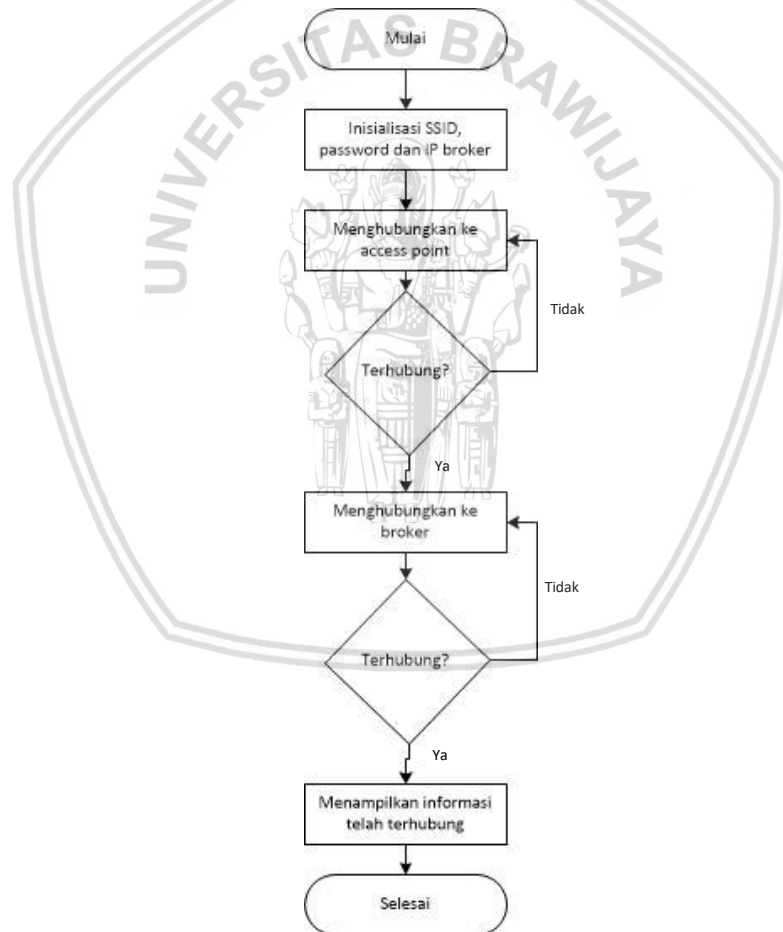
halangan (tetesan infus), sensor memberikan nilai analog yang besar, sedangkan jika terdapat tetesan infus, sensor memberikan nilai analog yang lebih kecil.

### 5.1.3 Perancangan Perangkat Lunak

Pada subbab perancangan perangkat lunak ini menjelaskan mengenai beberapa tahap perancangan pada perangkat lunak hingga terbentuk menjadi sebuah satu kesatuan sistem yang dapat berjalan.

#### 5.1.3.1 Perancangan Koneksi ke Access Point dan Broker

Perancangan ini merupakan perancangan awal yang dilakukan pada pembuatan sistem monitoring infus. Sebelum perangkat dapat berkomunikasi dengan perangkat lain, maka membutuhkan perangkat penghubung yang pada penelitian ini menggunakan jaringan *wifi*. Sensor node harus terhubung dengan jaringan *wifi* lokal dan terhubung dengan server. Berikut ini adalah *flowchart* sistem dalam melakukan koneksi terhadap *access point* dan *broker*.



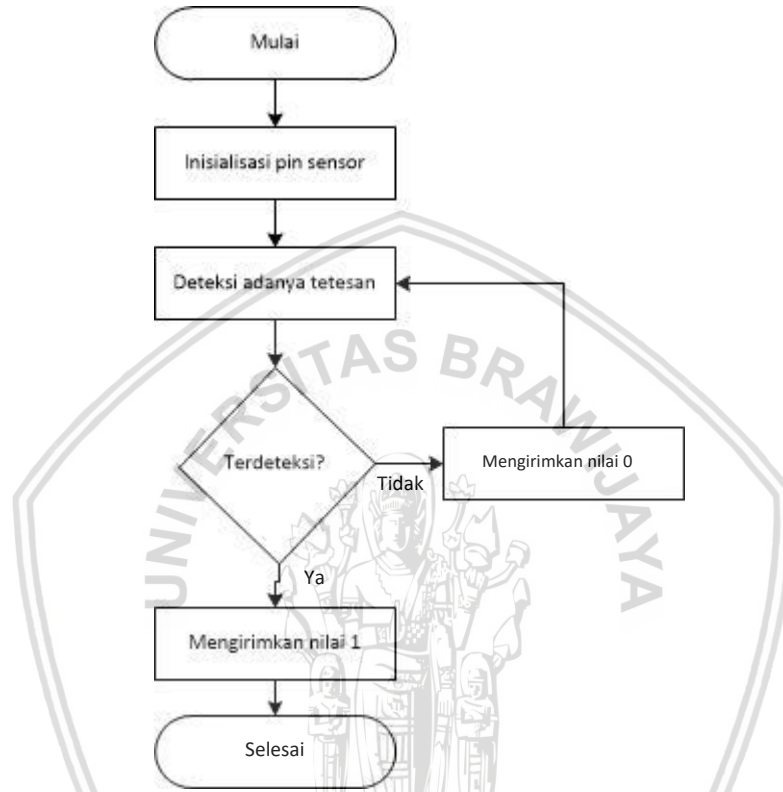
**Gambar 5.3 Flowchart untuk menghubungkan *hardware* dengan *access point* dan *broker***

Dari Gambar 5.3 dapat dijelaskan, pertama-tama membutuhkan inisialisasi variabel yang digunakan oleh program. Setelah itu, perangkat keras akan mencoba menghubungkan ke *access point* tujuan. Apabila ditemukan *access point* yang



dicari, maka perangkat akan terhubung dengan *access point* dan melakukan langkah selanjutnya. Kemudian, setelah terhubung dengan *access point*, perangkat akan menghubungkan dengan server dan ketika sudah terhubung, perangkat melakukan *publish* ke server untuk memberitahu server.

### 5.1.3.2 Perancangan Pengambilan dan Pengiriman Data



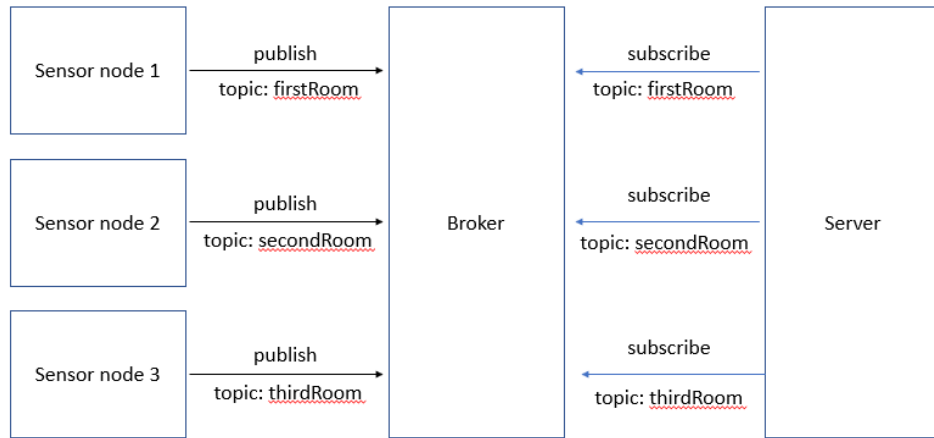
**Gambar 5.4 Diagram alir perancangan pengambilan dan pengiriman data**

Dalam melakukan proses pengambilan dan pengiriman data sensor menuju ke server dapat dilihat pada Gambar 5.4 yang mana tahap pertama adalah inisialisasi pin yang digunakan oleh sensor. Kemudian sensor melakukan pembacaan terhadap tetesan infus. Sebelum dilakukan pengiriman data, nilai sensor dibandingkan dengan *threshold*. Ketika nilai sensor kurang dari nilai *threshold* maka akan dilakukan pengiriman nilai 1 ke server. Apabila nilai sensor melebihi *threshold*, maka akan dilakukan pengiriman nilai 0 ke server.

### 5.1.3.3 Perancangan Protokol MQTT Sebagai Metode Pengiriman Data

Pada perancangan ini, terdapat tiga komponen utama yang berkomunikasi dan bertukar data ditampilkan pada Gambar 5.5.



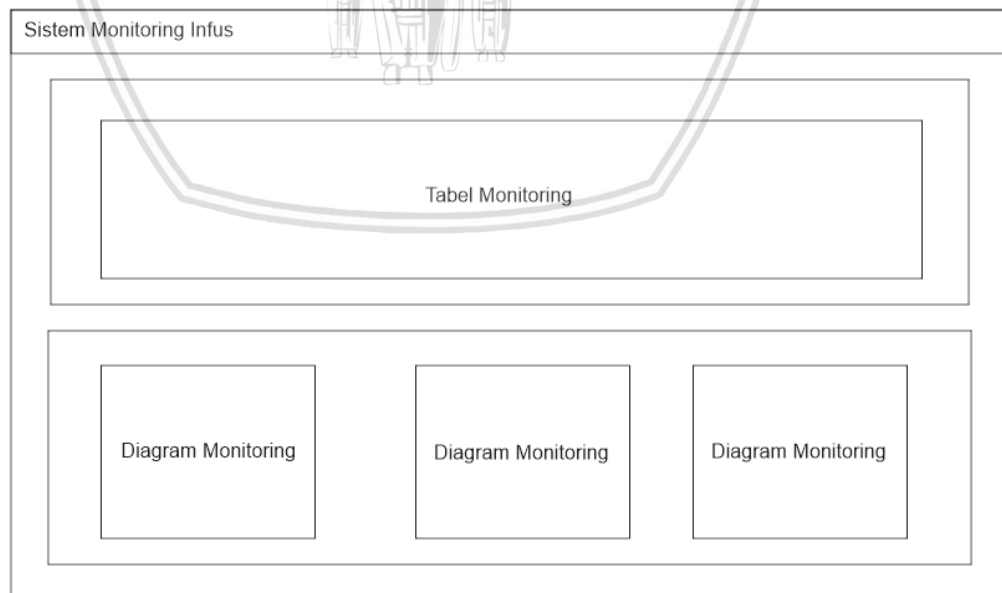


**Gambar 5.5 Diagram blok perancangan MQTT**

Berdasarkan Gambar 5.5, sensor node bertindak sebagai publisher dengan mentransmisikan data ke broker dengan topik tertentu. Adapun topik yang digunakan adalah *firstRoom* untuk sensor node 1, *secondRoom* untuk sensor node 2, dan *thirdRoom* untuk sensor node 3. Ketiga sensor melakukan publish ke broker HiveMQ. Server bertindak sebagai *subscriber* dengan men-*subscribe* data dengan topik *firstRoom*, *secondRoom*, dan *thirdRoom*.

**5.1.3.4 Perancangan Web Interface**

Pada web interface, data yang tampil pada halaman browser diperoleh dari server node js. Data mentah yang didapatkan kemudian diproses didalam fungsi yang terdapat pada kode program web. Data tersebut kemudian ditampilkan pada halaman web browser dalam bentuk tabel dan diagram.



**Gambar 5.6 Perancangan Web Inteface**



Berdasarkan Gambar 5.6, terdapat 2 media untuk menampilkan data yang diperoleh dari sensor node. Bagian pertama berisi data yang telah diproses pada web, meliputi jumlah tetesan infus yang mengindikasikan jumlah tetesan infus yang sudah terbaca oleh sensor, persentase kapasitas infus yang mengindikasikan jumlah kapasitas infus yang masih tersedia pada kantong infus, dan status infus yang mengindikasikan status infus. Cara penyajian dilakukan dalam bentuk table dan indikator berbentuk silinder. Penyajian data dilakukan dalam bentuk tabel tidak efektif apabila terdapat kondisi petugas medis tidak berada didepan monitor, agar monitoring dapat dijangkau dari jarak pandang yang jauh maka dapat dilakukan dengan menggunakan indikator tersebut.

Pada kolom jumlah tetesan infus, data diperoleh dari perhitungan *counter* dimana nilai *counter* akan bertambah jika data yang diperoleh dari sensor node adalah perubahan nilai dari 1 ke 0.

Pada kolom persentase kapasitas infus, data diperoleh dari **Persamaan 5. 1:**

$$\text{Persentase kapasitas} = \frac{\text{Jumlah seluruh tetesan} - \text{nilai counter}}{\text{Jumlah seluruh tetesan}} \times 100\% \quad (5.1)$$

Pada kolom status infus, terdapat 3 indikator label teks dan 4 indikator warna. Indikator berwarna biru jika persentase kapasitas infus lebih dari 35%, berwarna hijau jika persentase kapasitas infus antara 25 – 35%, berwarna kuning jika persentase kapasitas infus antara 15 – 25% dan berwarna merah jika persentase kapasitas infus antara 0 – 15%. Jika persentase kapasitas infus kurang dari 15%, halaman monitoring akan menampilkan pesan peringatan dan membunyikan *sound* komputer sebagai penanda bahwa infus hampir habis. Pesan peringatan dan sound dapat diberhentikan dengan menekan tombol berhenti.

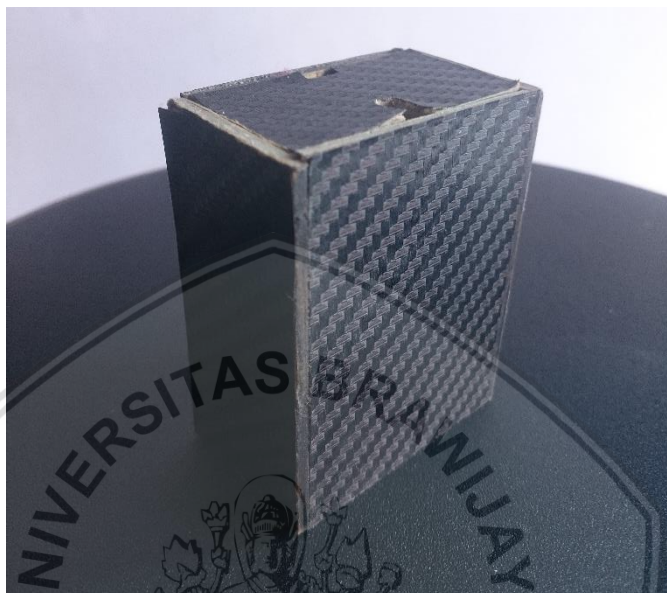
Perhitungan threshold didapatkan dari asumsi akurasi sensor jika lebih dari 90%. Artinya, dari 100 tetesan akan terdeteksi sebanyak 90 tetesan. Seperti yang sudah dijelaskan sebelumnya, disini peneliti menggunakan selang infus makro yang memiliki intensitas *drop* sebesar 20 *drop/ml* dan kantong infus dengan volume 100 ml. Dari 100 ml, akan menghasilkan sebanyak 2000 tetesan infus. Jika akurasi sensor sebanyak 90%, maka dari 2000 tetesan akan terdeteksi maksimal sebanyak 1800 tetesan saja. Jika threshold yang diambil adalah sebesar 15%, maka peringatan akan muncul pada tetesan ke-1700. Artinya, kondisi tersebut tidak dikhawatirkan jika infus habis namun peringatan tidak muncul.

## 5.2 Implementasi Sistem

Pada tahap implementasi merupakan tahap untuk merealisasikan perancangan yang telah dibuat. Pada bab ini akan dibahas mengenai implementasi *prototype*, implementasi perangkat keras dan implementasi perangkat lunak.

### 5.2.1 Implementasi *Prototype* Node Sensor

Pada perancangan ini mengacu pada perancangan yang terdapat di subbab 5.1.1 yaitu perancangan *prototype* node sensor dikemas pada box hitam dengan ketentuan panjang x lebar x tinggi adalah 10 x 6 x 4 cm. Untuk membuat box hitam, bahan yang digunakan adalah kertas karton dengan ketebalan 3mm. Hasil implementasi *prototype* ditunjukkan pada Gambar 5.7.



Gambar 5.7 Implementasi *Prototype* Sensor Node

*Prototype* yang telah diletakkan pada *chamber infus* dengan cara mengaitkan *chamber infus* ke dalam box *prototype* sensor node pada Gambar 5.8.

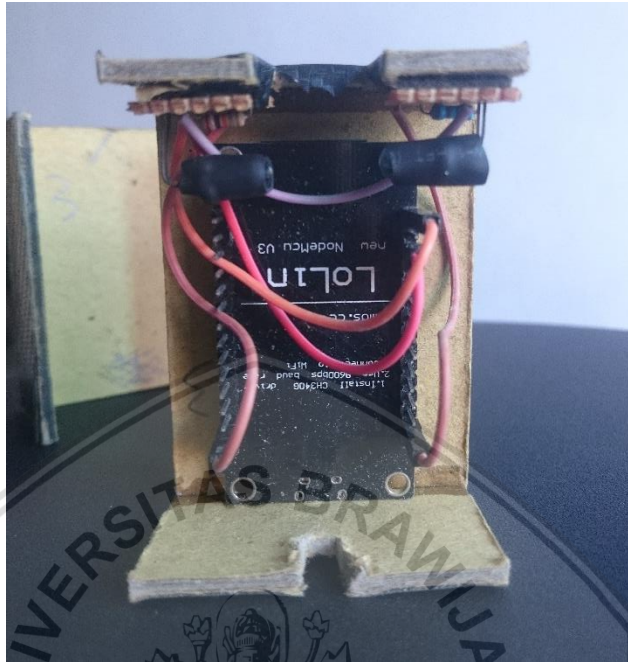


Gambar 5.8 Sensor node pada *chamber infus*

### 5.2.2 Implementasi Perangkat Keras

Pada subbab ini menjelaskan proses implementasi *prototype*, implementasi dari perangkat keras yang digunakan, dan implementasi perangkat

lunak. Proses implementasi dibuat berdasarkan perancangan yang telah dijabarkan pada sub bab perancangan.



**Gambar 5.9 Implementasi Perangkat Keras**

Pada Gambar 5.5 menunjukkan hasil implementasi perangkat keras yang telah dirangkai sesuai dengan perancangan, yaitu sensor photodiode diletakkan pada bagian dalam box hitam yang disusun berhadap-hadapan dengan LED dan diberi jarak 2 cm yang digunakan untuk meletakkan *chamber* infus.

### **5.2.3 Implementasi Perangkat Lunak**

Implementasi perangkat lunak menjelaskan realisasi program yang digunakan pada Sistem Monitoring Infus berdasarkan perancangan pada subbab 5.1.3. Dalam melakukan implementasi perangkat lunak, proses pengerjaan kode program dilakukan menggunakan Arduino IDE dan Microsoft Visual Code. Untuk membuat kode program yang digunakan pada node sensor menggunakan Arduino IDE, sedangkan untuk membuat kode program yang digunakan pada server node js dan *web browser* menggunakan Microsoft Visual Code.

#### **5.2.3.1 Implementasi Kode Program Koneksi ke *Access Point* dan *Broker***

Pada implementasi untuk menghubungkan perangkat keras ke *access point*, diperlukan satu library yaitu ESP8266Wifi.h yang berfungsi untuk memudahkan untuk menghubungkan perangkat keras ke *access point*. Selain itu, diperlukan inisialisasi variabel yang digunakan pada proses koneksi menuju *access point*. Setelah sensor node terhubung dengan *access point*, proses dilanjutkan

dengan menghubungkan sensor node ke server. Kode program yang berfungsi untuk menghubungkan sensor node ke server terdapat pada **Tabel 5.2** sampai .

**Tabel 5.2 Kode program untuk *import library* dan inisialisasi variabel**

Sensornode.ino	
1	#include <ESP8266WiFi.h>
2	#include <PubSubClient.h>
3	
4	const char* ssid = "Piranha";
5	const char* password = "passwordLama";
6	const char* mqtt_server = "192.168.100.6";
7	
8	WiFiClient espClient;
9	PubSubClient client(espClient);
	...

Implementasi kode program pada sensor node dalam menghubungkan ke access point diawali dengan pemanggilan *library* dan deklarasi variabel-variabel yang akan digunakan. Dari **Tabel 5.2** dapat dijabarkan, pada baris ke 1 – 2 adalah pemanggilan library, yaitu pada baris ke-1 pemanggilan library yang digunakan untuk menghubungkan sensor node ke *access point*. Kemudian pada baris ke-2 dilakukan pemanggilan *library* yang digunakan untuk menghubungkan sensor node ke broker.

Untuk inisialisasi variabel terdapat pada baris ke 4 – 6, dimana pada baris ke-4 mendeklarasikan nama dari access point yang digunakan, baris ke-5 mendeklarasikan password yang digunakan oleh access point, dan baris ke-6 mendeklarasikan alamat IP dari broker.

**Tabel 5.3 Kode program menghubungkan sensor node ke *access point***

Sensornode.ino	
1	...
2	void setup_wifi() {
3	delay(10);
4	Serial.println();
5	Serial.print("Menghubungkan ke ");
6	Serial.println(ssid);
7	WiFi.begin(ssid, password);
8	while (WiFi.status() != WL_CONNECTED) {
9	delay(500);
10	Serial.print(".");
11	}

12	<code>randomSeed(micros());</code>
13	<code>Serial.println("");</code>
14	<code>Serial.println("WiFi Terhubung");</code>
15	<code>Serial.println("Alamat IP: ");</code>
16	<code>Serial.println(WiFi.localIP());</code>
17	<code>}</code>

Selanjutnya pada Tabel 5.3 merupakan sebuah fungsi untuk menghubungkan sensor node ke *access point*. Pada baris ke-7, program memanggil fungsi yang terdapat pada library ESP8266Wifi yaitu fungsi *begin* dan mengirimkan parameter yaitu *ssid* dan *password*. Fungsi *begin* merupakan salah satu fungsi yang digunakan pada ESP8266Wifi untuk memulai menjalankan fungsi-fungsi yang lain didalamnya hingga perangkat berhasil terkoneksi dan mendapatkan alamat IP dari *access point*.

**Tabel 5.4 Kode program fungsi untuk menghubungkan sensor node ke broker**

Sensornode.ino	
1	<code>...</code>
2	<code>void reconnect() {</code>
3	<code>  while (!client.connected()) {</code>
4	<code>    Serial.print("Menghubungkan ke server...");</code>
5	<code>    String clientId = "ESP8266Client-Room1";</code>
6	<code>    if (client.connect(clientId.c_str())) {</code>
7	<code>      Serial.println("Terhubung");</code>
8	<code>      client.publish("firstRoom", "Kamar 1 Terhubung...");</code>
9	<code>    } else {</code>
10	<code>      Serial.print("Gagal, state=");</code>
11	<code>      Serial.print(client.state());</code>
12	<code>      Serial.println(" Mencoba kembali dalam 5 detik.");</code>
13	<code>      delay(5000);</code>
15	<code>    }</code>
16	<code>  }</code>
17	<code>}</code>
18	<code>...</code>

Pada Tabel 5.4 merupakan kode program yang digunakan untuk menghubungkan sensor node ke broker MQTT. Pada kode program tersebut, program akan melakukan perulangan dengan *while* dan perulangan tidak akan berhenti hingga sensor node berhasil terkoneksi dengan *broker* MQTT. Jika sensor node berhasil terhubung maka akan melakukan *publish* sebagai penanda bahwa



sensor node telah terhubung. Jika sensor node gagal terhubung dengan *broker* MQTT, maka program akan terus berjalan hingga dapat terhubung dengan *broker*.

**Tabel 5. 5 Kode program untuk menjalankan fungsi-fungsi yang ada**

Sensornode.ino	
1	...
2	void setup() {
3	...
4	setup_wifi();
5	client.setServer(mqtt_server, 1883);
6	client.setCallback(callback);
7	...
8	}
9	void loop() {
10	if (!client.connected()) {
11	reconnect();
12	}
13	client.loop();
14	...
15	}

Pada Tabel 5. 5 merupakan program utama untuk menjalankan fungsi-fungsi yang digunakan untuk menghubungkan sensor node ke *access point* dan *broker*.

### 5.2.3.2 Implementasi Kode Program Pengambilan dan Pengiriman Data

Pada implementasi ini, pengambilan data sensor dilakukan dengan inisialisasi pin yang digunakan oleh sensor terlebih dahulu. Pin yang digunakan sensor adalah pin analog sehingga cara pembacaan oleh program Arduino adalah dengan menggunakan fungsi *analogRead()* yang terdapat pada Serial Arduino. Nilai sensor yang didapatkan pada saat tidak ada tetesan adalah 1024 dan bernilai kurang dari 200 pada saat terdeteksi tetesan. Nilai tersebut tidak dapat konstan akibat adanya interferensi cahaya lain yang mempengaruhi nilai sensor tersebut. Namun, nilai rata-rata sensor pada saat mendeteksi adanya tetesan adalah kurang dari 500, sehingga nilai tersebut dijadikan sebagai *threshold* dilakukannya pengiriman data ke server. Berikut ini adalah kode program pengambilan dan pengiriman data sensor yang disajikan pada Tabel 5.6.

**Tabel 5.6 Kode program pengambilan dan pengiriman data**

Sensornode.ino	
1	int threshold = 500;
2	...

```
3 void setup() {
4     ...
5     pinMode(A0, INPUT);
6     ...
7 }
8 ...
9 void loop() {
10    int val = analogRead(A0); // read input value
11    if(val < threshold) {
12        client.publish(<topik>, "1");
13    }
14    else{
15        client.publish(<topik>, "0");
16    }
17 }
18 ...
```

Dari potongan kode program yang disajikan pada Tabel 5.6, dapat dijelaskan tahapan dalam melakukan pembacaan hingga pengiriman data menuju ke server. Pertama adalah inialisasi variabel yang digunakan. Pada baris ke-1 merupakan inialisasi *threshold* dari nilai sensor. Penentuan nilai *threshold* dilakukan dengan melakukan pembacaan nilai sensor pada masing-masing sensor node ketika sedang mendeteksi adanya tetesan. Kemudian menghitung nilai rata-rata dari ketiga hasil pembacaan.

Kemudian pada baris ke-5 adalah untuk menentukan pin yang digunakan oleh sensor dan mengatur pin A0 menjadi pin *INPUT*, dimana pin A0 dijadikan sebagai pin masukan. Kemudian untuk membaca nilai analog pada pin A0 terdapat pada baris ke-10 dengan menggunakan fungsi *analogRead()*. Kemudian nilai analog disimpan pada variabel *val* yang kemudian digunakan sebagai pembanding dengan nilai *threshold*. Pada kode program diatas, nilai *threshold* adalah 500. Jika nilai sensor yang terbaca kurang dari *threshold*, maka sensor node mem-*publish* nilai 1 ke broker. Sedangkan, jika nilai sensor lebih dari *threshold* maka sensor node mem-*publish* nilai 0 ke broker.

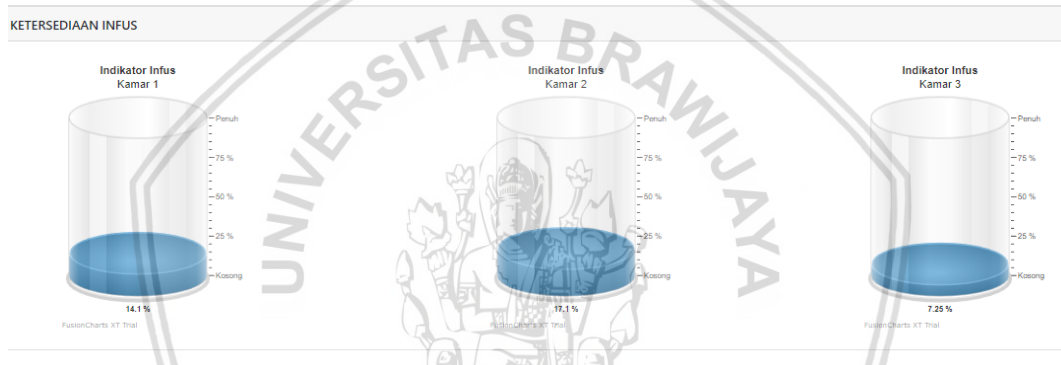
### 5.2.3.3 Implementasi Web Interface

Pada implementasi *web interface* dibuat sedemikian rupa sehingga menghasilkan *interface* monitoring infus dengan data *realtime*.

TABLE MONITORING			
Kamar	Jumlah Tetes	Persentase Ketersediaan	Status
Kamar 1	1718/ 2000 tetes	14.10%	Warning
Kamar 2	1658/ 2000 tetes	17.10%	Warning
Kamar 3	1855/ 2000 tetes	7.25%	Infus Hampir Habis

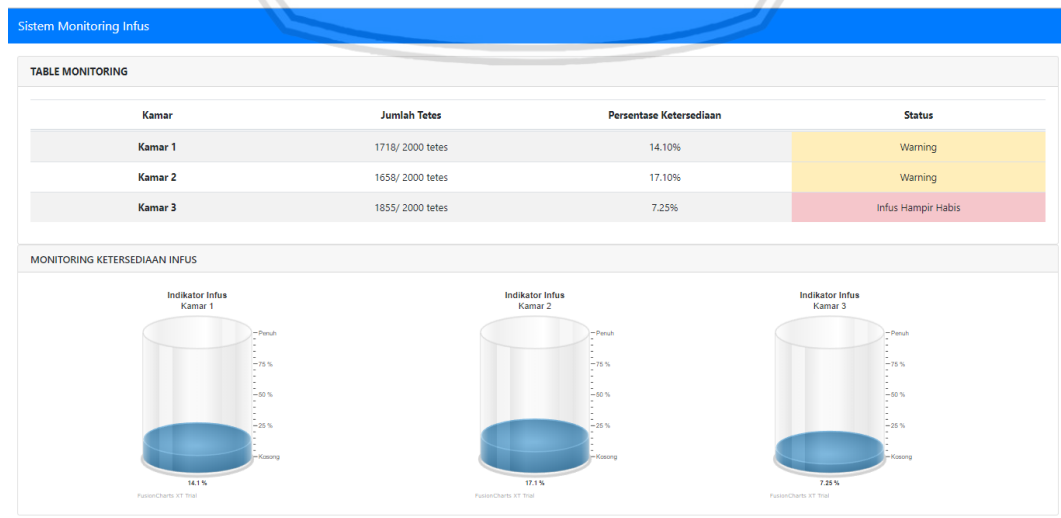
**Gambar 5.10** Tabel untuk menampilkan data monitoring

Pada Gambar 5.10 merupakan tabel yang digunakan untuk menampilkan data sensor yang telah diproses didalam program web. Nilai pada masing-masing kolom diperoleh dari server.



**Gambar 5.11** Diagram silinder sebagai indikator ketersediaan infus

Pada Gambar 5.11 merupakan diagram silinder yang mengindikasikan ketersediaan infus pada ruang monitoring. Data yang ditampilkan adalah persentase ketersediaan infus. Data ditampilkan secara *realtime*. Tampilan web secara keseluruhan disajikan pada Gambar 5.12.



**Gambar 5.12** Tampilan web secara keseluruhan



## BAB 6 PENGUJIAN

Pada bab ini, membahas pengujian dari sistem yang telah dibuat. Maksud dan tujuan dari dilakukannya pengujian sistem adalah untuk mengetahui apakah semua kebutuhan yang diharapkan sudah terpenuhi oleh sistem.

### 6.1 Pengujian Sensor Photodiode

Sensor photodiode adalah sensor yang digunakan untuk mendeteksi adanya cahaya. Pada penelitian ini, sensor digunakan untuk mendeteksi tetesan infus. Pengujian ini dilakukan dengan menjalankan sistem node sehingga didapatkan data tetesan pada serial monitor. Kemudian, hasilnya akan dibandingkan dengan hasil penglihatan manual.

#### 6.1.1 Tujuan Pengujian

Tujuan dari pengujian sensor photodiode adalah untuk mengetahui tingkat akurasi sensor dalam mendeteksi objek. Dengan dilakukan perbandingan pada nilai yang dibaca sensor dengan penglihatan manual, maka akan diperoleh selisih nilai pembacaan sensor sehingga akan diketahui nilai error sensor yang digunakan.

#### 6.1.2 Prosedur Pengujian

Berikut ini adalah prosedur pengujian sensor photodiode:

1. Menghubungkan sensor photodiode dengan NodeMCU.
2. Memasang sensor node pada *chamber* infus.
3. Menghubungkan NodeMCU dengan laptop.
4. Mengunggah kode program ke NodeMCU.
5. Mengatur kecepatan tetesan infus.
6. Membuka serial monitor pada Arduino IDE.
7. Menghitung tetesan infus sebanyak kelipatan 5 kali tetes dan mengamati tetesan secara manual dan mencatat hasilnya. Pengujian diulang hingga 5 kali pada masing-masing node.
8. Membandingkan hasil yang ditampilkan pada serial monitor dengan hasil yang diperoleh dari pengamatan secara manual. Dari kedua data tersebut diambil dan dilakukan perhitungan persentase akurasi sensor dalam bentuk nilai desimal.

Berikut ini adalah **Persamaan 6. 1** yang diberikan untuk menghitung persentase akurasi sensor:

$$\text{Persentase akurasi sensor} = \frac{\text{Pembacaan sensor}}{\text{Pembacaan manual}} \times 100\% \quad (6.1)$$

### 6.1.3 Hasil dan Analisis Pengujian

Dari beberapa kali pengujian sensor photodiode, diperoleh data sebagai berikut:

**Tabel 6.1 Pengujian sensor photodiode node pertama**

Pengujian Ke-	Pembacaan Manual	Pembacaan Sensor
1	5 tetesan	5 tetesan
2	10 tetesan	10 tetesan
3	15 tetesan	14 tetesan
4	20 tetesan	19 tetesan
5	25 tetesan	23 tetesan
<b>Total</b>	75 tetesan	71 tetesan

Dari hasil pengujian sensor photodiode pada node pertama yang terdapat pada Tabel 6.1, dapat dihitung dengan persamaan diatas sebagai berikut:

$$\text{Persentase akurasi sensor} = \frac{\text{pembacaan sensor}}{\text{Pembacaan manual}} \times 100\%$$

$$\text{Persentase akurasi sensor} = \frac{72}{75} \times 100\%$$

$$\text{Persentase akurasi sensor} = \frac{72}{75} \times 100\%$$

$$\text{Persentase akurasi sensor} = 96\%$$

Dari perhitungan diatas, diperoleh persentase akurasi sensor photodiode pada node pertama adalah sebesar 96%.

**Tabel 6.2 Pengujian sensor photodiode node kedua**

Pengujian Ke-	Pembacaan Manual	Pembacaan Sensor
1	5 tetesan	5 tetesan
2	10 tetesan	9 tetesan
3	15 tetesan	15 tetesan
4	20 tetesan	20 tetesan
5	25 tetesan	23 tetesan
<b>Total</b>	75 tetesan	72 tetesan

Dari hasil pengujian sensor photodiode pada node kedua yang terdapat pada Tabel 6.2, dapat dihitung dengan persamaan diatas sebagai berikut:

$$\text{Persentase akurasi sensor} = \frac{\text{pembacaan sensor}}{\text{Pembacaan manual}} \times 100\%$$



$$\text{Persentase akurasi sensor} = \frac{72}{75} \times 100\%$$

$$\text{Persentase akurasi sensor} = \frac{72}{75} \times 100\%$$

$$\text{Persentase akurasi sensor} = 96\%$$

Dari perhitungan diatas, diperoleh persentase akurasi sensor photodiode pada node kedua adalah sebesar 96%.

**Tabel 6.3 Pengujian sensor photodiode node ketiga**

Pengujian Ke-	Pembacaan Manual	Pembacaan Sensor
1	5 tetesan	5 tetesan
2	10 tetesan	9 tetesan
3	15 tetesan	12 tetesan
4	20 tetesan	20 tetesan
5	25 tetesan	23 tetesan
<b>Total</b>	75 tetesan	70 tetesan

Dari hasil pengujian sensor photodiode pada node ketiga yang terdapat pada Tabel 6.3, dapat dihitung dengan persamaan diatas sebagai berikut:

$$\text{Persentase akurasi sensor} = \frac{\text{pembacaan sensor}}{\text{Pembacaan manual}} \times 100\%$$

$$\text{Persentase akurasi sensor} = \frac{71}{75} \times 100\%$$

$$\text{Persentase akurasi sensor} = \frac{71}{75} \times 100\%$$

$$\text{Persentase akurasi sensor} = 94.6\%$$

Dari perhitungan diatas, diperoleh persentase akurasi sensor photodiode pada node ketiga adalah sebesar 96%.

Dari tiga pengujian yang dilakukan pada masing-masing sensor node, terdapat perbedaan hasil pembacaan sensor photodiode terhadap tetesan infus. Hal tersebut dikarenakan adanya perbedaan keadaan cahaya pada masing-masing box sensor node. Besar kecilnya cahaya yang diterima pada box sensor node mempengaruhi nilai yang diterima oleh sensor photodiode.

## 6.2 Pengujian Pengiriman Data Sensor

Pengiriman data dilakukan dari sensor node menuju ke broker MQTT melalui jaringan wifi lokal. Sensor node mengirimkan data yaitu berupa hasil deteksi sensor terhadap tetesan infus pada *chamber*.



### 6.2.1 Tujuan Pengujian

Pengujian ini dilakukan untuk mengetahui tingkat keberhasilan pengiriman yang dilakukan oleh sensor node menuju ke server. Pengujian dilakukan dengan membandingkan data yang dikirim dari sensor node dan data yang diperoleh server. Dari kedua data tersebut dapat dihitung tingkat keberhasilan sistem pengiriman pada penelitian ini.

### 6.2.2 Prosedur Pengujian

Adapun prosedur Pengujian Pengiriman Data Sensor adalah sebagai berikut:

1. Menyiapkan sensor node dan mengatur alamat IP broker.
2. Menyiapkan server node js dan mengatur alamat IP dari broker.
3. Menyalakan server node js.
4. Menyalakan sensor node dan memulai mendeteksi adanya tetesan dengan memanfaatkan serial monitor pada Arduino IDE.
5. Mengamati dan mencatat hasil pembacaan sensor yang ditampilkan pada serial monitor dan yang ditampilkan pada command prompt server node js.
6. Membandingkan data sebelum dan sesudah dikirimkan oleh sensor node untuk mencari selisih yang akan digunakan untuk mengetahui nilai error dari pengiriman data.

### 6.2.3 Hasil dan Analisis Pengujian

Dari beberapa kali pengujian, didapatkan data yang disajikan dalam tabel berikut ini:

**Tabel 6.4 Pengujian Pengiriman Data Sensor**

Pengujian Ke-	Jumlah Tetesan yang diharapkan	Serial Monitor	Server
1	5 tetesan	5 tetesan	5 tetesan
2	10 tetesan	10 tetesan	10 tetesan
3	15 tetesan	15 tetesan	15 tetesan
4	20 tetesan	20 tetesan	20 tetesan
5	25 tetesan	25 tetesan	25 tetesan
<b>Total</b>	75	75 tetesan	75 tetesan

Dari tabel diatas dapat disimpulkan bahwa untuk pengiriman data yang dilakukan dari sensor node menuju ke server node js berhasil dilakukan dengan tingkat keberhasilan mencapai 100%. Hal tersebut dikarenakan pada sistem ini menggunakan jaringan lokal wifi.

## 6.3 Pengujian Delay Pengiriman Data Sensor

### 6.3.1 Tujuan Pengujian

Pengujian ini dilakukan untuk mengetahui durasi pengiriman data dari sensor node menuju ke server. Delay dapat diketahui dengan menghitung selisih waktu pengiriman pada sensor node dan ketika diterima oleh server.

### 6.3.2 Prosedur Pengujian

Adapun prosedur Pengujian Delay Pengiriman Data Sensor antara lain dibawah ini:

1. Menyiapkan kode program pada sensor node dengan menambahkan perintah untuk mencetak waktu *publish* data, dan menyiapkan kode program server dengan menambahkan perintah untuk mencetak waktu *subscribe* data.
2. Menyalakan sensor node, menyalakan broker, dan menyalakan server node js
3. Mendeteksi adanya tetesan infus sehingga sensor node akan mengirimkan data menuju ke server.
4. Membuka serial monitor dan mengamati waktu pengiriman data sensor.
5. Mengamati waktu penerimaan data pada command prompt server node js.
6. Menghitung delay pengiriman data sensor.

### 6.3.3 Hasil dan Analisis Pengujian

Dari pengujian yang dilakukan sebanyak 10 kali percobaan, mendapatkan hasil sebagai berikut:

Tabel 6.5 Hasil Pengujian Delay

Pengujian Ke-	Waktu Pengiriman	Waktu Penerimaan	Delay (ms)
1	18:32:12:341	18:32:12:756	415
2	18:32:15:105	18:32:15:532	427
3	18:32:18:233	18:32:18:687	454
4	18:32:21:156	18:32:21:559	403
5	18:32:23:209	18:32:23:567	358
6	18:32:26:522	18:32:26:977	455
7	18:32:29:140	18:32:29:602	462
8	18:32:31:111	18:32:31:675	564
9	18:32:35:437	18:32:35:809	372



10	18:32:38:766	18:32:39:403	636
<b>Total</b>			454,6

13/11/2018 18:32:12:341	18:32:12:341	Data received on 18:32:12:756
13/11/2018 18:32:15:105	18:32:15:105	Data received on 18:32:15:532
13/11/2018 18:32:18:233	18:32:18:233	Data received on 18:32:18:687
13/11/2018 18:32:21:156	18:32:21:156	Data received on 18:32:21:559
13/11/2018 18:32:23:209	18:32:23:209	Data received on 18:32:23:567
13/11/2018 18:32:26:522	18:32:26:522	Data received on 18:32:26:987
13/11/2018 18:32:29:140	18:32:29:140	Data received on 18:32:29:588
13/11/2018 18:32:31:111	18:32:31:111	Data received on 18:32:31:675
13/11/2018 18:32:35:437	18:32:35:437	Data received on 18:32:35:809
13/11/2018 18:32:38:766	18:32:38:766	Data received on 18:32:39:403

**Gambar 6.1 Hasil Pengujian Delay**

Dari hasil pengujian diatas, menghasilkan data yang dapat dilihat pada Tabel 6.5. Dari 10 kali percobaan pengiriman data dari sensor node ke server, dapat dihitung rata-rata delay yang terjadi adalah sebesar 454.6 ms. Rata-rata *delay* yang terjadi tergolong rendah dikarenakan antara *publisher* dan *subscriber* menggunakan jaringan lokal. Dengan menggunakan jaringan lokal, traffic data yang terjadi tidak terlalu banyak sehingga dapat meminimalisir terjadinya *delay* pengiriman.

## 6.4 Pengujian Tampilan Pada Web Interface

*Web interface* digunakan untuk menampilkan hasil pembacaan sensor yang diperoleh dari server.

### 6.4.1 Tujuan Pengujian

Pengujian web interface bertujuan untuk mengetahui apakah data yang diperoleh sudah sesuai dengan data yang dibaca oleh sensor dan diterima oleh server.

### 6.4.2 Prosedur Pengujian

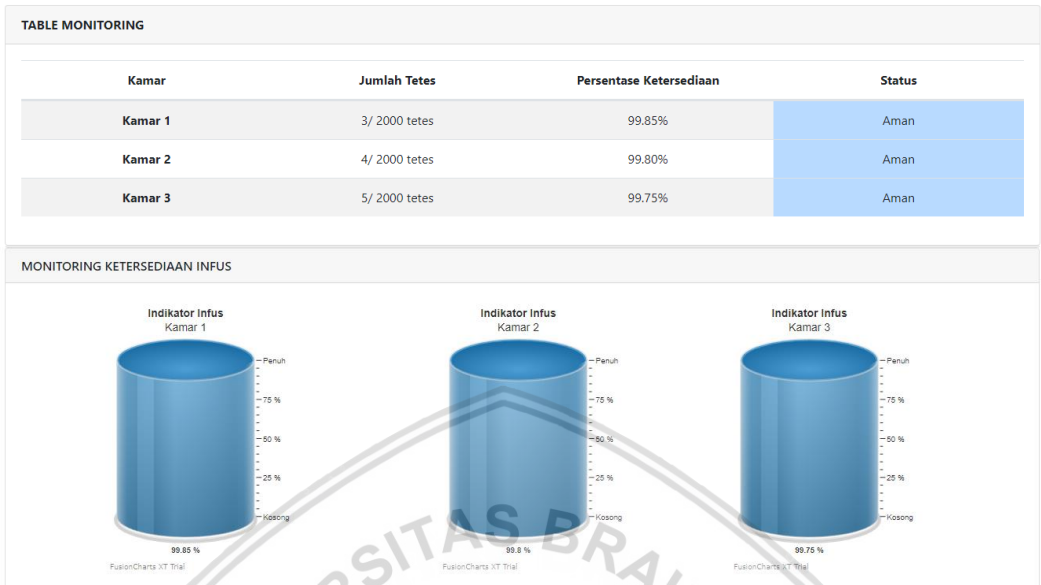
Adapun prosedur pengujian ini adalah sebagai berikut:

1. Menyiapkan sensor node dan menyambungkan ke broker.
2. Menyalakan server node js.
3. Membuka halaman web dengan menggunakan browser.
4. Mengamati data yang ditampilkan pada web browser apakah sudah sesuai dengan yang diharapkan.

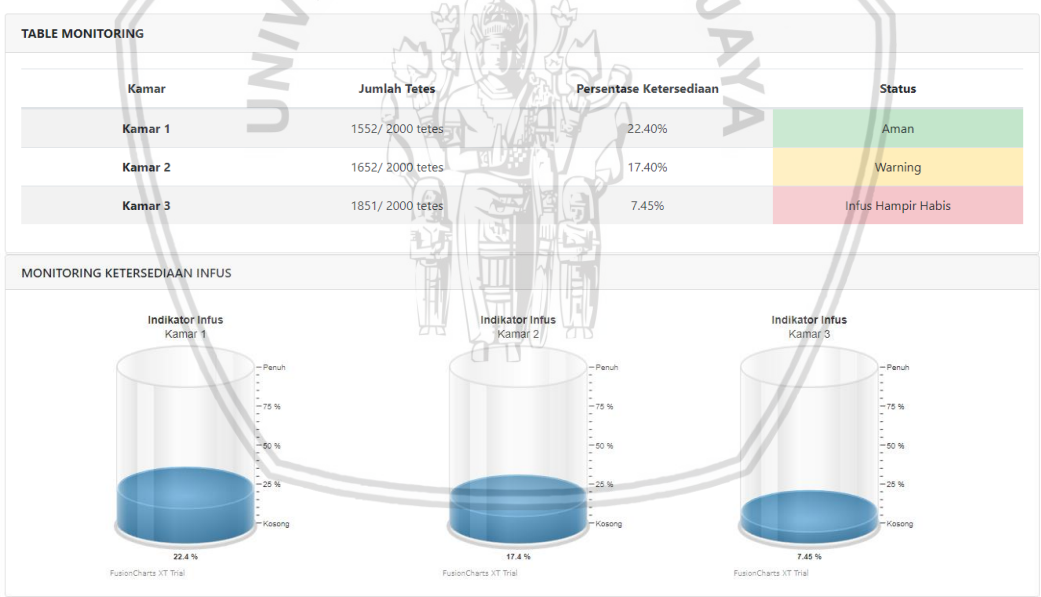
### 6.4.3 Hasil dan Analisis Pengujian

Di bawah ini merupakan gambar hasil dari pengujian web interface.

Sistem Monitoring Infus



Sistem Monitoring Infus



**Gambar 6.2 Hasil Pengujian Web Interface**

Gambar 6.2 merupakan hasil dari pengujian tampilan halaman web. Dari hasil tersebut menunjukkan bahwa halaman web telah berhasil menampilkan data yang diperoleh dari server node js.

**6.5 Pengujian Sistem Peringatan Pada Web Interface**

Sistem peringatan digunakan untuk memberikan informasi kepada petugas rumah sakit bahwa infus akan segera habis.



### 6.5.1 Tujuan Pengujian

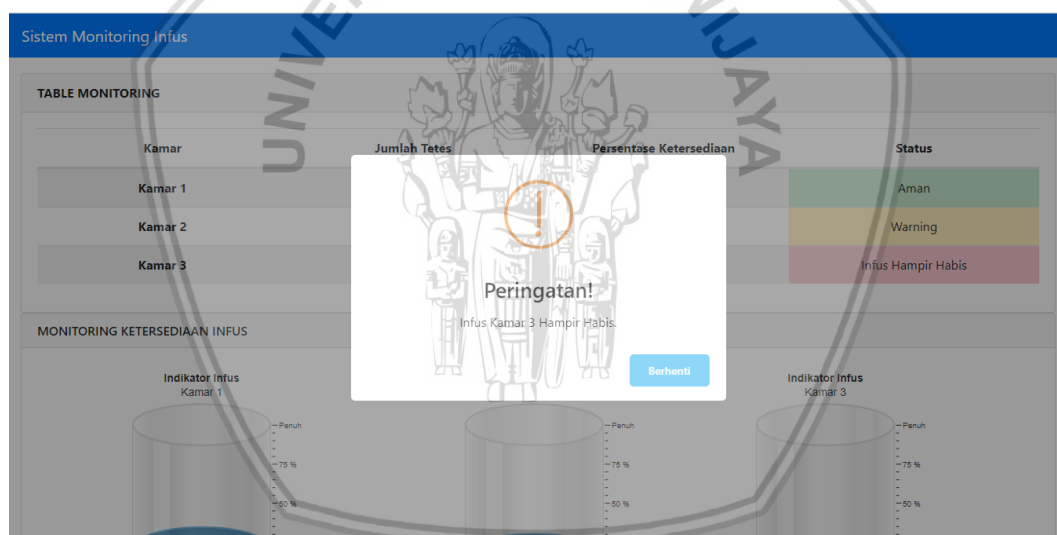
Pengujian sistem peringatan pada web interface bertujuan untuk mengetahui apakah implementasi yang dilakukan pada penelitian sudah berjalan sesuai harapan atau belum sesuai harapan.

### 6.5.2 Prosedur Pengujian

Adapun prosedur pengujian sistem peringatan pada web interface antara lain:

1. Menyiapkan sensor node dan menyambungkan ke broker.
2. Menyalakan server node js.
3. Membuka halaman web dengan menggunakan browser.
4. Menunggu hingga infus hampir habis.
5. Mengamati halaman web apakah sudah menampilkan peringatan ketika infus hampir habis (persentase infus kurang dari 10%).

### 6.5.3 Hasil dan Analisis Pengujian



Gambar 6. 3 Pengujian Sistem Peringatan

Dari Gambar 6. 3 diatas dapat disimpulkan jika sistem peringatan telah bekerja sesuai dengan yang diharapkan. Pada halaman browser dapat menampilkan peringatan ketika infus hampir habis.

## BAB 7 PENUTUP

Pada bab ini melakukan pengambilan kesimpulan dari keseluruhan penelitian yang dilakukan. Selain itu, pada bab ini terdapat saran yang dapat digunakan untuk penelitian selanjutnya yang mana untuk membuat sistem yang lebih efektif, efisien dan akurat sehingga dapat bermanfaat untuk kebutuhan medis.

### 7.1 Kesimpulan

Berdasarkan hasil dari beberapa tahap penelien yaitu perancangan, implementasi pengujian dan analisis hasil pengujian, penelitian ini dapat diambil beberapa kesimpulan sebagai berikut:

1. Sensor photodiode dan nodemcu yang dikemas pada kotak hitam untuk mencegah adanya gangguan dari cahaya dari luar dan meletakkan kotak sensor node pada chamber infus sebagai sensor node. Antara photodiode dan led dipasang berhadap-hadapan untuk mendeteksi adanya tetesan infus. Tingkat akurasi pembacaan tetesan infus masing-masing sensor adalah sebesar 96%, 96%, dan 94.6%.
2. Pada penelitian ini telah mengimplementasikan metode MQTT sebagai metode pengiriman data ke server dengan ruang lingkup jaringan lokal. Selain menerapkan protocol MQTT dalam pengiriman, juga diterapkan *Websocket* untuk mendapatkan hasil monitoring secara realtime. Dari pengujian yang telah dilakukan, metode ini memiliki tingkat keberhasilan pengiriman data yang tinggi. Untuk pengujian *delay* yang telah dilakukan, diperoleh hasil rata-rata *delay* pengiriman adalah sebesar 0.1 detik. Hal tersebut dikarenakan, lingkup jaringan yang digunakan adalah jaringan lokal sehingga *traffic* data yang terjadi tidak besar dan tidak saling tunggu. Selain itu, implementasi protokol MQTT juga mempengaruhi besar kecilnya delay.
3. Antarmuka monitoring menggunakan web yang disajikan dalam bentuk table dan diagram yang mengindikasikan ketersediaan cairan infus pada ruang pasien. Data yang diperoleh dari server dapat ditampilkan secara *realtime*. Namun, untuk menampilkan data monitoring pada diagram, memiliki delay sebesar 1 detik dari data sebenarnya. Hal tersebut dikarenakan, pada *library* yang digunakan memerlukan waktu untuk *me-render* data dalam bentuk diagram.

### 7.2 Saran

Pada penelitian ini, peneliti mempunyai beberapa saran yang dapat dijadikan acuan untuk pengembangan sistem serupa selanjutnya antara lain sebagai berikut:

1. Menggunakan metode pembacaan tetesan infus dengan menggunakan sensor atau media lain yang memiliki tingkat akurasi pembacaan tetesan dengan baik.

2. Menerapkan sistem pengontrol kecepatan infus, sehingga kecepatan tetesan infus dapat diatur secara langsung pada komputer petugas.
3. Menerapkan sistem monitoring yang dapat diakses pada perangkat bergerak seperti telepon gengam, sehingga monitoring dapat dilakukan diberbagai tempat yang terjangkau jaringan.



## DAFTAR PUSTAKA

- Budioko, T., 2016. Sistem Monitoring Suhu Jarak Jauh Berbasis Intenet of Things Menggunakan Protokol MQTT. pp. 1-9.
- Hardianto, F., 2015. Aplikasi Groupchat di Android Menggunakan *Websocket*.
- James, M., 2015. *Programming Electronics Academy*. [Online]  
Available at: <https://programmingelectronics.com/tutorial-3-arduino-ide-and-sketch-overview/> [Diakses Februari 2018].
- Fathkurrahman. 2009. Diduga akibat Perawat Lalai, Bayi 4 Hari Tewas. [Online].  
<https://news.okezone.com/read/2009/09/24/340/259679/diduga-akibat-perawat-lalai-bayi-4-hari-tewas>. [Diakses Januari 2018]
- MQTT.org, 2015. *Getting Started With MQTT*. [Online] Available at :  
<http://mqtt.org/> [Diakses Januaryi 2018].
- Natalina, D., 2016. Alat Monitoring Infus Set pada Pasien Rawat Inap. Volume 4, pp. 1-15.
- Rochman, H. A., 2017. Sistem Kendali Berbasis Mikrokontroler Menggunakan Protokol MQTT.
- Syahrul, 2010. Sistem Pemantauan Pasien Secara Terpusat.
- Sinauarduino. 2016. Mengenal Arduino Software (IDE). Diakses Januari 3, 2018.  
<http://www.sinauarduino.com/artikel/mengenal-arduino-software-ide/>.
- Samir, 2016. *MQTT over WebSockets with HiveMQ*. [Online] Tersedia di  
<https://www.hivemq.com/blog/mqtt-over-websockets-with-hivemq>.  
[Diakses Februari 2018]
- Pelayo, Roland. 2017. *Intro to NodeMCU and Arduino IDE*. [Online] Tersedia di  
<https://www.teachmemicro.com/intro-nodemcu-arduino/>. [Diakses Februari 2018]
- Kelleher, Fionn. 2014. *Understanding Socket.io*. [Online] Tersedia di  
<https://nodesource.com/blog/understanding-socketio/>. [Diakses Februari 2018]