



TUGAS AKHIR - EB184803

***MULTI-PARAMETER MONITORING MENGGUNAKAN JARINGAN
SENSOR NIRKABEL UNTUK APLIKASI MONITORING DAN DATABASE
MEDIS***

**Beryl Abdiel Sahata Ebenezer Sitorus
0731154000004**

**DOSEN PEMBIMBING:
Dr. Rachmad Setiawan, S.T., M.T.
Nada Fitriyatul Hikmah, S.T., M.T.**

**PROGRAM SARJANA
DEPARTEMEN TEKNIK BIOMEDIK
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2020**



TUGAS AKHIR - EB184803

**MULTI-PARAMETER MONITORING USING WIRELESS SENSOR
NETWORKS FOR MEDICAL MONITORING APPLICATION AND
MEDICAL DATABASE**

**Beryl Abdiel Sahata Ebenezer Sitorus
0731154000004**

**DOSEN PEMBIMBING:
Dr. Rachmad Setiawan, S.T., M.T.
Nada Fitriyatul Hikmah, S.T., M.T.**

**UNDERGRADUATE PROGRAM
BIOMEDICAL ENGINEERING DEPARTMENT
FACULTY OF INTELLIGENT ELECTRICAL AND INFORMATICS
TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2020**

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa sebagian maupun keseluruhan isi tugas akhir saya yang berjudul "**Multi-Parameter Monitoring Menggunakan Jaringan Sensor Nirkabel Untuk Aplikasi Monitoring dan Database Medis**" merupakan hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya mandiri. Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apa bila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Januari 2020



Beryl Abdiel SE Sitorus

NRP. 07311540000004

Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Teknik (ST)
di
Institut Teknologi Sepuluh Nopember


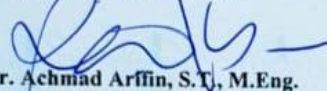
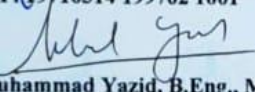
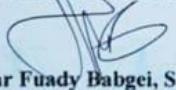
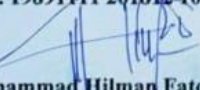
Oleh
Beryl Abdiel Sahata Ebenezer Sitorus
NRP. 07311540000004

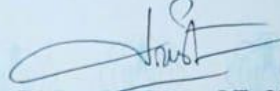
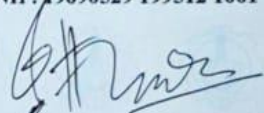
Tanggal Ujian : 06 Januari 2020
Periode Wisuda : Maret 2020

Disetujui oleh:

Dosen Penguji :

Dosen Pembimbing :

- 
1. Dr. Tri Arief Sardjono, S.T., M.T.
NIP. 19700212 199512 1001
 2. Dr. Achmad Arffin, S.T., M.Eng.
NIP. 19710314 199702 1001
 3. Muhammad Yazid, B.Eng., M.Eng.
NIP. 19800420 201504 1001
 4. Atar Fuady Babgei, S.T., M.Sc.
NIP. 19891111 201812 1001
 5. Muhammad Hilman Fatoni, S.T., M.T.
NIP. 19910325 201504 1001

- 
1. Dr. Rachmad Setiawan, S.T., M.T.
NIP. 19690529 199512 1001
 2. Nada Fitriyatul Hikmah, S.T., M.T.
NIP. 19900107 201803 2001



Departemen Teknik Biomedik,
Dr. Achmad Arffin, S.T., M.Eng.
NIP. 19710314 199702 1001

Multi-Parameter Monitoring Menggunakan Jaringan Sensor Nirkabel untuk Aplikasi Monitoring dan Database Medis

Nama : Beryl Abdiel SE Sitorus
Dosen Pembimbing : 1. Dr. Rachmad Setiawan, S.T., M.T.
2. Nada Fitriyatul Hikmah, S.T., M.T.

ABSTRAK

Teknologi dalam bidang kesehatan dari tahun ke tahun semakin berkembang. Termasuk teknologi monitoring. Teknologi monitoring pada bidang kesehatan dibuat untuk memonitor *vital sign* manusia yang paling penting untuk di monitor. Teknologi pada bidang komunikasi juga sudah berkembang dari yang menggunakan kabel, sekarang sudah banyak perangkat yang berkomunikasi secara *wireless*. Untuk itu, dalam tugas akhir ini diusulkan sebuah alat yang dapat melakukan monitoring dan mengirimkan data tersebut secara *wireless*. Alat ini menghasilkan sebuah *healthcare monitoring system* yang dapat digunakan dalam ruang ICU maupun ruang rawat inap biasa. Alat tersebut terdiri dari tiga sensor yang memberi informasi berupa *infusion weight scale* (HX711), *temperature* (DS18B20) dan *pulse sensor* (AK90), semua informasi yang dihasilkan oleh setiap sensor akan dikirim melalui mikrokontroler. Pengiriman secara *wireless* pada alat ini dilakukan dengan menggunakan komponen *wireless module* yang terhubung dengan mikrokontroler. Data dari mikrokontroler tersebut akan diterima dan ditampilkan di aplikasi berbasis web dan disimpan pada database. Berdasarkan hasil penelitian, akurasi dari penerimaan data secara *wireless* yang dikirimkan oleh sensor memiliki *success rate* sebesar 100% setiap data berhasil ditambahkan ke dalam database. Selain itu, berdasarkan hasil penelitian dapat disimpulkan setiap *perintah add, edit, update* dan *delete* pada aplikasi akan langsung merubah isi data dari tabel database yang ada di aplikasi. Aplikasi yang dibangun pada tugas akhir ini juga dapat membuat *report* harian secara otomatis dalam bentuk *file pdf* yang di dalamnya terdapat data dari setiap sensor *monitoring*.

Kata Kunci: *Vital Sign, Wireless Sensor Networks, Database*

Multi-Parameter Monitoring Using Wireless Sensor Networks for Medical Monitoring Application and Medical Database

Name : Beryl Abdiel SE Sitorus
Lecturer : 1. Dr. Rachmad Setiawan, S.T., M.T.
2. Nada Fitriyatul Hikmah, S.T., M.T.

ABSTRACT

Technology in the health sector from year to year is increasingly developing. Including monitoring technology. Monitoring technology in the health sector was created to monitor the most important human sign to be monitored. Technology in the field of communication has also developed from using cables to wireless, now many devices are communicating wirelessly. For this reason, in this thesis a tool that can monitor and transmit data wirelessly is proposed. This tool produces a healthcare monitoring system that can be used in the ICU or inpatient rooms. The tool consists of three sensors that provide information in the form of infusion weight scale (HX711), temperature (DS18B20) and pulse sensor (AK90), all information generated by each sensor will be sent via the microcontroller. Wireless delivery on this device is done by using a wireless module component that is connected to the microcontroller. Data from the microcontroller will be received and displayed in a web-based application and stored in a database. Based on the results of the study, the accuracy of the wireless data reception sent by the sensor has a success rate of 100% each data successfully added to the database. In addition, based on the results of the study it can be concluded that each command add, edit, update and delete the application will immediately change the contents of the data from the database tables in the application. The application that was built in this final project can also create a daily report automatically in the form of a pdf file that contain data from each monitoring sensor.

Keywords: Vital Sign, Wireless Sensor Networks, database

KATA PENGANTAR

Penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa karena hanya atas kasih sayang-Nya penulis mampu menyelesaikan tugas akhir ini dengan lancar dan dapat selesai tepat pada waktunya. Penulis juga mengucapkan terima kasih atas bantuan dari berbagai pihak, maka penulis mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa yang telah memberikan anugerah-Nya dan selalu melancarkan pengerjaan Tugas Akhir selama satu semester ini.
2. Dr. Rachmad Setiawan, S.T., M.T. selaku dosen pembimbing satu yang selalu sabar dalam berdiskusi dan memberikan kritik serta saran.
3. Nada Fitriyatul Hikmah, S.T., M.T. selaku dosen pembimbing dua yang selalu sabar dalam berdiskusi dan memberikan kritik serta saran.
4. Dr. Achmad Arifin, S.T., M.Eng. selaku kepala departemen yang telah memberi masukan, serta bimbingan selama pengerjaan tugas akhir terlebih selama perkuliahan di ITS.
5. Prof. Dr. Ir. Mohammad Nuh, DEA. selaku dosen wali penulis selama 8 semester ini.
6. Rekan-rekan Teknik Biomedik angkatan 2015 yang sangat banyak memberi dukungan.
7. Bapak Ibu dosen pengajar Teknik Biomedik ITS.
8. Serta seluruh pihak yang tidak dapat disebutkan yang telah membantu penulis hingga Tugas Akhir ini bisa diselesaikan dengan semaksimal mungkin.

Terlepas dari semua itu, penulis sadar bahwa masih terdapat kekurangan pada berbagai aspek dan masih banyak hal yang perlu diperbaiki. Oleh karena itu, penulis dengan terbuka menerima kritik dan saran dari pembaca agar dapat memperbaiki tugas akhir ini. Semoga penelitian ini dapat memberikan manfaat untuk banyak orang dan dapat dikembangkan lebih baik lagi pada penelitian selanjutnya.

Surabaya, Januari 2020

Penulis

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR	Error! Bookmark not defined.
LEMBAR PENGESAHAN	Error! Bookmark not defined.
ABSTRAK	v
ABSTRACT	vi
KATA PENGANTAR.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan dan Manfaat	3
1.5 Kontribusi Penelitian	4
BAB 2 TINJAUAN PUSTAKA.....	5
2.1 <i>Wireless Sensor Networks</i>	5
2.1.1 Topologi Jaringan Star	5
2.1.2 Karakteristik Jaringan <i>Star</i>	6
2.1.3 Kelebihan Topologi <i>Star</i>	6
2.1.4 Kekurangan Topologi <i>Star</i>	6
2.1.5 Komunikasi Jaringan <i>Wireless</i>	6
2.2 Protokol Komunikasi	7
2.2.1 <i>Open Systems Interconnect (OSI) Layer</i>	7
2.2.2 Cara Kerja <i>OSI Layer</i>	10
2.2.3 Desain Protokol Komunikasi.....	11
2.2.4 <i>Data Structure</i>	12
2.3 <i>Multi-Parameter Monitoring</i>	13
2.4 <i>Sensor Monitoring</i>	13
2.4.1 <i>Heart Rate Sensor</i>	14
2.4.2 <i>Sensor Infus (Weight Scale)</i>	14
2.4.3 <i>Sensor Temperature</i>	15
2.5 Mikrokontroler	16

2.5.1 Arduino Mega 2560 With Wifi Built-in ESP8266	16
2.6 Database	17
2.7 Rasional	18
BAB 3 PERANCANGAN SISTEM	20
3.1 Gambaran Umum Sistem	20
3.2 Pengguna Sistem	22
3.3 Perancangan <i>Use Case</i>	22
3.4 Perancangan <i>Hardware</i>	23
3.5 Perancangan <i>Software</i>	25
3.5.1 Protokol Komunikasi	26
3.5.2 <i>Pulse sensor Monitoring</i>	27
3.5.3 <i>Weight Scale Infusion</i>	28
3.5.4 <i>Temperature</i>	28
3.6 Rancangan <i>Interface</i>	28
3.6.1 Rancangan <i>Interface Login</i>	29
3.6.2 Rancangan <i>Interface Dashboard</i>	30
3.6.3 Rancangan <i>Interface User</i>	30
3.6.4 Rancangan <i>Interface Petugas</i>	30
3.6.5 Rancangan <i>Interface Ruangan</i>	30
3.6.6 Rancangan <i>Interface Tugas</i>	33
3.6.7 Rancangan <i>Interface Alat</i>	33
3.6.8 Rancangan <i>Interface Laporan</i>	33
3.7 Aspek <i>Real-Time Sensor</i>	34
3.7.1 Akuisisi Data <i>Temperature Sensor</i>	34
3.7.2 Akuisisi Data <i>Pulse Sensor</i>	35
3.7.3 Akuisisi Data <i>Weight Scale Sensor</i>	36
3.7.4 Akuisisi Data pada Aplikasi	37
3.8 Perancangan <i>Database</i>	38
3.8 Pengujian	42
BAB 4 PENGUJIAN SISTEM	43
4.1 Pengujian <i>Black Box</i>	43
4.1.1 <i>Use Case Login</i>	43
4.1.2 <i>Use Case User</i>	47
4.1.3 <i>Use Case Ruangan</i>	48

4.1.4	<i>Use Case</i> Petugas.....	49
4.1.5	<i>Use Case</i> Tugas	51
4.1.6	<i>Use Case</i> Alat.....	52
4.2	Pengujian Laporan Otomatis	54
4.3	Pengujian <i>Relation Database</i>	56
4.3.1	<i>Relation Database</i> Tugas	57
4.3.2	<i>Relation Database</i> Alat.....	58
4.4	Pengujian Komunikasi Sistem.....	59
BAB 5 PEMBAHASAN.....		62
5.1	Pengujian <i>Black Box</i>	62
5.2	Pengujian Laporan Otomatis	63
5.3	Pengujian <i>Relation Database</i>	64
5.4	Pengujian Komunikasi Sistem.....	65
BAB 6 KESIMPULAN DAN SARAN.....		66
6.1	Kesimpulan.....	66
6.2	Saran	66
DAFTAR PUSTAKA		67
BIOGRAFI PENULIS		68

DAFTAR GAMBAR

Gambar 2.1	Topologi Jaringan Star	5
Gambar 2.2	Modul ESP8266	7
Gambar 2.3	OSI Model	8
Gambar 2.4	Protokol Komunikasi.....	12
Gambar 2.5	Modul AK90	13
Gambar 2.6	Modul <i>Loadcell</i>	14
Gambar 2.7	Modul HX711	14
Gambar 2.8	Modul DS18B20	15
Gambar 2.9	Arduino MEGA 2560 <i>with</i> ESP8266.....	16
Gambar 2.10	<i>Fishbone Diagram</i>	19
Gambar 3.1	Gambaran Umum Sistem.....	20
Gambar 3.2	<i>Use Case</i> Sistem Aplikasi	21
Gambar 3.3	<i>Base Node</i>	24
Gambar 3.4	<i>Child Node</i>	24
Gambar 3.5	Rancangan Komunikasi.....	26
Gambar 3.6	Metode <i>Weight Scale</i>	28
Gambar 3.7	Rancangan <i>Login</i>	29
Gambar 3.8	Rancangan <i>Dashboard</i>	31
Gambar 3.9	Rancangan <i>Interface User</i>	31
Gambar 3.10	Rancangan <i>Interface</i> Petugas	32
Gambar 3.11	Rancangan <i>Interface</i> Ruangan.....	33
Gambar 3.12	Rancangan <i>Interface</i> Tugas	33
Gambar 3.13	Rancangan <i>Interface</i> Alat	34
Gambar 3.14	Rancangan <i>Interface</i> Laporan.....	34
Gambar 3.15	Akuisisi Data <i>Real-Time</i> Sensor DS18B20.....	35
Gambar 3.16	Akuisisi Data <i>Real-Time</i> Sensor <i>Pulse</i> (AK-90).....	36

Gambar 3.17	Akuisisi Data <i>Real-Time Weight Scale</i> (HX711).....	36
Gambar 3.18	Rancangan Pengiriman Dari <i>Client</i>	37
Gambar 3.19	Relasi Antar Tabel.....	41
Gambar 4.1	Halaman <i>Login</i>	44
Gambar 4.2	<i>Login Admin</i>	45
Gambar 4.3	<i>Menu Dashboard</i>	45
Gambar 4.4	<i>Login Petugas</i>	46
Gambar 4.5	<i>Menu Dashboard</i>	46
Gambar 4.6	<i>Database Awal User</i>	47
Gambar 4.7	<i>Input Data User</i>	47
Gambar 4.8	<i>Database User Terbaru</i>	48
Gambar 4.9	<i>Database Ruangan Awal</i>	48
Gambar 4.10	<i>Input Data Ruangan</i>	49
Gambar 4.11	<i>Database Ruangan terbaru</i>	49
Gambar 4.12	<i>Database Petugas Awal</i>	50
Gambar 4.13	<i>Form Input Petugas</i>	50
Gambar 4.14	<i>Update Database Petugas</i>	50
Gambar 4.15	<i>Database Tugas Awal</i>	51
Gambar 4.16	<i>Form Input Tugas</i>	51
Gambar 4.17	<i>Database Tugas Terbaru</i>	52
Gambar 4.18	<i>Database Alat Awal</i>	53
Gambar 4.19	<i>Form Tambah Alat</i>	53
Gambar 4.20	<i>Database Alat Terbaru</i>	53
Gambar 4.21	<i>Form Report Alat</i>	54
Gambar 4.22	Laporan.	55
Gambar 4.23	Data Sensor.....	55
Gambar 4.24	<i>Relation Database</i>	56

Gambar 4.25	<i>Form Database Tugas</i>	57
Gambar 4.26	<i>Database Ruangan</i>	57
Gambar 4.27	<i>Database Petugas</i>	58
Gambar 4.28	<i>Form Tambah Alat</i>	58
Gambar 4.29	<i>Database Ruangan</i>	59
Gambar 4.30	<i>Serial Monitor Arduino</i>	59
Gambar 4.31	<i>Inisiasi Komunikasi Client-Server</i>	60
Gambar 4.32	<i>Updated Database</i>	60
Gambar 4.33	<i>Dashboard Aplikasi</i>	61

DAFTAR TABEL

Tabel 3.1 Rancangan Tabel tbl_user.....	39
Tabel 3.2 Rancangan Tabel tbl_petugas.....	39
Tabel 3.3 Rancangan Tabel tbl_ruangan.....	39
Tabel 3.4 Rancangan Tabel tbl_alat.....	40
Tabel 3.5 Rancangan Tabel tbl_tugas.....	40
Tabel 3.6 Rancangan Tabel tbl_histori.....	40
Tabel 5.1 <i>Use Case Login</i>	62
Tabel 5.2 <i>Use Case User</i>	62
Tabel 5.3 <i>Use Case Ruangan</i>	62
Tabel 5.4 <i>Use Case Petugas</i>	63
Tabel 5.5 <i>Use Case Tugas</i>	63
Tabel 5.6 <i>Use Case Alat</i>	63
Tabel 5.7 Laporan Otomatis.....	64
Tabel 5.8 <i>Relation Database</i>	65
Tabel 5.9 Komunikasi Sistem.....	65

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Teknologi dari tahun ke tahun semakin berkembang. Salah satunya adalah teknologi dalam bidang kesehatan, banyak sekali teknologi yang sudah berkembang dalam bidang kesehatan, salah satunya adalah teknologi monitoring pada pasien yang biasanya digunakan untuk mengetahui kondisi kesehatan pasien. teknologi monitoring pada bidang kesehatan tertuju pada bagian *vital sign* manusia yang paling penting untuk dimonitor [6].

Vital sign adalah ukuran statistik berbagai fisiologis yang digunakan untuk membantu menentukan status kesehatan seseorang, terutama pada pasien yang secara medis tidak stabil atau memiliki faktor-faktor resiko komplikasi kardiopulmonal dan untuk menilai respon terhadap intervensi. Terdapat 8 *vital sign* yang penting untuk dimonitoring yaitu temperatur tubuh, *pulse*, tekanan darah, SpO₂, *urine, pain*, tingkat pernapasan, *level of consciousness* [1].

Dari beberapa *vital sign* tersebut dibuatlah sebuah alat monitoring yang berfungsi untuk memantau setiap keadaan dari *vital sign* yang ada pada manusia, karena dengan melakukan *vital sign monitoring* manusia, perawat atau dokter dapat mengetahui kondisi pasien setiap saat dan jika ada perubahan kondisi pasien, dengan terpasangnya *vital sign monitoring* pada pasien tersebut perawat atau dokter dapat langsung mengambil tindakan. Maka dari itu *vital sign monitoring* sangatlah penting dalam dunia *healthcare*, lebih tepatnya dalam perkembangan *monitoring healthcare* banyak *researcher* yang tertarik untuk mengembangkannya [3].

Dikarenakan mempertahankan suhu tubuh yang optimal sangat penting untuk kehidupan, mengambil dan mencatat suhu pasien yang sedang dirawat inap adalah hal yang mendasar di dalam kegiatan rumah sakit, maka dari itu *temperature* tubuh merupakan *vital sign* yang penting untuk di-*monitoring*. Dengan adanya *monitoring* suhu tubuh dapat menjadi indikator klinis penting dari perubahan status fisiologis yang memerlukan penyelidikan lebih lanjut. Teknik yang benar dan pencatatan yang cermat penting untuk akurasi karena dokter mengandalkan pengamatan klinis untuk membuat diagnosis.

Heart rate atau yang disebut dengan denyut jantung adalah *predictor* dari penyakit yang berhubungan dengan kardiovaskular, maka dari itu *heart rate monitoring* pada pasien sangatlah penting karena bukan hanya bisa mengetahui kondisi jantung secara *continuous* tapi kita juga dapat mengetahui kondisi jantung seorang pasien dan memprediksi penyakit lebih awal.

Infusion atau cairan infus merupakan suatu cairan yang difungsikan untuk menggantikan cairan di dalam tubuh, pengganti nutrisi maka dari itu cairan infus sangatlah penting dalam keberlangsungan pasien. Dikarenakan cairan infus dapat habis dan pergantian cairan infus sangat sering dilakukan di rumah sakit, banyak masalah yang muncul dari hal ini yaitu keterlambatan pergantian cairan infus yang

habis dapat menyebabkan darah dalam tubuh tertarik keluar melewati selang infus maka dari itu dibutuhkan pemecahan masalah dengan menggunakan *monitoring healthcare* pada cairan infus.

Healthcare selalu menjadi perhatian besar bagi dunia, karena menyangkut kualitas hidup yang dapat dimiliki seorang individu. Selalu lebih baik untuk mencegah penyakit daripada mengobatinya, sehingga dibutuhkanlah *monitoring kesehatan* secara berkala, pengambilan data *Healthcare monitoring* dilakukan secara berkala. maka dari itu perkembangan *healthcare monitoring* yang awalnya masih menggunakan kabel, sekarang sudah mulai dikembangkan untuk menggunakan nirkabel, dari hal tersebut digunakanlah *wireless sensor networks* untuk menghasilkan *healthcare monitoring* nirkabel [5].

Perkembangan teknologi komunikasi pada tahun ke tahun sudah mulai menuju pada *internet of things*, yang dimaksud *internet of things* adalah sebuah konsep dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer. IoT telah berkembang dari konvergensi teknologi nirkabel, *micro-electromechanical systems* (MEMS), dan Internet.

“A Things” pada *Internet of Things* dapat didefinisikan sebagai subjek misalkan sebuah sensor *monitoring* yang melakukan *monitoring* dengan otomatis. Sejauh ini, IoT paling erat hubungannya dengan komunikasi *machine-to-machine* (M2M) di bidang manufaktur dan listrik, perminyakan, dan gas. Produk dibangun dengan kemampuan komunikasi M2M yang sering disebut dengan sistem cerdas atau “*smart*”.

Wireless sensor networks terdiri dari banyak perangkat otonom yang terdistribusi secara spasial, maka dari itu *wireless sensor networks* disebut dengan *smart sensor nodes* yang dapat melakukan *monitoring* lingkungan atau fisik di lokasi yang berbeda. *Wireless sensor networks smart node* memiliki kemampuan komunikasi, modul pemrosesan dengan memori, memiliki fungsi *sensing* dan modul daya yang membuat dapat beroperasi secara otonom. Maka dari itu *wireless sensor networks* sangat tepat untuk digunakan pada *healthcare application* [3].

Pada saat melakukan *monitoring* terdapat beberapa metode yang dapat dilakukan, salah satunya dengan menggunakan teknologi WSN (*Wireless Sensor Network*). Teknologi *Wireless Sensor Network* adalah suatu sistem *embedded* yang di dalamnya terdapat satu atau lebih sensor dan dilengkapi dengan peralatan sistem komunikasi. Sensor di sini digunakan untuk menangkap informasi sesuai dengan karakteristik [4].

Berdasarkan perkembangan *monitoring* dan *healthcare*, perkembangan sistem *database* juga sudah mulai digunakan dalam *healthcare*, penggunaan *database* ditujukan untuk menyimpan data pasien yang dapat diakses secara mudah oleh pihak rumah sakit, termasuk data *monitoring*. Tidak hanya itu saja *database* juga dapat menyimpan data rekam medis pasien. Dikarenakan semua

perkembangan sudah menuju pada *system online, healthcare* juga sudah mulai berkembang di dalam penyimpanan online yaitu menggunakan *database* sistem untuk menyimpan semua data pasien dari histori kesehatan pasien, registrasi pasien, sampai histori obat yang digunakan oleh pasien disimpan di database dan akhirnya berkembang pada database untuk data *vital sign monitoring*. Data *vital sign* disimpan dalam *database* yang nantinya dapat digunakan untuk menjadi referensi kesehatan pasien yang sedang dirawat intensif di rumah sakit [5].

Berdasarkan penelitian-penelitian sebelumnya, hasil akhir yang diharapkan adalah dapat membuat sebuah sistem *monitoring* yang memiliki beberapa sensor *vital sign* yang sudah terintegrasi dan memiliki komunikasi secara *wireless*, yang nantinya data tersebut dapat ditampilkan pada *display monitor*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah dipaparkan dapat diperoleh rumusan masalah tentang bagaimana perancangan *wireless sensor network* yang digunakan untuk melakukan *vital sign monitoring* dan *infusion indicator*. *Wireless sensor networks* yang dirancang haruslah dapat berkomunikasi secara *wireless* dengan aplikasi medis

Rumusan masalah selanjutnya adalah bagaimana perancangan multi-parameter *monitoring* pada jaringan sensor nirkabel agar data dari sensor dapat diterima oleh aplikasi medis dengan baik dan tidak ada data yang hilang pada saat proses pengiriman data secara *wireless* berlangsung.

Rumusan masalah selanjutnya adalah bagaimana perancangan *database* dalam aplikasi medis yang dapat berfungsi sebagai *management* ruang dan *management* koneksi sensor dengan aplikasi medis.

1.3 Batasan Masalah

Batasan masalah dari penelitian ini adalah banyaknya jumlah sensor yang dijadikan *wireless sensor network* hanya berjumlah 3 sensor yaitu modul *weight scale sensor*, modul *temperature sensor*, modul *pulse sensor*. Batasan *security* pada protokol komunikasi hanya sebatas perlindungan wifi menggunakan *password*. Batasan *database* yang digunakan adalah *database* yang berhubungan dengan *management* koneksi pada aplikasi dan *management* pengguna dalam aplikasi.

1.4 Tujuan dan Manfaat

Berdasarkan latar belakang dan perumusan masalah yang telah dijelaskan maka hasil akhir dari penelitian ini adalah sebuah produk *multiple vital sign monitoring* yang terdiri dari *integrated sensor*, *wireless module* dan *software* dengan *database*. Rincian tujuan dan manfaat penelitian akan dijabarkan di bawah ini.

Tujuan dan manfaat dari tugas akhir ini adalah menghasilkan sebuah alat yang dapat melakukan *multiple vital sign monitoring* dan *infusion indicator* pada pasien dan mengirim data monitoring tersebut secara *wireless* yang nantinya terhubung dengan aplikasi medis yang di dalamnya terdapat database management ruang. Dalam mencapai *multiple sign monitoring* membutuhkan sebuah mikrokontroler yang dapat menerima dan mengolah data sensor seterusnya mengirimkan semua data tersebut secara *wireless* dan *real time*.

Manfaat yang dapat dihasilkan oleh penelitian ini adalah dapat mengetahui cara membuat sebuah *integrated sensor* menjadi *wireless sensor network*.

Manfaat berikutnya yang didapat adalah terciptanya sebuah *hardware* dengan *multiple vital sign sensor* dan *software* yang dihubungkan yang berfungsi untuk menampilkan data yang diterima oleh sensor dan aplikasi yang memiliki *database management* ruang.

1.5 Kontribusi Penelitian

terdapat dua aspek kontribusi dari penelitian ini yaitu aspek ilmiah dan aspek *practical*. Dalam aspek ilmiah, kontribusi yang diberikan adalah menjadi suatu pengembangan untuk penelitian dalam bidang biomedik. *Vital sign* adalah tanda *vital* pada manusia yang penting untuk dilakukan *monitoring* dengan adanya penelitian ini, *vital sign monitoring* secara *multiple* dalam satu alat yang terintegrasi dapat terealisasi.

Dalam aspek *practical*, kontribusinya adalah *multiple vital sign monitoring* mempunyai banyak parameter sensor yang dapat melakukan *monitoring* secara *real time*. Parameter-parameter tersebut dapat membantu dokter dalam melakukan *monitoring* pasien yang sedang di dalam perawatan intensif.

BAB 2 TINJAUAN PUSTAKA

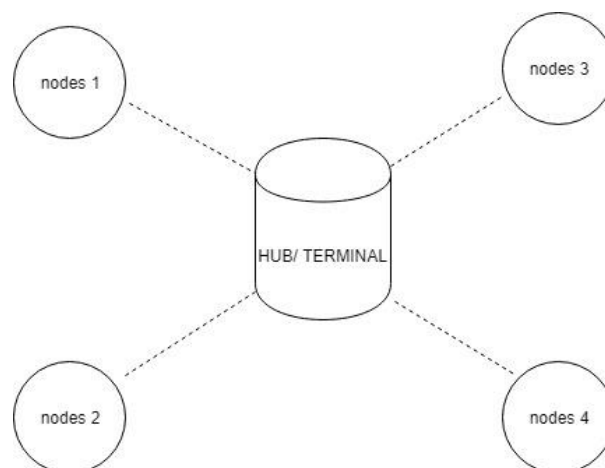
2.1 *Wireless Sensor Networks*

Wireless sensor network merupakan sebuah jaringan yang menghubungkan perangkat-perangkat seperti *sensor node*, *router* dan *sink node*. Perangkat ini terhubung secara ad-hoc dan mendukung komunikasi multi-hop. Istilah ad-hoc merujuk pada kemampuan perangkat untuk berkomunikasi satu sama lain secara langsung tanpa memerlukan infrastruktur jaringan seperti *router* atau akses *point*. Sedangkan *multi-hop* yaitu istilah yang merujuk pada komunikasi beberapa perangkat yang melibatkan perangkat antara (*intermediate*), *multi-hop* melibatkan perangkat seperti *router* untuk meneruskan sebuah paket dari satu *node* menuju ke *node* lain dalam jaringan [7].

2.1.1 Topologi Jaringan Star

Jaringan komunikasi yang menghubungkan antara *nodes* memiliki banyak model topologi yang dapat diimplementasikan, topologi jaringan yang diimplementasikan di dalam *wireless sensor network* ini adalah topologi jaringan star.

Topologi star merupakan bentuk topologi jaringan yang berupa konvergensi dari *node* tengah ke setiap *node* atau pengguna. Masing-masing *workstation* dihubungkan secara langsung ke Server atau *Hub/Switch*. Intinya topologi ini menggunakan *Hub/Switch* untuk menghubungkan dari komputer satu ke komputer yang lain. *Hub/ Switch* berfungsi untuk menerima sinyal-sinyal dari komputer dan meneruskan ke semua komputer yang terhubung dengan *Hub/Switch* tersebut



Gambar 2.1. Topologi Jaringan Star

2.1.2 Karakteristik Jaringan *Star*

Terdapat beberapa karakteristik pada topologi *star* yaitu:

1. Setiap *Node* berkomunikasi secara langsung dengan *central node*. *Traffic* data mengalir dari *node* ke *central node* dan kembali lagi.
2. Mudah dikembangkan karena setiap *node* hanya memiliki kabel yang langsung terhubung ke *central node*
3. Jika terjadi kerusakan pada salah satu *node* maka hanya pada *node* tersebut yang terganggu tanpa mengganggu jaringan lain
4. Dapat digunakan Kabel *Lower* karena hanya meng-*handle* satu trafik *node* dan biasanya menggunakan kabel UTP.

2.1.3 Kelebihan Topologi *Star*

Berikut kelebihan-kelebihan yang ada pada topologi *star*:

1. Dapat digunakan Kabel *Lower* karena hanya meng-*handle* satu trafik *node* dan biasanya menggunakan kabel UTP.
2. Tingkat keamanan termasuk tinggi
3. Tahan terhadap lalu lintas jaringan yang sibuk
4. Penambahan dan pengurangan *station* dapat dilakukan dengan mudah.
5. Akses Kontrol terpusat.
6. Kemudahan deteksi dan isolasi kesalahan/kerusakan pengelolaan jaringan
7. Paling fleksibel

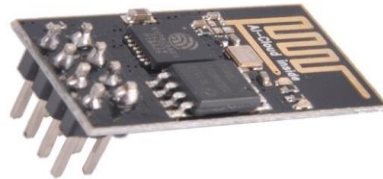
2.1.4 Kekurangan Topologi *Star*

Terdapat beberapa kekurangan yang ada pada topologi *star* yaitu:

1. Jika *node* tengah mengalami kerusakan, maka seluruh rangkaian akan berhenti
2. Boros dalam pemakaian kabel
3. Hub jadi elemen kritis karena kontrol terpusat
4. Peran hub sangat sensitif sehingga ketika terdapat masalah dengan hub maka jaringan tersebut akan *down*
5. Jaringan tergantung pada terminal pusat
6. Jika menggunakan *switch* dan lalu lintas data padat dapat menyebabkan jaringan lambat
7. Biaya jaringan lebih mahal dari pada *bus* atau *ring*

2.1.5 Komunikasi Jaringan *Wireless*

Dalam sebuah komunikasi berbentuk *wireless* pada dasarnya haruslah ada komunikasi antara *nodes* dengan terminal, agar dapat melakukan komunikasi secara *wireless* dibutuhkan modul yang dapat membuat sebuah *IP address* agar dapat



Gambar 2.2. Modul ESP8266

terkoneksi secara nirkabel dengan aplikasi medis, modul komunikasi yang digunakan merupakan wifi modul ESP8266.

ESP8266 merupakan modul wifi yang berfungsi sebagai perangkat tambahan mikrokontroler Arduino agar dapat terhubung langsung dengan wifi dan membuat koneksi *TCP/IP*.

Modul pada Gambar 2.2 membutuhkan daya sekitar 3.3v yang memiliki tiga mode wifi yaitu *Station*, *Access Point* dan *Both* (Keduanya).

Wireless module atau biasa disebut wifi banyak digunakan pada perangkat yang membutuhkan konektivitas nirkabel penggunaan *wireless* modul ESP8266 digunakan karena dapat membuat koneksi *TCP/IP*, dengan sistem koneksi ini maka setiap mikrokontroler dapat memiliki *IP address* yang berbeda-beda dengan mikrokontroler lainnya.

2.2 Protokol Komunikasi

Protokol komunikasi merupakan peraturan atau prosedur untuk mengirimkan sebuah data pada perangkat elektronik. Pada sistem komputer, setiap komputer berkomunikasi dengan komputer lainnya menggunakan sebuah protokol. Standar komunikasi yang diterapkan di dalam jaringan komputer disebut *Open Systems Interconnect (OSI) layer*. Standar itulah yang menyebabkan seluruh alat komunikasi dapat saling berkomunikasi melalui jaringan.

2.2.1 *Open Systems Interconnect (OSI) Layer*

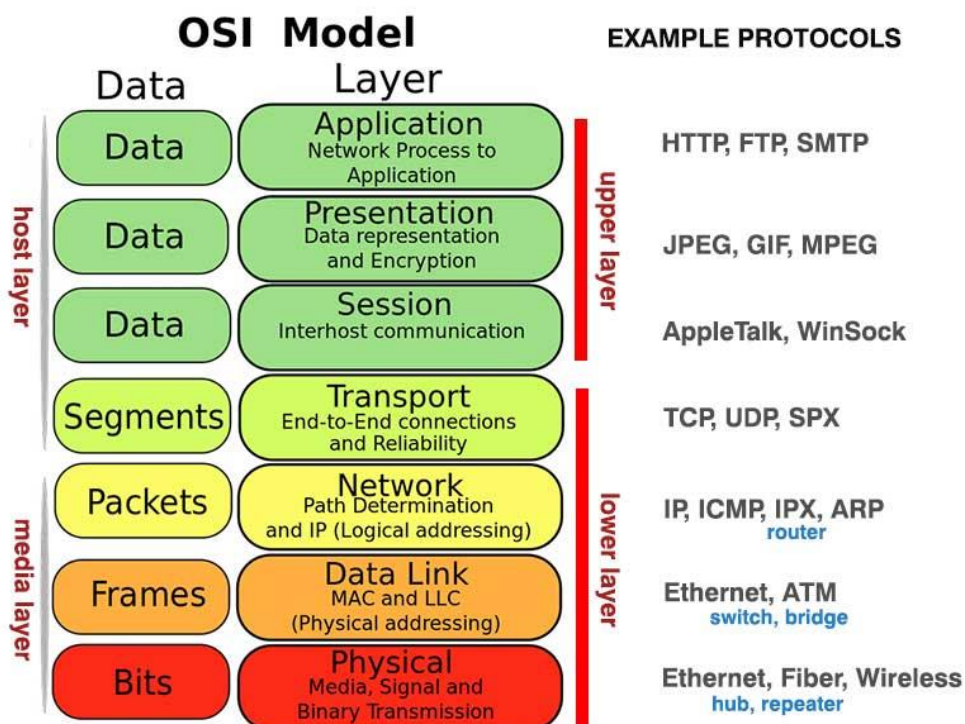
Model referensi OSI secara konseptual terbagi ke dalam 7 lapisan dimana masing-masing lapisan memiliki fungsi jaringan yang spesifik. Model *Open Systems Interconnection (OSI)* diciptakan oleh *International Organization for Standardization (ISO)* yang menyediakan kerangka logika terstruktur bagaimana proses komunikasi data berinteraksi melalui jaringan. *Standard* ini dikembangkan

untuk industri komputer agar komputer dapat berkomunikasi pada jaringan yang berbeda secara efisien.

Model *Layer OSI* dibagi dalam dua group yaitu *upper layer* dan *lower layer*. *Upper layer* fokus pada aplikasi pengguna dan bagaimana file direpresentasikan di komputer. Untuk *Network Engineer*, bagian utama yang menjadi perhatiannya adalah pada *lower layer*. *Lower layer* adalah intisari komunikasi data melalui jaringan aktual. Tujuan utama penggunaan model OSI adalah untuk membantu desainer jaringan memahami fungsi dari tiap-tiap layer yang berhubungan dengan aliran komunikasi data. Termasuk jenis-jenis protokol jaringan dan metode transmisi.

Sesuai dengan namanya, *OSI Layer*, berarti merupakan lapisan – lapisan, Bentuk lapisan – lapisan inilah yang nantinya harus dilewati oleh paket data. Proses transmisi melewati *OSI layer* ini terjadi setiap kali paket data akan ditransmisikan, baik itu transmisi paket data dari *server*, serta transmisi paket data menuju *client*. Jadi, apabila bisa dilihat secara kasat mata, *OSI layer* terdapat di dalam komputer *server* dan juga komputer *client*.

OSI Layer disebut sebagai lapisan, karena memang model referensi OSI ini diciptakan berlapis – lapis. Lapisan – lapisan pada *OSI layer* ini dibuat dengan tujuan agar setiap paket data dalam sebuah jaringan bisa melewati *layer* tersebut sebelum pada akhirnya bisa saling terkoneksi. Berikut ini adalah gambaran tujuh lapisan dari *OSI Layer* yang dapat dilihat pada Gambar 2.3.



Gambar 2.3 OSI Model

Karena setiap OSI *layer* memiliki fungsi masing-masing maka di bawah ini adalah penjelasan fungsi dari ke-tujuh lapisan OSI *layer*:

1. *Physical Layer*

Layer pertama adalah *physical layer*. Fungsi dari *physical layer* adalah mendefinisikan media transmisi jaringan, mendefinisikan metode persinyalan, mendefinisikan arsitektur jaringan dan pengaplikasian topologi jaringan.

2. *Data Link Layer*

Lapisan berikutnya pada OSI *layer* adalah *data link layer*. Merupakan salah satu *layer* yang penting, karena memiliki fungsi untuk menentukan bagaimana setiap bit dari data dikelompokkan ke dalam *frame* dan menentukan bagaimana sebuah perangkat keras dapat beroperasi. Terdapat dua *level* pada lapisan *data link layer* yaitu *logical link control* (LLC) dan *media access control* (MAC).

3. *Network Layer*

Lapisan selanjutnya adalah *network layer*. Fungsi utama dari *network layer* ini adalah untuk membantu mendefinisikan *IP address* atau internet protokol, sehingga tiap komputer dapat terhubung dengan satu jaringan. Fungsi berikutnya dari *network layer* adalah membuat header pada paket-paket data dan melakukan proses *routing*. Fungsi dari beberapa *hardware* jaringan, seperti *router* dan juga fungsi *hub* berjalan pada *layer* ini, dengan cara melakukan pemecahan paket data dan juga melakukan proses *routing*.

4. *Transport Layer*

Sesuai dengan namanya, *transport layer* merupakan lapisan OSI yang memiliki tugas sebagai pengantar. Fungsi utama dari *transport layer* pada lapisan OSI ini adalah memecahkan data ke dalam paket-paket data yang selanjutnya melakukan penomoran pada paket-paket data, sehingga nantinya dapat disusun kembali dengan mudah. fungsi lainnya adalah mentransmisikan data dari *session layer* menuju *network layer*, maupun sebaliknya.

5. *Session Layer*

Lapisan selanjutnya pada OSI adalah *session layer*. Lapisan *session layer* ini memiliki fungsi utama untuk mendefinisikan bagaimana sebuah koneksi bisa dibangun, serta dapat mendefinisikan *management* dari sebuah koneksi, seperti memutuskan dan juga memelihara koneksi.

6. *Presentation Layer*

Layer kedua pada saat data mulai ditransfer dan bertindak sebagai *layer* ke-6 ketika sebuah komputer menerima paket data disebut dengan nama *presentation layer*. Fungsi utama dari *presentation layer* adalah melakukan *translate* data yang akan ditransmisikan menuju sebuah *application* (aplikasi). Apabila merupakan proses awal, lapisan ini berfungsi untuk

menterjemahkan aplikasi menjadi sebuah data yang akan ditransmisikan, begitupun sebaliknya, ketika memasuki proses akhir, *presentation layer* akan menterjemahkan data yang ditransmisikan ke dalam aplikasi.

7. *Application Layer*

Application layer merupakan lapisan yang pertama pada saat sebuah data mulai ditransfer dan merupakan lapisan terakhir yang dilewati pada saat komputer *client* menerima data tersebut. Fungsi dari *application layer* adalah menyajikan interface antara aplikasi dengan jaringan, membuat pesan-pesan berupa kesalahan pada jaringan, menampilkan display dari sebuah jaringan dan mengatur bagaimana sebuah aplikasi mampu untuk mengakses jaringan. Protokol pada *application layer* adalah HTTP, FTP dan SMTP.

2.2.2 Cara Kerja OSI Layer

Seperti sudah disebutkan sebelumnya, OSI memiliki beberapa *layer* atau lapisan, tepatnya adalah 7 OSI *layer*. Cara kerja dari ke-7 OSI *layer* ini adalah dua kali dalam setiap transmisi paket data di dalam sebuah jaringan, yaitu:

1. Saat paket atau *bit* data ditransmisikan dari *server*

Merupakan proses awal dari sebuah jaringan. Pada saat ini, paket atau *bit* data yang dimiliki *server* akan ditransfer dan ditransmisikan melalui jaringan ke komputer-komputer *client*. Proses transmisi jaringan ini menggunakan beberapa *hardware* jaringan, seperti *hub*, *switch*, *router*, dan sebagainya.

Dalam aplikasinya dan hubungannya dengan model referensi OSI, paket atau bit data pertama kali ditransmisikan oleh *server* menuju jaringan dengan melewati *application layer* terlebih dahulu. Baru kemudian, setelah *application layer* bisa bekerja dengan optimal dan baik, proses tersebut dilanjutkan melewati *layer-layer* berikutnya, yaitu *presentation*, *session* dan *physical*.

Bit data yang sudah melewati *physical layer* akan langsung diproses oleh perangkat keras jaringan komputer, seperti *hub*, *switch*, *router* dan sebagainya. Proses ini terjadi baik dengan menggunakan koneksi jaringan kabel, maupun koneksi jaringan *wireless* atau tanpa kabel.

2. Saat paket atau bit data ditransmisikan menuju *client*

Proses kedua yang terjadi setelah paket atau *bit* data sudah ditransmisikan melalui jaringan adalah proses dimana *bit* data tersebut bisa muncul di dalam komputer *client*. agar paket atau *bit* data tersebut bisa terhubung dengan komputer *client*, maka proses transfer data kembali terjadi melewati *layer-layer* pada OSI.

Namun, perbedaannya dari proses pertama adalah pada saat paket atau *bit* data akan ditransmisikan ke komputer *client*, *layer* yang digunakan

terbalik. Jadi, yang pertama kali digunakan adalah *physical layer*. Data yang sudah berada pada perangkat keras jaringan, seperti *hub*, *switch*, *router*, dan sebagainya akan melewati *physical layer*. Pada saat melewati *physical layer*, maka *layer* ini akan mengidentifikasi topologi jaringan yang digunakan, serta mulai memproses setiap *bit* data yang akan ditransmisikan pada komputer *client*.

Barulah setelah *physical layer* terlewati, setiap paket atau *bit* data ditransmisikan melewati *layer-layer* berikutnya, hingga mencapai *application layer*. Pada *application layer*, setiap *bit* atau paket data yang ditransmisikan akan muncul pada komputer *client* dalam bentuk *software*.

Itulah kedua proses jaringan yang terjadi. Meskipun kelihatannya jaringan adalah sesuatu yang sederhana, namun ternyata prosesnya cukup panjang agar bisa berjalan dengan sangat lancar.

2.2.3 Desain Protokol Komunikasi

Dalam desain sistem komunikasi sederhana, terdapat 4 bagian penting yang harus diketahui di dalam protokol komunikasi yaitu *application*, *transport*, *network* dan *link*. Setiap bagian memiliki tugas masing-masing, berikut penjelasannya

1. *Application*

Pada *layer application* komunikasi menggunakan *hypertext transfer protocol* (HTTP) adalah protokol yang mengatur komunikasi antara *client* dan *server*. Yang menjadi *client* adalah *web browser* atau *device* lain yang dapat mengakses, menerima dan menampilkan konten *web*.

2. *Transport*

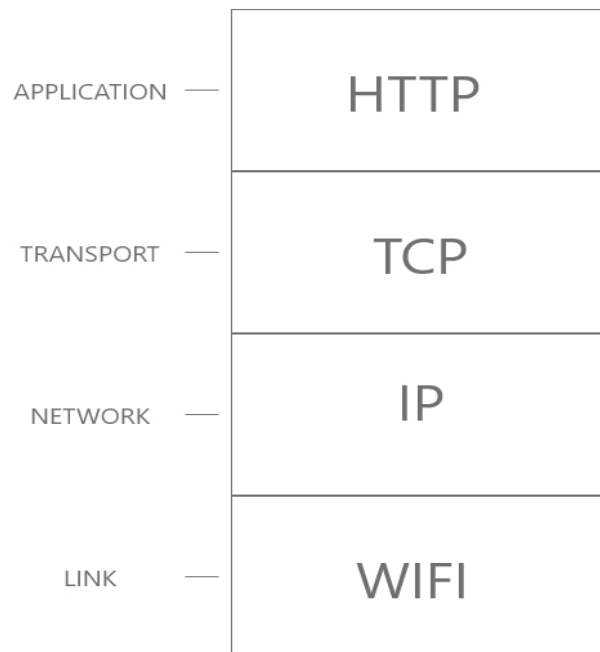
Pada *layer transport* komunikasi menggunakan TCP (*Transmission Control Protocol/Internet Protocol*) memastikan bahwa semua paket diterima, bahwa paket-paket sudah tertata, dan bahwa paket-paket yang rusak dikirimkan kembali. Ini berarti dapat digunakan untuk komunikasi antara beberapa aplikasi, tanpa harus khawatir tentang integritas data atau kehilangan paket.

3. *Network*

Pada *layer network* komunikasi menggunakan *IP Address*, *IP Address* merupakan penomoran untuk alamat agar *sender* dan *reciever* mengetahui setiap alamat dari data yang dikirimkan agar bisa melakukan pengiriman data lebih dari satu [2].

4. *Link*

Link merupakan cara komunikasi yang digunakan pada komunikasi antar alat dan *localhost*. *WIFI* merupakan *platform* yang digunakan untuk perpindahan data antara *server* dan *client*.



Gambar 2.4 Protokol Komunikasi

2.2.4 Data Structure

Dalam melakukan pengiriman data secara *wireless* di dalam komunikasi, data yang dikirim sering disebut dengan *packet*. *Packet* dalam sebuah komunikasi *wireless* biasanya dibagi menjadi tiga yaitu *header* berisi instruksi tentang data yang dibawa oleh *packet*. Isi dari sebuah header seperti *sender IP address* dan *reciever IP address*. Jadi dengan *header* dapat diketahui *IP reciever* dan *sender* dengan begitu komunikasi dapat dilakukan tanpa ada error. Bagian berikutnya adalah *payload*. *Payload* adalah bagian *packet* yang berisikan mengenai data yang ingin

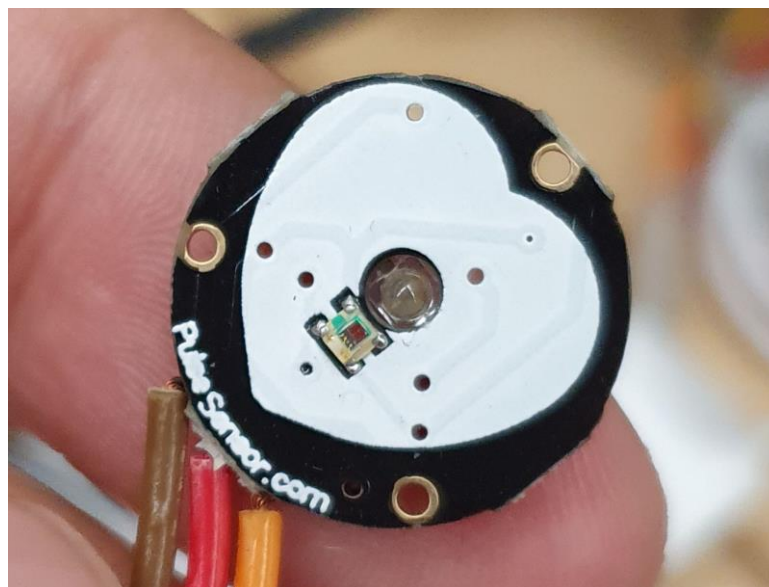
dikirimkan dapat berupa *message* atau *stream* data. Bagian terakhir dalam sebuah *packet* adalah *trailer*. *Trailer* kadang-kadang disebut juga sebagai *footer*, biasanya berisi beberapa bit yang memberi informasi mengenai perangkat penerima bahwa data telah mencapai akhir *packet*.

2.3 Multi-Parameter Monitoring

Monitoring dalam pengertian umum adalah proses rutin pengumpulan data dan pengukuran kemajuan atas objektif program atau Memantau perubahan, yang fokus pada proses dan keluaran. *Monitoring* menyediakan data dasar untuk menjawab permasalahan, pengertian *multi parameter monitoring* adalah dalam pengumpulan data tidak hanya satu data sensor saja yang diolah melainkan terdapat beberapa data sensor yang diolah di dalam *multi parameter monitoring*, jadi *multi parameter monitoring* adalah sebuah teknologi yang memungkinkan untuk melakukan *monitoring* lebih dari satu sensor yang terintegrasi di dalam satu *controller* yang sama yang akan menghasilkan data dari beberapa sensor yang terkoneksi dengan *controller* tersebut.

2.4 Sensor Monitoring

Sensor adalah komponen elektronika yang berfungsi untuk mengubah besaran mekanis, magnetis, panas, sinar, dan kimia menjadi besaran listrik berupa tegangan, resistansi dan arus listrik. *Sensor* sering digunakan untuk melakukan deteksi pada saat pengukuran. Pada penelitian ini, sensor yang digunakan sudah berbentuk modul. Di dalam *monitoring* ini sensor yang digunakan adalah *heart rate sensor*, *weight scale sensor* dan *temperature sensor*.



Gambar 2.5 Modul *Pulse Sensor* AK90

2.4.1 Heart Rate Sensor

Heart Rate Sensor adalah sensor yang dapat menghitung denyut jantung manusia yang diproduksi oleh funky corporation, modul ini dinamakan *pulse sensor* AK90. Sensor ini menggunakan *infrared* dan *photodiode*. *Infrared* akan memancarkan sinyal yang menembus kulit pada tangan yang kemudian akan ditangkap oleh *photodiode*. Konsepnya adalah *infrared* dan *photodiode* akan menangkap perubahan volume darah pada jari tangan pada saat jantung memompa darah keseluruh tubuh dari sinilah data denyut jantung akan didapatkan untuk kemudian diproses.

2.4.2 Sensor Infus (*Weight Scale*)

Weight scale sensor digunakan untuk menghitung berat sebuah infus secara terus-menerus, agar dapat di-*monitoring* kapan infus tersebut habis. Modul *Weight scale* sensor adalah timbangan digital, yang ADC-nya disambung agar dapat membaca berat secara *continuous*, dengan begitu sensor dapat membaca berat dari infus secara terus-menerus dan akan dibuat *limiter* yang nantinya berguna untuk *alarm*. *Alarm* akan berfungsi jika data berat di bawah berat yang diatur pada sensor, ini dapat menjadi pengingat untuk mengganti infus yang sedang digunakan pasien.

Modul HX711 merupakan modul *amplifier* yang digunakan untuk dapat membaca berat infus secara lebih presisi dan tepat, karena *error* di dalam *heathcare* haruslah mendekati 0 maka dari itu digunakan modul ini untuk *amplify* hasil dari *load cell sensor*.



Gambar 2.6 Modul *Load Cell* 20 kg

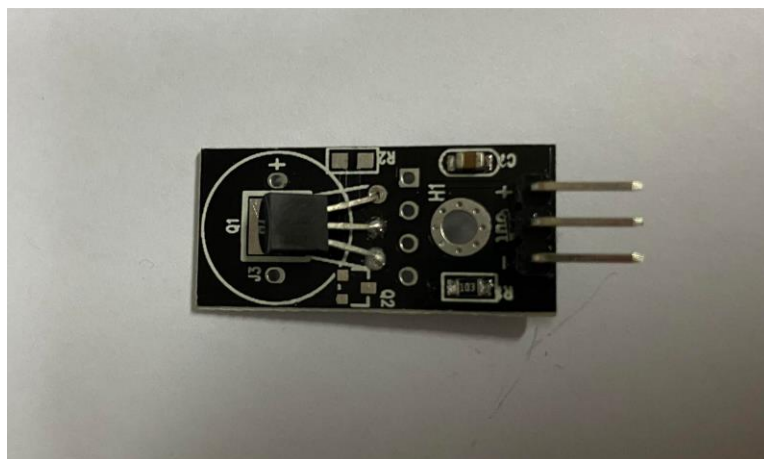


Gambar 2.7 Modul HX711 Amplifier

2.4.3 Sensor *Temperature*

Modul *temperature* yang digunakan adalah modul *temperature* yang berbasis *thermometer digital*, mengapa menggunakan *thermometer digital*, dikarenakan penggunaan *thermometer digital* pada bidang kesehatan merupakan hal yang sering dilakukan oleh pihak rumah sakit karena hasil yang didapat dari *thermometer digital* memiliki Batasan *error* yang kecil, maka dari itu penggunaan *thermometer digital* pada modul *temperature* dipilih karena hasil *temperature* yang dihasilkan oleh *thermometer digital* sangat tepat dan memiliki *error* yang sangat kecil, karena suhu tubuh manusia sangat *sensitive* maka dibutuhkan modul *temperature* yang *sensitive* juga.

Modul *temperature* yang digunakan adalah DS18B20. Sensor suhu DS18B20 adalah sensor suhu yang memiliki keluaran digital. DS18B20 memiliki



Gambar 2.8 Modul DS18B20

tingkat akurasi yang cukup tinggi, yaitu $0,5^{\circ}\text{C}$ pada rentang suhu -10°C sampai $+85^{\circ}\text{C}$. Sensor suhu pada umumnya membutuhkan ADC dan beberapa pin *port* pada mikrokontroler, namun DS18B20 ini tidak membutuhkan ADC agar dapat berkomunikasi dengan mikrokontroler dan hanya membutuhkan *1 wire* saja.

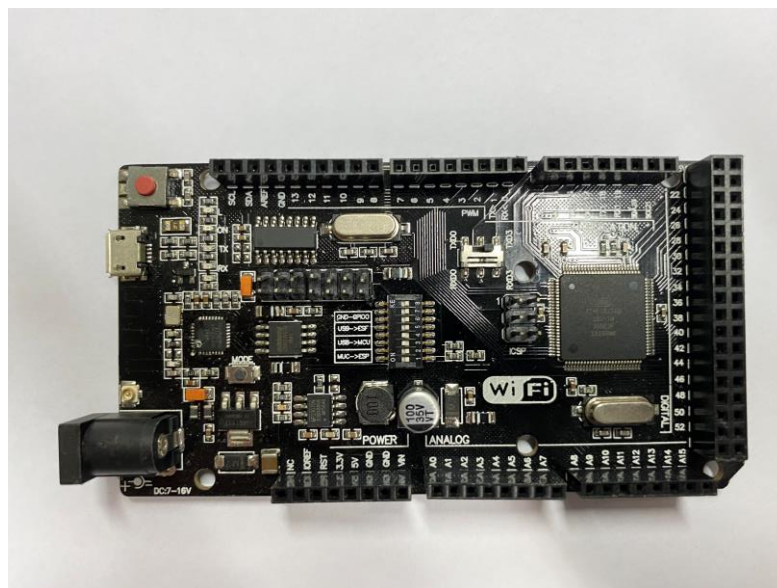
Cara kerja secara umum besaran fisis yang diukur adalah suhu tubuh. Suhu tubuh tersebut akan diolah dalam sensor suhu DS18B20 yang mengubah besaran fisis menjadi tegangan. Data tegangan tersebut akan diolah dengan mikrokontroler.

2.5 Mikrokontroler

Sebuah alat yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program di dalamnya, di dalam mikrokontroler juga terdapat *memory* penyimpanan data. Mikrokontroler yang digunakan adalah mikrokontroler arduino, menggunakan arduino karena pengendali mikro *single-board* yang bersifat *open-source*.

2.5.1 Arduino Mega 2560 With Wifi Built-in ESP8266

Arduino adalah platform *open-source* yang digunakan untuk membangun proyek-proyek elektronik. Arduino terdiri dari kedua papan sirkuit yang dapat diprogram fisik (sering disebut sebagai mikrokontroler) dan perangkat lunak, atau IDE (*Integrated Development Environment*) yang berjalan di komputer, digunakan untuk menulis dan mengunggah kode komputer ke papan fisik. Penggunaan arduino saat ini sangat bervariasi. Arduino dapat menjalankan proyek elektronik sederhana dengan *library* yang disediakan pada IDE dan beberapa tersedia di internet. Pada penelitian ini, arduino digunakan sebagai kontroler dalam pengambilan *data Pulse sensor, weight scale sensor* dan *temperature*.



Gambar 2.9 Arduino Mega 2560 With Wifi Built-in ESP8266

2.6 Database

Database adalah kumpulan fakta dan angka yang saling berkaitan dan dapat diproses untuk menghasilkan informasi. Pendefinisian *database* meliputi spesifikasi berupa tipe data, struktur data dan juga batasan-batasan pada data yang akan disimpan. *Database* merupakan aspek yang sangat penting dalam sistem informasi karena berfungsi sebagai gudang penyimpanan data yang akan diolah lebih lanjut. *Database* menjadi penting karena dapat mengorganisasi data, menghindari duplikasi data, menghindari hubungan antar data yang tidak jelas dan juga *update* yang rumit. Proses memasukkan dan mengambil data dari media penyimpanan data memerlukan perangkat lunak yang disebut dengan *Database Management System* (DBMS). DBMS merupakan sistem perangkat lunak yang memungkinkan *database user* untuk memelihara, mengontrol dan mengakses data secara praktis dan efisien. DBMS berfungsi sebagai penghubung antara *database* dengan pengguna yang ingin mengakses data dari *database*. DBMS dapat berbentuk aplikasi *interface* yang memudahkan pengguna awam untuk mendapatkan informasi dari *database*. Penggunaan DBMS dapat lebih efektif jika diterapkan pada sebuah *relational database*.

Terdapat beberapa tipe dalam membuat sebuah database yang pertama adalah *database* NoSQL, *database* NoSQL adalah *database* yang tidak menggunakan relasi antar tabel dan tidak menyimpan data dalam format tabel kaku (kolom yang fix) seperti layaknya *relasional database*. Kedua adalah *database* SQL, *database* MySQL adalah perangkat lunak sistem manajemen basis data SQL atau dalam bahasa Inggris disebut *database management system* (DBMS) yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia.

Perbedaan MySQL dan NoSQL terletak pada cara penulisan *database*.SQL menggunakan *relasional* sebagai penyambung antara data-data di dalam tabel *database*. Sedangkan NoSQL tidak menggunakan *relasional* sebagai cara mereka untuk menyambungkan antar data. NoSQL tidak menggunakan *schema relational*, Pada SQL user harus mendefinisikan tabel yang akan digunakan. Pada NoSQL tidak perlu untuk mendefinisikan terlebih dahulu Tabel yang akan digunakan.

Dalam *database* SQL data berbentuk tabel yang terdiri dari sejumlah baris, sedangkan Pada NoSQL data tidak memiliki definisi skema standar yang harus dipatuhi. NoSQL memiliki skema yang dinamis sedangkan pada *database* SQL mengikuti skema yang telah ditetapkan. *Database* NoSQL merupakan *horizontal* terukur sedangkan pada SQL *database vertical* terukur.

Dari penjelasan di atas maka *database* yang akan digunakan adalah database MySQL dikarenakan dengan MySQL dapat membuat skema pada data dan database MySQL merupakan *database vertical* terukur yang tepat digunakan untuk *database* pada penelitian ini.

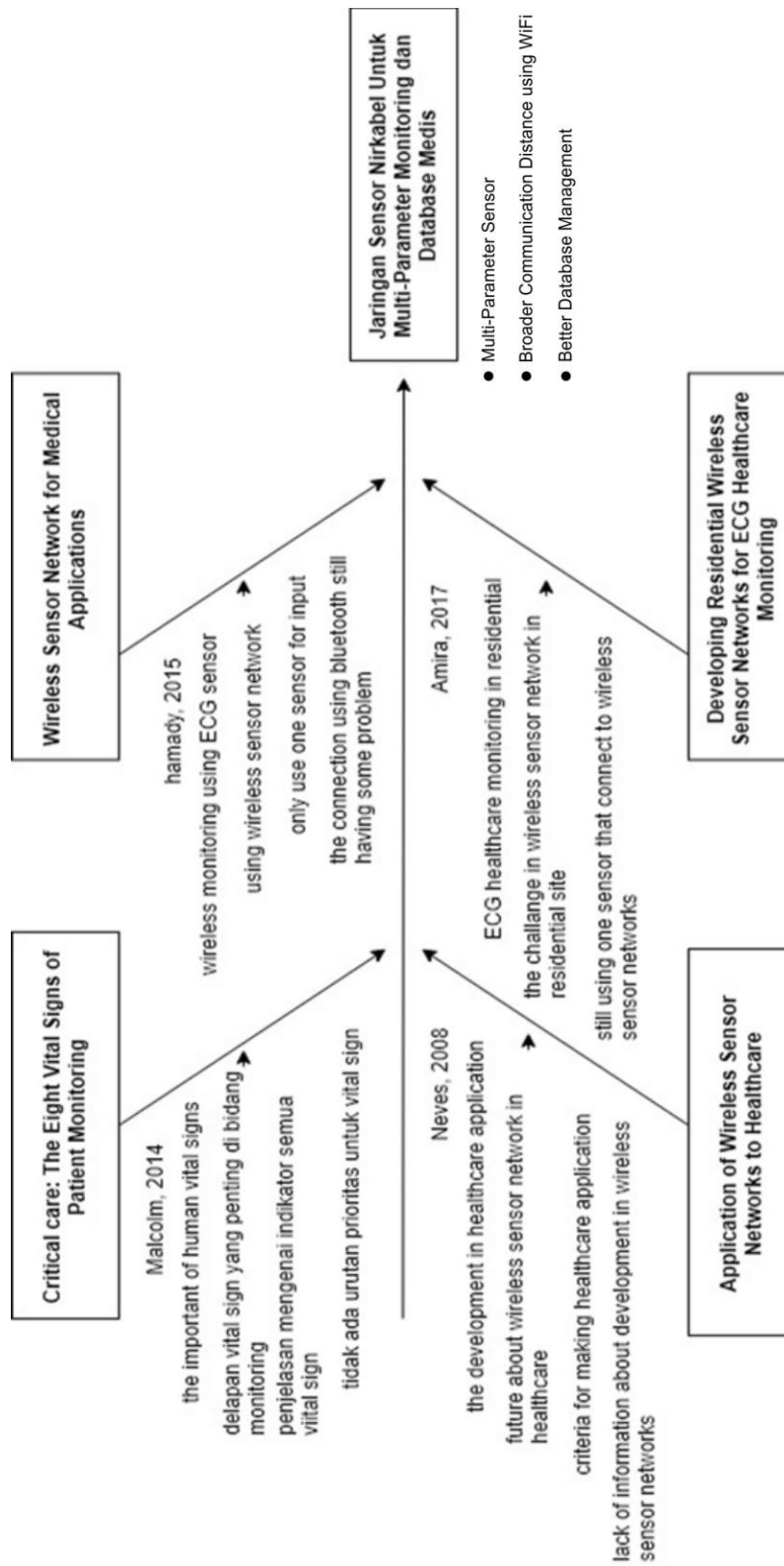
2.7 Rasional

Pada penelitian tugas akhir ini suatu *system vital sign monitoring* yang dimana terdapat tiga sensor yang digunakan, tiga sensor ini memiliki fungsinya masing-masing yaitu membaca berat dari infus, *heart rate* dan suhu tubuh, *Pulse Sensor* yang digunakan di sini difungsikan untuk membaca *heart rate* pada pasien. Semua sensor yang ada pada *monitoring* ini terhubung dengan mikrokontroler yang memproses data dari sensor menjadi data yang dapat ditampilkan dalam monitor nantinya. Setelah diproses pada mikrokontroler maka data tersebut akan dikirim secara wireless yaitu menggunakan wifi dengan menggunakan modul ESP8266. Selanjutnya data yang sudah dikirim oleh ESP8266 akan diterima oleh wifi pada monitor, data tersebut akan diproses menjadi informasi yang dapat di-*display* dalam monitor yang ada di komputer. Informasi akhir dari alat ini digunakan untuk memudahkan pengguna maka data dari sensor dapat disimpan dalam *database* dan dapat langsung dijadikan sebuah laporan.

Secara *hardware* akan dirancang menggunakan beberapa komponen yaitu dengan modul *Pulse Sensor* AK90 dengan menggunakan modul ini maka bisa didapatkan data *Heart Rate* secara *real-time* selanjutnya akan diproses di mikrokontroler untuk proses ADC (*Analog to Digital*). Selain untuk proses ADC ini mikrokontroler juga melakukan proses matematis sehingga didapatkan BPM nya. Pada tahap selanjutnya yaitu dilakukan transfer melalui *wireless*, *wireless* disini menggunakan wifi untuk melakukan transfer ke komputer berbasis *windows* yang nantinya akan menampilkan informasi tersebut pada *display* di komputer.

Hasil yang diharapkan dari proyek ini adalah sebuah *monitoring* yang memiliki *multi-parameter* yang dapat meningkatkan kualitas dari rumah sakit dan dapat meningkatkan kualitas *monitoring* pada rumah sakit. Hasil akhir dari penelitian ini dapat langsung diakses dalam komputer walaupun pasien yang dimonitor lebih dari satu, dikarenakan sudah menggunakan *wireless*, komputer dapat menampilkan semua kondisi pasien yang sedang dirawat di rumah sakit tersebut.

Berdasarkan metode pada penelitian-penelitian sebelumnya, penelitian ini memperoleh sebuah konsep yang bertujuan untuk mendeteksi *multi vital sign* dalam satu *monitoring system* dan dapat melakukan *monitoring* lebih dari satu pasien secara bersamaan. Secara umum rasional dari tugas akhir ini digambarkan dalam fishbone diagram pada Gambar 2.10.

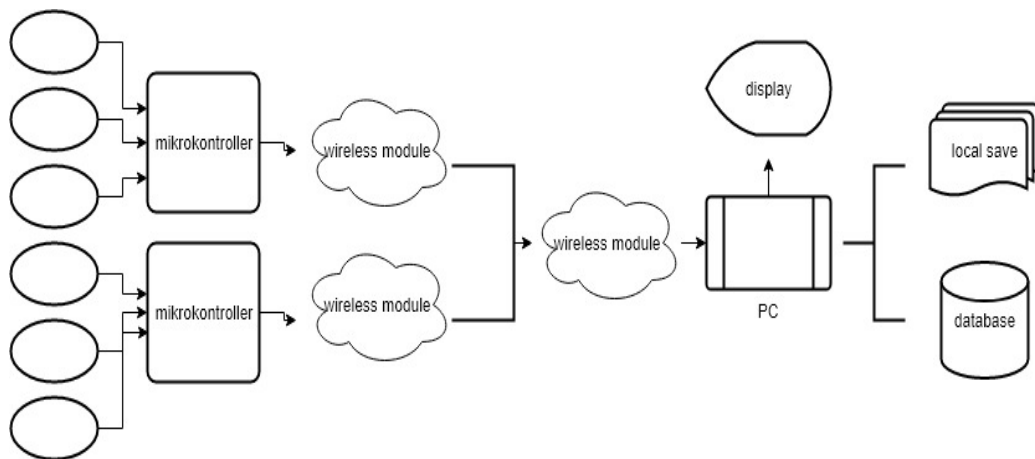


Gambar 2.10 Fishbone Diagram

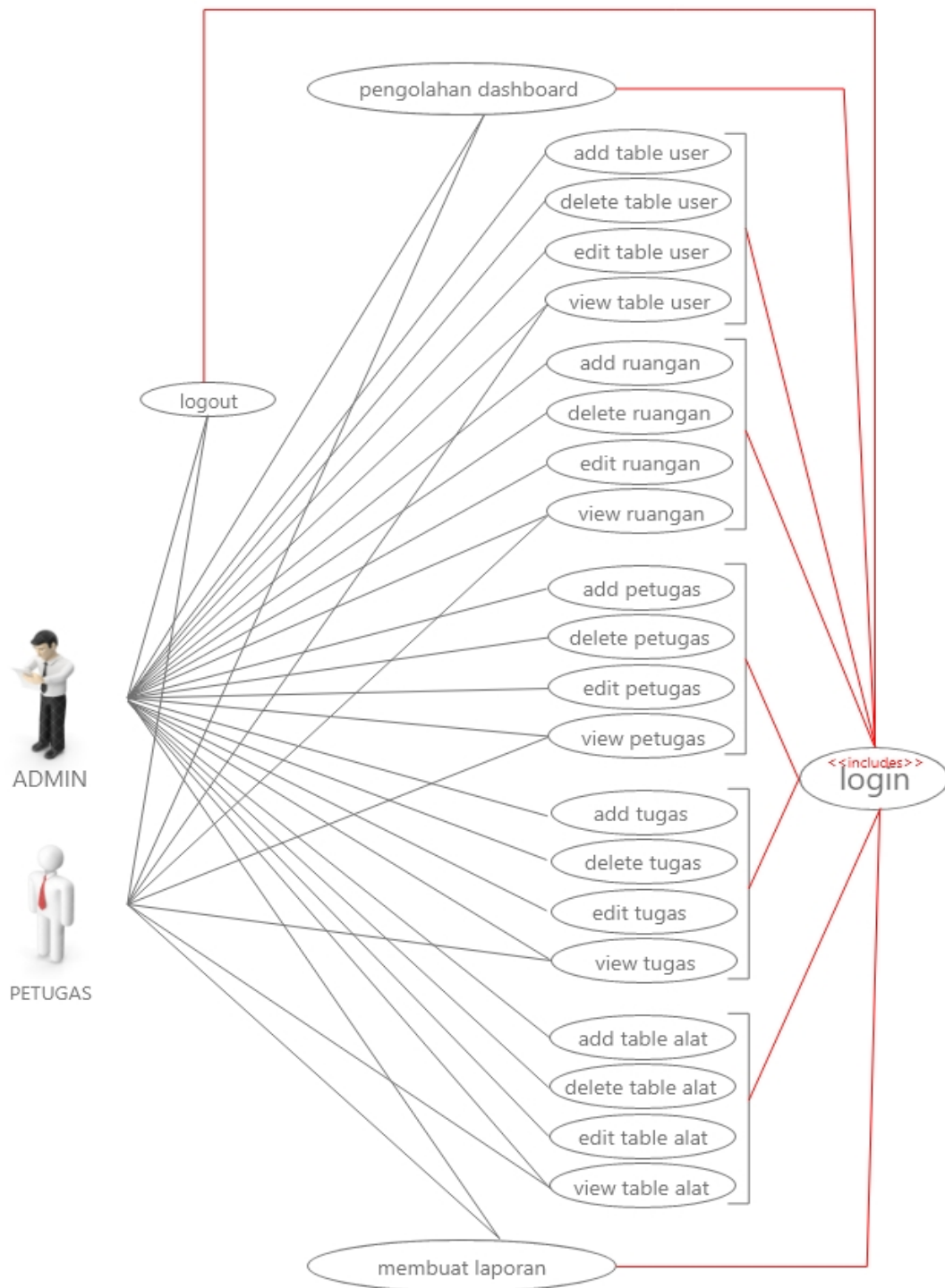
BAB 3 PERANCANGAN SISTEM

3.1 Gambaran Umum Sistem

Secara umum *system monitoring wireless* yang dibuat digambarkan oleh Gambar 3.1 blok pertama yaitu macam-macam sensor yang dibutuhkan untuk melakukan *healthcare monitoring*, *monitoring* disini adalah *monitoring* suhu tubuh, *monitoring* berat cairan infus dan *monitoring heart rate*. Jadi pada bagian blok pertama terdapat 3 sensor *monitoring* yang akan disambungkan dengan mikrokontroler. Pada blok kedua adalah mikrokontroler yang digunakan untuk melakukan proses pengambilan data dari sensor-sensor yang ada pada blok pertama. Mikrokontroler ini juga digabungkan dengan *module* wifi agar dapat mengirimkan data dari sensor secara *wireless*. Pada blok ketiga adalah *system* penggambaran *system wireless* yang digunakan. Pada *system* ini setiap *wireless module* yang terhubung dengan mikrokontroler memiliki *IP address* masing-masing yang berfungsi untuk memberikan perbedaan antara mikrokontroler satu dengan mikrokontroler yang lainnya. Pada blok keempat adalah PC atau *application* yaitu dimana *application* dapat mengambil setiap *IP address* dari setiap mikrokontroler yang digunakan untuk *monitoring*. Pada aplikasi ini juga ditampilkannya *display* data *monitoring* yang didapat dari setiap sensor dari blok pertama. Aplikasi ini juga memiliki *system database* ruang yang dapat melakukan *management* ruang dan *system database user* yang dapat membagi otoritas pada setiap *level user* yang melakukan *login* pada aplikasi.



Gambar 3.1 Gambaran Umum Sistem



Gambar 3.2 Use Case Sistem Aplikasi

3.2 Pengguna Sistem

Pengguna dalam *system monitoring* ini adalah petugas atau perawat dan admin. Admin merupakan pengguna yang bertugas untuk melakukan registrasi pasien seperti nama, tanggal lahir, tempat lahir dan data pasien umum lainnya. Admin disini juga memiliki otoritas untuk melakukan *delete* dan *edit* pada setiap tabel *management* ruangan yang ada pada aplikasi. Admin juga memiliki otoritas untuk melakukan perubahan *IP address* untuk setiap *table* ruangan dan berwenang untuk menambahkan *user* petugas yang dapat menggunakan aplikasi *healthcare* ini. Petugas atau perawat merupakan pengguna yang hanya memiliki otoritas untuk melakukan *monitoring* pada setiap ruang yang sedang aktif, perawat tidak memiliki otoritas untuk melakukan perubahan *system* pada setiap *management* ruang yang ada di aplikasi. Penggunaan tombol *edit* dan *delete* akan dimatikan pada saat *user* adalah petugas atau perawat.

3.3 Perancangan Use Case

Penggunaan *system* serta fungsi-fungsi yang dapat dikerjakan oleh masing-masing pengguna akan digambarkan dengan *use case* diagram, definisi dari setiap *use case* adalah sebagai berikut:

- a. *Use Case Login/ Sign in*
Use Case login dilakukan oleh setiap pengguna yang sudah didaftarkan oleh admin untuk masuk ke dalam *system*.
- b. *Use Case Log Out*
Use Case Log Out digunakan oleh pengguna yang ingin keluar dari *system*.
- c. *Use Case Dashboard*
Use Case dashboard adalah *interface* yang menampilkan berbagai data inti seperti data dari sensor *monitoring* yang ditampilkan dalam bentuk *graph* dan data ruang serta informasi pasien pada ruang tersebut.
- d. *Use Case User*
Use Case user merupakan sebuah *interface* yang berfungsi untuk mendaftarkan *level user* dan melakukan registrasi *user*, di dalam *Use Case* ini terdapat *database* yang berisikan user-user yang dapat *login* pada aplikasi ini.
- e. *Use Case Ruangan*
Use Case ruangan merupakan tempat registrasi *database* ruangan yang di dalamnya akan ada informasi mengenai nomer ruangan dan pasien yang dirawat diruang tersebut.
- f. *Use Case Tugas*
Use Case tugas merupakan *database* yang menyimpan data mengenai petugas dan nomer ruang, dengan *database* ini kita dapat

mengetahui nama petugas yang sedang bertanggung jawab pada ruang yang aktif.

g. *Use Case* Alat

Use Case alat merupakan tempat database yang berfungsi untuk meregistrasi alat pada setiap ruangan yang mau digunakan, dengan database ini maka setiap alat memiliki nomer ruang masing-masing. Setiap alat dibedakan dengan nomer *IP address*.

h. *Use Case* Laporan

Use Case laporan merupakan bagian dari aplikasi yang bertugas untuk membuat sebuah laporan *monitoring* secara otomatis yang berfungsi untuk mempermudah *user* mendapatkan data *monitoring* dari pasien.

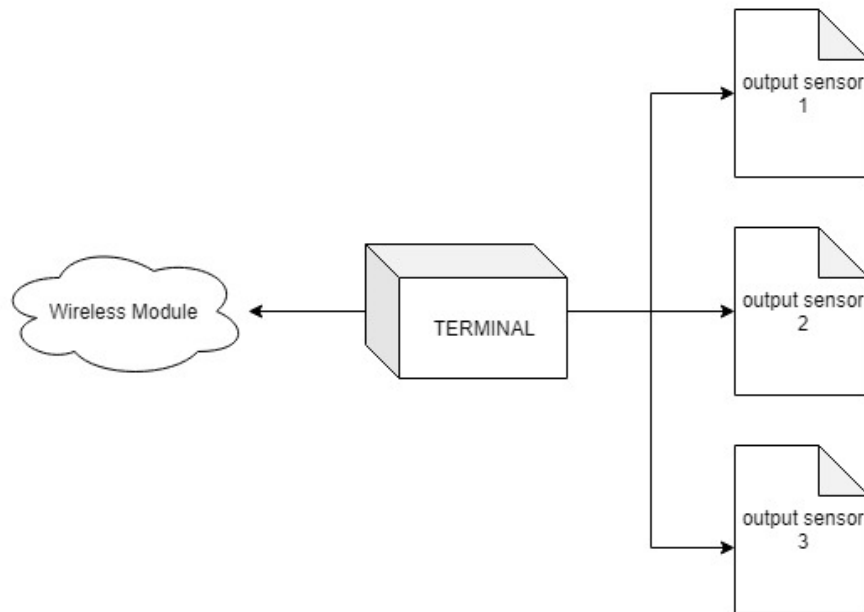
3.4 Perancangan *Hardware*

Rancangan *Hardware* terdiri dari 3 bagian utama yaitu *base node* dan *wireless node* dan *database system*. Secara menyeluruh system dapat digambarkan seperti Gambar 3.1. Pada gambar tersebut dicontohkan dengan jumlah pasien 2 orang yang data dari pasien tersebut dikirim secara *wireless* menuju ke PC untuk di-*process* pada *display* dan disimpan pada *database*. Pada bagian pertama setiap pasien mengirimkan 3 data dari sensor yang ditempelkan pada pasien, setiap data diproses terlebih dahulu pada mikrokontroler dan mikrokontroler tersebut akan mengirim secara *wireless* data dari sensor yang sudah diproses, data yang diterima oleh PC akan diolah menjadi data yang akan ditampilkan pada *display* dan disimpan pada *local save* PC maupun dikirim ke *database* pada server.

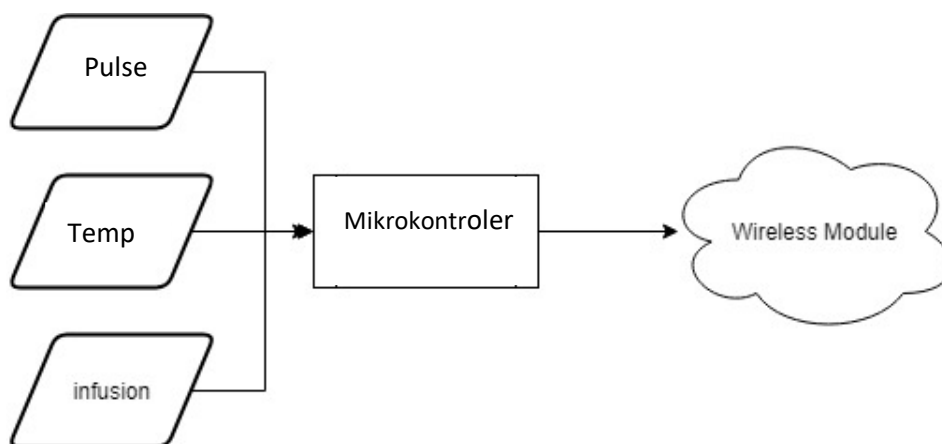
Proses pengiriman data secara *wireless* menggunakan *system address* agar pengiriman data yang dilakukan setiap sensor tidak saling menimpa, dengan menggunakan *system address* pengiriman data pada setiap sensor tidak akan saling menimpa dan transfer data akan selalu masuk pada *address* yang sudah ditentukan.

Setiap rancangan *hardware* yang dibuat memiliki fungsi masing-masing, untuk perancangan *hardware* pada *base node*, fungsi *base node* itu sendiri adalah sebagai terminal atau sebagai tempat data diterima dan diolah. Terminal menggunakan komputer yang dihubungkan dengan *wireless module* agar terminal dapat menerima data dengan metode nirkabel. Komputer akan menghasilkan *output* yaitu kondisi isi dari infus dengan menggunakan metode *weight scale*, *temperature display* dan *display pulse sensor*.

nodes



Gambar 3.3 Base Node



Gambar 3.4 Child Node

Perancangan *hardware* pada *wireless node*, fungsi utama *hardware* ini adalah sebagai media komunikasi data agar dapat mencapai terminal. Pada *wireless node* terdapat 3 sensor yang digunakan yaitu *weight scale sensor*, *temperature sensor* dan *pulse sensor*, ketiga sensor tersebut dihubungkan dengan Arduino mega. Arduino mega disini berfungsi untuk mengolah data *input* agar dapat menjadi data yang bisa dikirim ke PC atau aplikasi, setelah itu terdapat *wireless module* yang

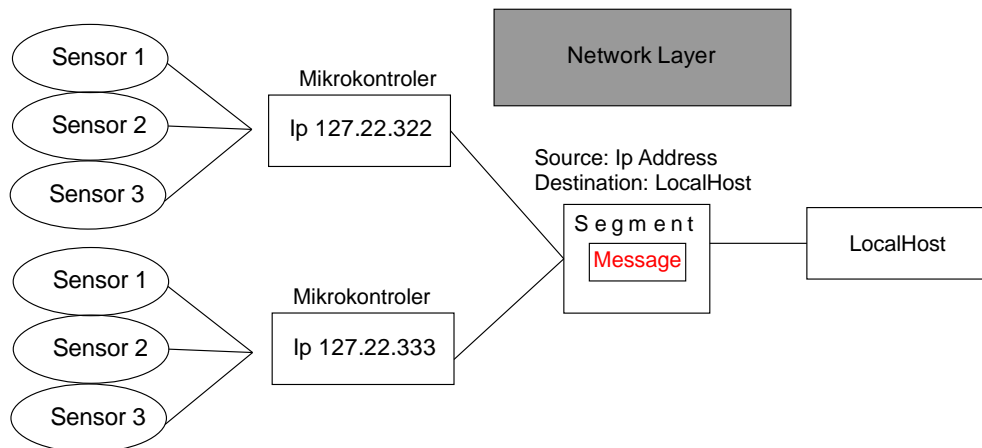
berfungsi sebagai koneksi antara *wireless node* dengan *base node* koneksi yang digunakan adalah koneksi *TCP/IP*.

Perancangan *hardware* pada *database system*, lebih pada *hardware server* yaitu *hardware* yang dapat menyimpan data yang masuk ke dalam *database*, *system database* yang digunakan merupakan *database* *mySQL* yang menyimpan setiap *management* data seperti *management* ruang, *management* alat, *management* user, *management* tugas.

3.5 Perancangan Software

Pada bagian ini, perancangan *software* dibagi menjadi 2 bagian yaitu *software* terminal dan *software* mikrokontroler, untuk *software* mikrokontroler yang digunakan adalah *software* *Arduino*, *software* ini dibuat untuk dapat membaca data dari setiap *input* sensor yang dimiliki, *input* sensor itu adalah sinyal *pulse sensor*, *Temperature* dan *weight scale* untuk berat infus.

Pada bagian *software* *PC*, *software* dibuat untuk dapat menerima data yang dikirimkan oleh sensor, *software* ini juga memiliki fungsi untuk mengolah dan memproses data yang dikirim oleh sensor untuk menjadi informasi yang diinginkan. Pada perancangan *software* pengiriman data secara *wireless* membutuhkan sebuah sistem *security* dalam melakukan transfer data, agar data yang dikirimkan aman dan data yang diterima nantinya merupakan data yang *valid*. Masalah dalam sistem *security* yang sering terjadi di dalam komunikasi *wireless* adalah pengiriman data yang tidak terenkripsi, pengiriman data yang tidak terenkripsi dapat menyebabkan data dapat dengan mudah dibaca pada saat pengiriman berlangsung, data dapat dibaca pada saat pengiriman dikarenakan data yang dikirimkan masih berbentuk data yang berisikan nilai-nilai sensor yang asli yang belum terenkripsi oleh user. Data yang berhasil dibaca oleh pihak luar dapat menyebabkan data yang dikirimkan ke terminal dapat diduplikasi dan dimasukkan data yang tidak *valid*. Hal tersebut dapat menyebabkan informasi yang diterima oleh terminal bukanlah sebuah data *valid*. Maka dari itu *security* dalam pengiriman data *wireless* adalah hal yang penting, pengimplementasian sistem *security* dapat dilakukan dengan melakukan enkripsi data agar data yang dikirimkan tidak mudah untuk dibaca oleh pihak luar, melakukan enkripsi data merupakan hal yang sering dilakukan di dalam komunikasi data secara *wireless*, data yang dikirimkan oleh sensor pertama dirubah menjadi data yang tidak bisa dibaca oleh user pada saat pengiriman dan pada saat data diterima oleh terminal, data tersebut didekripsi menjadi data yang dapat diolah nantinya di terminal, dengan begitu data yang dikirimkan tidak bisa dibaca oleh pihak luar dikarenakan data yang dikirimkan sudah terenkripsi. Cara lain dalam mengatasi permasalahan *security* adalah menggunakan protokol yang memiliki *security* yang cukup baik.



Gambar 3.5 Rancangan Komunikasi

3.5.1 Protokol Komunikasi

Rancangan desain sistem komunikasi sederhana, terdapat 4 bagian penting yang harus diketahui *application*, *transport*, *network* dan *link*. Dalam melakukan rancangan sebuah protokol komunikasi setiap data yang dikirimkan terdapat layer yang memiliki fungsi tersendiri dan berhubungan satu sama lain. Setiap bagian memiliki tugasnya masing-masing, berikut penjelasannya

1. *Application*

Perancangan komunikasi disini pertama pada *application* menggunakan HTTP (*hyperText Transfer Protocol*), penggunaan HTTP dapat dicontohkan seperti jika pada saat memasukan *IP localhost* pada *address bar* yang ada pada *browser (client)*, HTTP akan mengirimkan HTTP GET yang merupakan sebuah *request* pada server. Server memproses *request* yang dikirim HTTP dan merespon dengan menampilkan halaman web yang di-*request* tersebut. Pada perancangan disini HTTP pada aplikasi medis melakukan HTTP GET yang nantinya data tersebut akan diterima di aplikasi dan langsung dimasukan kedalam *database*, pada saat masuk ke dalam *database* data tersebut diberi *time and date* agar dapat diketahui kapan data tersebut ditangkap oleh aplikasi.

2. *Transport*

Pada bagian *transport* digunakan TCP (*Transmission Control Protocol/Internet Protocol*) penggunaan TCP pada bagian sender dapat dilihat pada Gambar 3.5, pada gambar tersebut dimisalkan terdapat 2 alat yang berbeda keduanya memiliki *IP address* yang berbeda *IP address* itu adalah *source* setiap alat agar nantinya *reciever* mengetahui *source* dari data tersebut dari *IP address*. Dapat digambarkan jika salah satu *port* mengirimkan data, data tersebut diletakan di dalam sebuah *segment* yang

dalamnya terdapat informasi *source* data dan *destination* data. Pada bagian *reciever* dapat dilihat pada Gambar 3.5, terlihat data yang sudah diletakan dalam *segment* akan masuk ke *destination* yang dituju oleh data tersebut

3. *Network*

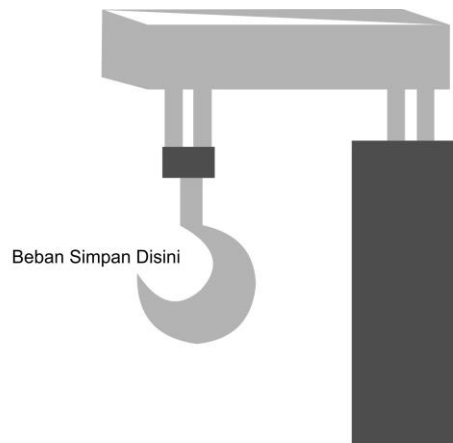
Penggunaannya adalah untuk pemberi tanda pada setiap alat jadi data yang dikirim oleh alat A dan alat B dapat dibedakan, pembedanya adalah *IP address* dari setiap alat tersebut dengan begitu *localhost* dapat membedakan setiap alatnya, yang dimaksud dari membedakan adalah alat A akan masuk pada id alat 1 pada *localhost* dan alat B akan masuk pada id alat 2. Maka dari itu data yang dikirimkan dapat dibedakan pada saat proses transfer berlangsung. Data yang masuk ke dalam *database* juga memiliki penomoran yang difungsikan untuk memberi tanda pada setiap data yang diterima di aplikasi, dengan penomoran tersebut data yang diterima oleh aplikasi dapat dilihat *success rate* pengirimannya. Jika terdapat nomer data yang terlewatkan maka terjadi data *loss* atau terdapat data yang tidak masuk ke dalam *database* aplikasi.

4. *Link*

Link disini adalah cara komunikasi yang digunakan, pada komunikasi antar alat dan *localhost* yang digunakan adalah wifi. Selain wifi yang sering digunakan sebagai *link* dalam sebuah komunikasi data adalah koneksi dengan menggunakan kabel LAN. Pada rancangan komunikasi disini dikarenakan komunikasi yang diinginkan adalah komunikasi *wireless* maka yang digunakan sebagai penghubung data adalah wifi.

3.5.2 *Pulse sensor Monitoring*

Perancangan *Pulse Sensor* memiliki fungsi agar sensor ini dapat menangkap *heart rate* dari pasien yang sedang dirawat di rumah sakit, Nalai *heart rate* pada *module* sensor ini didapatkan dengan membandingkan waktu terjadinya detak jantung sampai akhir dari detak jantung tersebut, perancangan *pulse sensor* ini pada dasarnya menggunakan metode *photoplethysmography* (PPG) yang mendeteksi perubahan volume darah dilapisan mikrovaskular jaringan. Perubahan volume dideteksi dengan cara menggunakan *green LED*, penggunaan green LED dipilih karena darah menyerap *green light*. setiap kali jantung berdetak, jantung memompa darah ke pembuluh darah maka hasilnya lebih banyak *green light* yang diserap oleh darah menyebabkan intensitas cahaya dari *green light* yang dikirimkan oleh sensor berkurang pada saat ditangkap kembali oleh sensor, dari situ didapatkan kepadatan pulsa darah. Proses berikutnya melakukan perbandingan waktu pulse interval. Pulse interval didapatkan dengan cara menghitung selisih dari waktu second pulse time dikurang dengan first pulse time, setelah itu akan didapatkan *beats per minute* (BPM).



Gambar 3.6 Metode *Weight Scale*

3.5.3 *Weight Scale Infusion*

Metode yang digunakan disini adalah metode membaca berat dengan menggunakan kait, sebenarnya terdapat 2 metode untuk mendapatkan berat dari sebuah objek, yaitu dengan tekanan berat seperti timbangan berat badan atau seperti timbangan neraca yang mengukur berat barang dengan *tension factor*. Metode yang digunakan ditugas akhir ini menggunakan kail yang nantinya akan membaca data berat infus secara *continuous*. Membaca berat infus secara *continuous* dengan *weight scale sensor* memiliki fungsi untuk mengetahui secara *real-time* berapa berat dari tabung infus dengan begitu dapat diketahui jika tabung infus tersebut sudah mau habis dan harus digantikan.

3.5.4 *Temperature*

Metode yang digunakan untuk mendapatkan nilai *temperature* adalah dengan menggunakan sensor yang berfungsi seperti *thermometer digital*, *temperature* yang dihasilkan *thermometer* harus memiliki nilai *error* yang kecil agar hasil lebih presisi. Dalam mendapatkan hasil *temperature* yang baik, pada saat sensor mengirimkan data mentah ke mikrokontroler, mikrokontroler bukan hanya melakukan proses *plotting* tetapi terlebih dahulu melakukan proses konfigurasi yang berfungsi untuk mengecilkan nilai *error* dari sensor *temperature* agar pada saat *plotting* dilakukan, nilai suhu yang ditampilkan nantinya memiliki nilai *error* yang kecil dan dapat diabaikan.

3.6 Rancangan *Interface*

Perancangan *system interface* memiliki beberapa bagian yaitu *login*, *dashboard*, *interface table* petugas, *interface table* tugas, *interface table* user, *interface* laporan, *interface table* alat dan *interface* ruangan, *interface* yang disebutkan di atas dapat diakses sepenuhnya oleh admin, hak akses admin adalah pengguna yang dapat melakukan perintah *add*, *delete* dan *edit* pada setiap *table*

The image shows a login form with a rounded rectangular border. At the top, the word "LOGIN" is written in a large, bold, sans-serif font. Below it, there are two rectangular input fields. The first field is labeled "USERNAME" and the second is labeled "PASSWORD". At the bottom of the form, there is a rounded rectangular button labeled "MASUK".

Gambar 3.7 Rancangan *Login*

yang ada pada *interface*, sedangkan pengguna dengan hak akses petugas dapat mengakses semua *interface* yang ada di dalam aplikasi tapi tidak memiliki akses untuk melakukan *add*, *delete* dan *edit* pada setiap table yang ada pada *interface*, petugas hanya diberikan akses untuk melekuakan *monitoring* pada *dashboard* dan melihat *table* yang ada.

3.6.1 Rancangan *Interface Login*

Perancangan *interface* untuk *menu login* dapat dilihat pada Gambar 3.7. *form login* adalah *menu* yang ditemukan pertama kali sebelum pengguna dapat mengakses aplikasi. Untuk dapat melakukan *login* pengguna baru harus meminta pada admin untuk dibuatkan akses agar dapat melakukan *login*. Jadi, penambahan pengguna hanya dapat dilakukan oleh pengguna yang memiliki akses sebagai

admin. Setelah terdaftar maka masukan *username* dan *password* pada *form login* yang nantinya akan mengantarkan ke dalam aplikasi.

3.6.2 Rancangan *Interface Dashboard*

Rancangan *interface dashboard* terdiri dari rancangan *widget* dari setiap sensor, disini terdapat 3 *widget* sensor yaitu *sensor pulse*, *weight scale sensor* dan *sensor temperature*. Untuk *widget sensor temperature* tampilannya adalah memperlihatkan nilai suhu yang paling terbaru, untuk *widget weight scale sensor* dikarenakan fungsi utamanya adalah untuk melihat berat dari infus agar diketahui infus yang harus digantikan maka dibuat sebuah animasi tabung yang jika *weight* berkurang, isi tabung juga akan berkurang. Untuk *widget* pada *sensor heart rate* menggunakan nilai *heart rate* yang paling terakhir didapatkan. Dalam *dashboard* juga terdapat informasi mengenai ruangan, nama pasien, foto pasien dan alat. rancangan *dashboard* dapat dilihat pada Gambar 3.8.

3.6.3 Rancangan *Interface User*

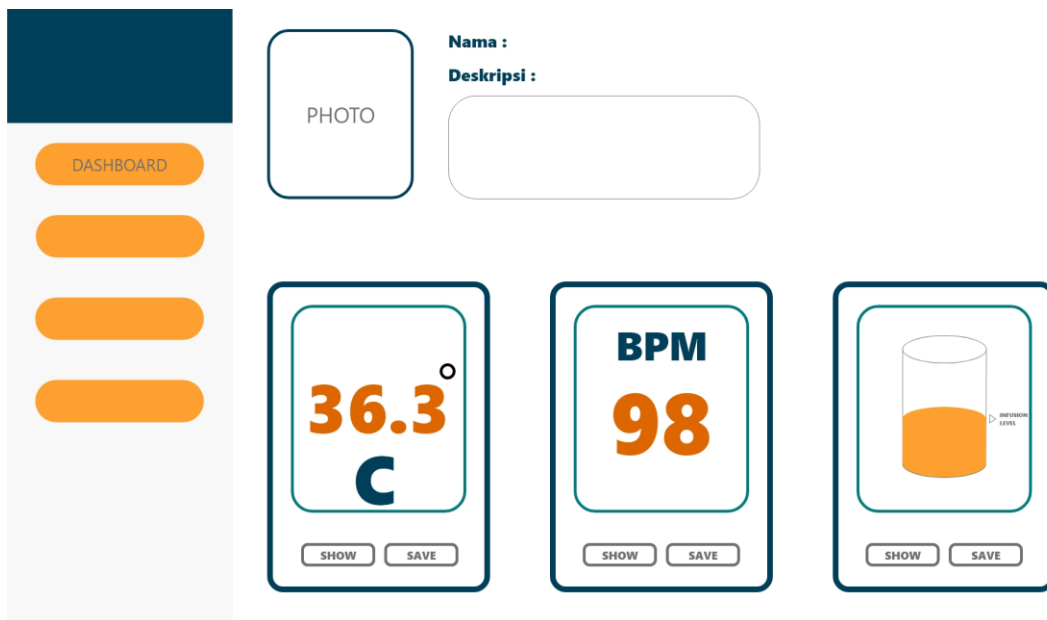
Rancangan *interface User* adalah Rancangan yang dibuat berdasarkan *table user* pada *database*, fungsi dari *interface user* ini adalah untuk menampilkan *table user* agar pada *interface* ini dapat dilihat *user* yang memiliki hak akses sebagai admin. Fungsi lainnya pada *interface* ini adalah *table user* dapat langsung dirubah pada *interface* ini, dirubah yang dimaksud disini adalah fungsi *add*, *delete* dan *edit* dapat dilakukan langsung di dalam *interface user*. Rancangan *interface user* dapat dilihat pada Gambar 3.9.

3.6.4 Rancangan *Interface Petugas*

Rancangan *interface* Petugas adalah Rancangan yang dibuat berdasarkan *table* Petugas pada *database*, fungsi dari *interface* Petugas ini adalah untuk menampilkan *table* Petugas agar pada *interface* ini dapat dilihat *user* yang memiliki hak akses sebagai Petugas. Fungsi lainnya pada *interface* ini adalah *table* Petugas dapat langsung dirubah pada *interface* ini, dirubah yang dimaksud di sini adalah fungsi *add*, *delete* dan *edit* dapat dilakukan langsung di dalam *interface* Petugas. Rancangan *interface* Petugas dapat dilihat pada Gambar 3.10.

3.6.5 Rancangan *Interface Ruangan*

Rancangan *Interface* Ruangan adalah Rancangan yang dibuat berdasarkan *table* ruangan pada *database*, fungsi dari *interface* ini adalah untuk menampilkan *table* ruangan pada *interface*. Pada *interface* ini difungsikan untuk menambahkan nama, nomor ruang dan nama pasien diruang tersebut yang nantinya akan otomatis muncul pada *dashboard*. Fungsi *interface* ini adalah untuk menambahkan, mengganti dan mengurangi jumlah ruang atau nama pasien yang ada di dalam ruangan. Rancangan *interface* ruangan dapat dilihat pada Gambar 3.11.



Gambar 3.8 Rancangan *Dashboard*

Table User

ADD

NAME	USERNAME
admin	admin
Beryl Sitorus	berylsitorus

Form Tambah user

Harap isi isian di bawah ini:

Name *

Username *

Password *

Gambar 3.9 Rancangan *Interface User*

Table Petugas

ADD

Show entries

NAME	USERNAME	EMAIL	NO TELP
petugas 1	petugas	petugas@email.com	08123456789

Harap isi isian di bawah ini:

Petugas Name *

Petugas Email *

Petugas No Telp *

Username *

Password *

Gambar 3.10 Rancangan *Interface* Petugas

Table Ruangan

ADD

Show entries

NO	RUANGAN NAME
1	Mawar
2	melati

Form Tambah Ruangan

Harap isi isian di bawah ini:

Ruangan Name *

Ruangan No *

Pasient Name

Pasient Photo No file chosen

Gambar 3.11 Rancangan *Interface* Ruangan

3.6.6 Rancangan Interface Tugas

Rancangan *Interface* Tugas adalah rancangan yang dibuat berdasarkan *table* *tbl_tugas* yang di dalamnya ditambahkan perintah untuk melakukan *add*, *delete* dan *edit* untuk dapat merubah isi dari *table* *tbl_tugas* langsung pada aplikasinya. Isi dari *tbl_tugas* adalah data mengenai no ruangan dan data petugas yang bertanggung jawab pada ruangan tersebut. Rancangan *Interface* Tugas dapat dilihat pada Gambar 3.12.

3.6.7 Rancangan Interface Alat

Rancangan *Interface* Alat adalah rancangan yang dibuat berdasarkan *database* dari *table* *tbl_alat*, rancangan *interface* di dalamnya berupa isi dari data yang ada di *table* *tbl_alat* yaitu code alat, nomer ruangan, *IP* dan berat infus. Fungsi dari data *IP* adalah untuk mengetahui *IP* alat yang terhubung dengan aplikasi dan untuk berat infus adalah data berat *maximum* pada infus agar animasi tabung yang ada pada *dashboard* dapat dijalankan, fungsi lainnya adalah untuk mengetahui keberadaan alat karena terdapat data nomer ruangan untuk mengetahui ruang, di mana alat ini diletakkan. Rancangan *Interface* Alat dapat dilihat pada Gambar 3.13.

3.6.8 Rancangan Interface Laporan

Rancangan *Interface* Alat adalah rancangan yang dibuat berdasarkan *database* dari *table* *tbl_histori*, *interface* ini difungsikan agar *user* dapat membuat sebuah laporan otomatis berbentuk *file* pdf yang dapat menyimpan data pasien setiap harinya. Dalam *interface* ini terdapat *datetime* yang difungsikan untuk memilih hari dan tanggal *monitoring* yang ingin dibuat laporannya jadi, dapat memudahkan petugas maupun dokter untuk melihat histori data *monitoring* dari pasien pasien. Rancangan *Interface* Laporan dapat dilihat pada Gambar 3.14.

Table Tugas

ADD

Show 10 entries

NO	PETUGAS NAMA	RUANGAN NO	ACTION
1	petugas 1	101	EDIT DELETE

Showing 1 to 1 of 1 entries

Form Tambah Tugas

Harap isi isian di bawah ini:

Petugas * PETUGAS 1

Ruangan No * 101/MAWAR

ADD RESET

Gambar 3.12 Rancangan *Interface* Tugas

Table Alat

Show entries

NO	ALAT CODE	RUANGAN NO	IP	BERAT INFUS
1	asd	101	12412431353	1000
2	MWR	102	395239457234	1000

Showing 1 to 2 of 2 entries

Form Tambah Alat

Harap isi isian di bawah ini:

Alat Code *

Ruangan No *

IP *

Berat Infus *

Gambar 3.13 Rancangan *Interface* Alat

Form Report Alat

Harap isi isian di bawah ini:

Tanggal *

Ruangan No *

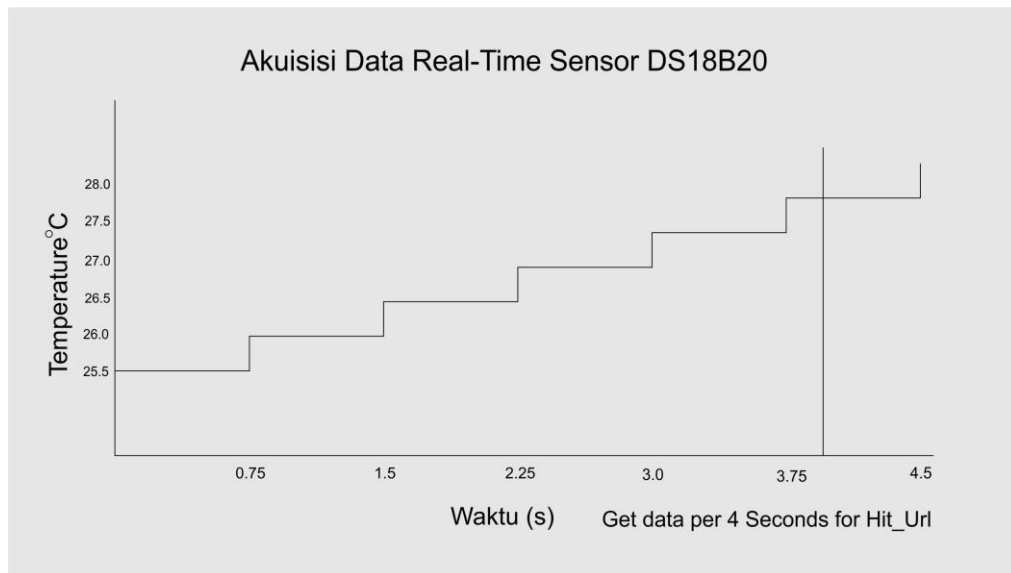
Gambar 3.14 Rancangan *Interface* Laporan

3.7 Aspek *Real-Time* Sensor

untuk aspek *real-time* pada sensor, setiap sensor yaitu *weight scale sensor*, *temperature sensor* dan *pulse sensor* memiliki akuisisi data yang berbeda-beda. Untuk mengetahui akuisisi data *real-time* setiap sensor yang digunakan yaitu *weight scale sensor*, *temperature sensor* dan *pulse sensor* dapat dilihat dari jumlah data yang diterima dalam satuan waktu. Untuk delay pada setiap pengiriman data dari client ke server menggunakan delay dengan waktu 4 seconds.

3.7.1 Akuisisi Data *Temperature Sensor*

Temperature sensor yang digunakan adalah modul DS18B20 yang merupakan *temperature sensor* yang metode akuisisi datanya menggunakan *onewire* yang sering digunakan dalam sebuah *wireless sensor networks* dan *internet of things (IoT) application*. Modul *temperature* ini mengirimkan data digital 12 bit ke mikrokontroler dengan *conversion time* 750 ms. Akuisisi data *temperature sensor* ini dapat dilihat pada Gambar 3.15.

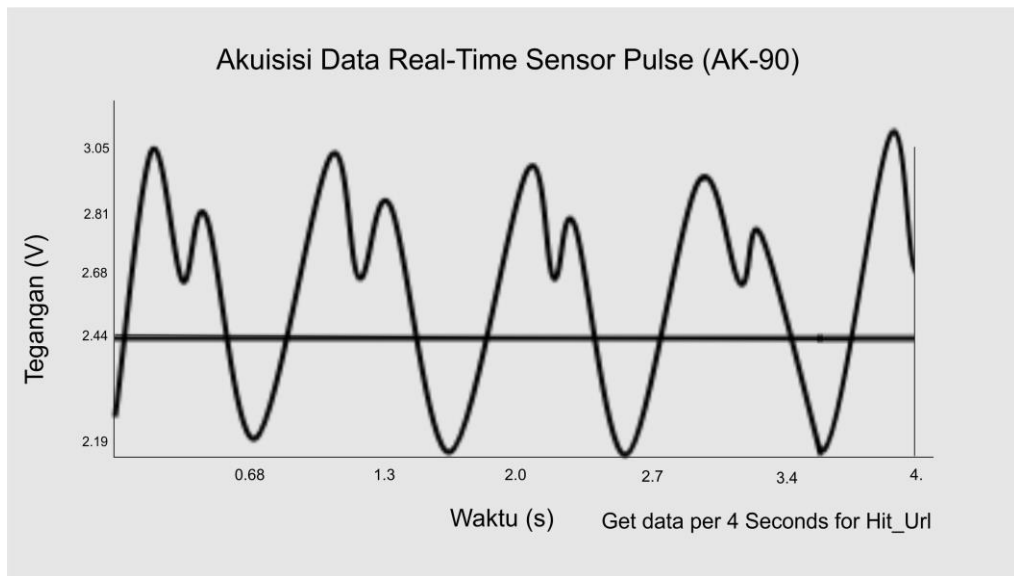


Gambar 3.15 Akuisisi Data *Real-Time* Sensor DS18B20

Seperti yang dilihat pada Gambar 3.15 *temperature sensor* mendapatkan satu data dalam waktu 0.75 s, ini dikarenakan *conversion time* yang dibutuhkan untuk merubah data digital menjadi data *temperature* pada mikrokontroler membutuhkan waktu 750ms. Dengan begitu dalam waktu 4 detik pada saat *hit_url* meminta data sensor, data yang diambil oleh *hit_url* adalah akuisisi data *real-time* yang ke-4.

3.7.2 Akuisisi Data *Pulse Sensor*

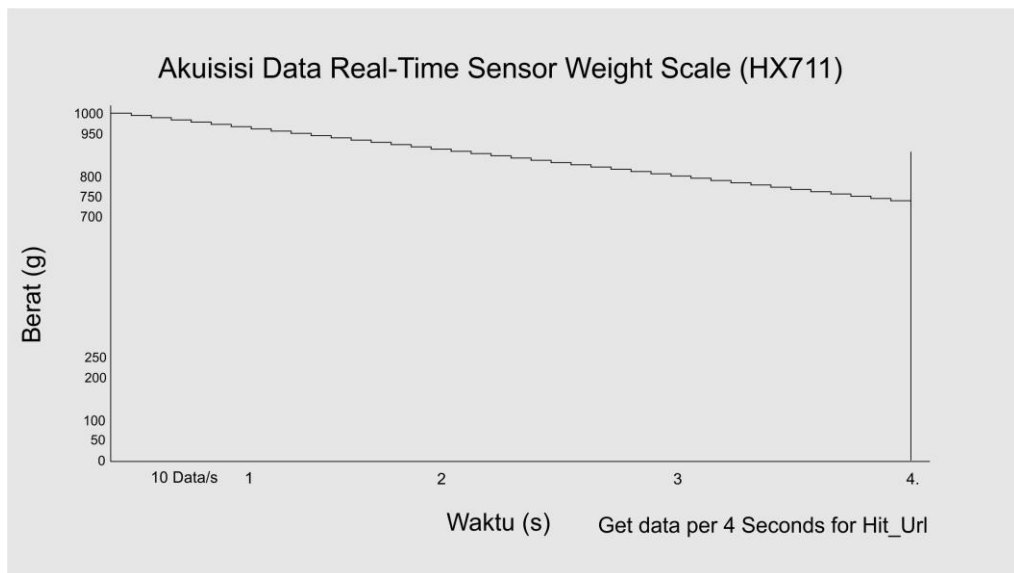
Pulse sensor yang digunakan adalah modul AK-90 yang merupakan *pulse sensor* dengan menggunakan metode *photoplethysmography* (PPG). Metode ini menangkap *volume* dari darah pada *vessel*. Metode ini menggunakan *green light* untuk mengetahui *volume* darah pada *vessel*, penggunaan *green light* dikarenakan darah menyerap sebagian *green light* dengan begitu *volume* darah pada *vessel* dapat diketahui, nantinya dapat dikonversi menjadi data *beats per minute* (BPM). Modul *pulse sensor* ini adalah modul sensor analog yang tegangannya sebesar 5 volt yang berasal dari mikrokontroler. Jadi, sensor ini memiliki *output* maksimal 5 volt, biasanya *output* pada sensor ini berada disetengah tegangan maksimal pada 2,5 volt yang diubah ke bentuk digital. Dikarenakan pin *analog* yang digunakan pada mikrokontroler adalah 10 bit jadi, *range* nilai dari 0 sampai 1023. Nilai tersebut didapat dari hasil 2 pangkat 10. Dari hal tersebut dapat diketahui nilai tegangan 2,5 volt memiliki nilai digital 512. Nilai digital tersebut dikonversi oleh mikrokontroler yang nantinya akan berbentuk sebuah data *volume* darah pada *vessel*. Dapat dilihat pada Gambar 3.16 akuisisi data *real-time* dari *pulse sensor*.



Gambar 3.16 Akuisisi Data *Real-Time* Sensor Pulse (AK-90)

3.7.3 Akuisisi Data *Weight Scale Sensor*

Weight scale sensor yang digunakan adalah *loadcell* yang dihubungkan dengan modul HX711, HX711 merupakan *precision* ADC dengan analog to digital converter 24 bit. Penerimaan data pada sensor ini sebanyak 10 data per detik. Pada Gambar 3.17 dibuat sebuah gambaran akuisisi data *real-time weight scale sensor*.



Gambar 3.17 Akuisisi Data *Real-Time* Weight Scale (HX711)

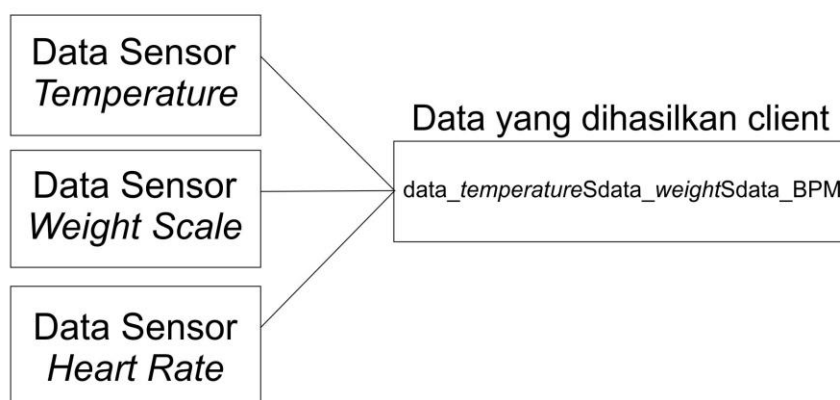
Dari Gambar 3.17 di atas dapat diketahui bahwa nilai *update* dari sensor adalah 10 Hz yang berarti data diakuisisi sebanyak 10 kali dalam 1 detik. Jadi, data yang akan diambil oleh *hit_url* adalah data *real-time* dari *weight scale sensor* yang ke-40.

3.7.4 Akuisisi Data pada Aplikasi

Dalam melakukan akuisisi data untuk dikirimkan ke aplikasi web yang harus diperhatikan adalah data yang dikirimkan harus secara bersamaan dikarenakan jumlah sensor yang digunakan lebih dari satu, yaitu data dari *temperature sensor*, data dari *weight scale sensor*, dan data dari *pulse sensor*.

Pengiriman data secara *continuous* tersebut dilakukan dengan cara mengirimkan setiap data dari sensor secara bersamaan, pengiriman data dari tiga sensor yang ada secara serentak dimasukkan dalam satu *hit url* agar data dari tiga sensor tersebut dapat diterima secara bersamaan pada *server*.

Dalam melakukan pengiriman data dari client ke server terdapat delay sebesar 4 detik yang membuat data pada *client* dikirimkan ke *server* setiap 4 detik. Untuk pengiriman data dari tiga sensor secara bersamaan dilakukan dengan menggabungkan setiap data dari sensor pada satu *hit_url*. Setiap data sensor diberikan batasan berupa huruf s yang digunakan sebagai pemisah pada saat data diterima oleh *server*. Pemisah tersebut berfungsi pada saat data sampai pada server, server dapat langsung memiliki ketiga data sensor secara bersamaan dengan begitu data dari ketiga sensor dapat ditampilkan secara bersamaan pada web aplikasi yang ada. Data yang dihasilkan oleh *client* dapat dilihat pada Gambar 3.18.



Gambar 3.18 Rancangan Pengiriman Dari *Client*

Dari Gambar 3.18 dapat dilihat bahwa data yang dikirimkan oleh tiga sensor sudah dapat disatukan menjadi sebuah data yang di dalamnya terdapat data dari ketiga sensor tersebut. Dengan begitu data dari ketiga sensor dapat diterima oleh server secara bersamaan yang bertujuan untuk merealisasikan data yang *real-time* pada ketiga sensor yang digunakan. Dengan sistem seperti ini server dapat menerima ketiga data secara bersamaan dan database pada server akan langsung memiliki ketiga data sensor tersebut dalam waktu yang bersamaan.

Dengan sistem rancangan seperti ini aspek *real-time* dapat terealisasi pada *monitoring* yang memiliki parameter lebih dari satu, karena setiap data yang dikirimkan oleh *client* sudah mengandung semua data sensor yang ingin dikirimkan. Pentingnya data dari setiap sensor diterima oleh *server* secara bersamaan adalah agar pada saat masuk ke *database* aplikasi data dari ketiga sensor tersebut memiliki tanggal dan waktu yang sama.

3.8 Perancangan *Database*

System database di sini adalah *system database* yang menggunakan basis penggunaan *foreign key* dan *primary key*. *System database* ini adalah model *database* MySQL yang dibagi fungsinya pada setiap *table* yang dibuat.

Perancangan *database* adalah proses untuk menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan sistem. Tujuan perancangan *database* adalah untuk memenuhi informasi yang berisikan kebutuhan-kebutuhan *user* secara khusus dan aplikasi – aplikasinya, memudahkan pengertian struktur informasi dan mendukung kebutuhan-kebutuhan pemrosesan dan beberapa obyek penampilan. Struktur data adalah cara menyimpan atau merepresentasikan data di dalam komputer agar bisa dipakai secara efisien. *Database* yang akan dirancang adalah *database* untuk menyimpan data parameter.

Berdasarkan *system* yang telah dibuat terdapat 6 tabel yang digunakan pada *system wireless monitoring* ini. Perancangan *database* berfokus pada *interface* yang dibuat dalam aplikasi *monitoring* dan fungsionalitas *system*. *Table* yang akan diimplementasikan adalah *tbl_user*, *tbl_ruangan*, *tbl_alat*, *tbl_petugas*, *tbl_histori* dan *tbl_tugas*.

Table *tbl_user* memuat informasi mengenai data dari *user* yang memiliki tingkatan admin pada aplikasi dan digunakan untuk data pengguna yang dapat login pada aplikasi. Tingkatan admin adalah *user* yang memiliki akses ke seluruh bagian dari aplikasi. *Table* *tbl_ruangan* memuat informasi mengenai data ruangan, di dalam *table* ruangan terdapat data mengenai nomer ruang, nama ruang dan nama pasien. *Table* *tbl_petugas* memuat informasi mengenai pengguna yang merupakan petugas atau perawat yang bertanggung jawab dengan pasien yang sedang dirawat disebuah ruangan dan *table* ini berfungsi untuk membuat akses login dengan tingkatan petugas yang hanya dapat melakukan *monitoring* pada aplikasi. *Table* *tbl_histori* memuat informasi setiap data *monitoring* yang nantinya difungsikan

untuk membuat laporan otomatis yang dapat memudahkan rumah sakit. *Table* *tbl_alat* memuat informasi mikrokontroler di dalam *table* ini data yang dimasukkan adalah id alat, nama alat, *IP address* alat dan id_ruangan. *Table* *tbl_tugas* memuat informasi berupa nama petugas dan nomer ruangan yang fungsinya untuk mengetahui siapa petugas yang bertanggung jawab atas ruangan tersebut. Rincian informasi mengenai *table* yaitu tipe data setiap barisnya akan ditunjukkan oleh Tabel 3.1 sampai 3.6.

Tabel 3.1 Rancangan *Table* *tbl_user*

<i>Field</i>	<i>Tipe</i>	Keterangan
User_id	Int(11)	User id pengguna (primary key)
name	Varchar(255)	Nama lengkap pengguna
Username	Varchar(255)	Username pengguna untuk login
password	Varchar(255)	Password pengguna untuk login

Tabel 3.2 Rancangan *Table* *tbl_petugas*

<i>Field</i>	<i>Tipe</i>	Keterangan
Petugas_id	Int(11)	Petugas id bagi petugas (primary key)
Petugas_name	Varchar(255)	Nama lengkap petugas
Petugas_email	Varchar(255)	Email petugas
Petugas_no_telp	Varchar(255)	Nomor telepon petugas
Username	Varchar(255)	Username petugas untuk login
password	Varchar(255)	Password petugas untuk login

Tabel 3.3 Rancangan *Table* *tbl_ruangan*

<i>Field</i>	<i>Tipe</i>	Keterangan
Ruangan_id	Int(11)	Ruangan id (primary key)
Ruangan_name	Varchar(255)	Nama pada ruangan
Ruangan_no	Varchar(255)	Nomer pada ruangan
Patient_name	Varchar(255)	Nama pasien pada ruangan
Patient_foto	Varchar(255)	Foto pasien

Notif_status	Tinyint(4)	Trigger untuk berat infus
---------------------	------------	---------------------------

Tabel 3.4 Rancangan *Table* tbl_alat

<i>Field</i>	<i>Tipe</i>	Keterangan
Alat_id	Int(11)	Nomer id untuk alat (<i>primary key</i>)
Alat_code	Varchar(255)	Code untuk alat atau <i>hardware</i>
Id_ruangan	Varchar(255)	Ruangan id (<i>foreign key</i>)
IP	Varchar(255)	Nomer <i>IP address</i> pada <i>hardware</i>
Berat_infus	Varchar(255)	nilai maksimum dari berat infus

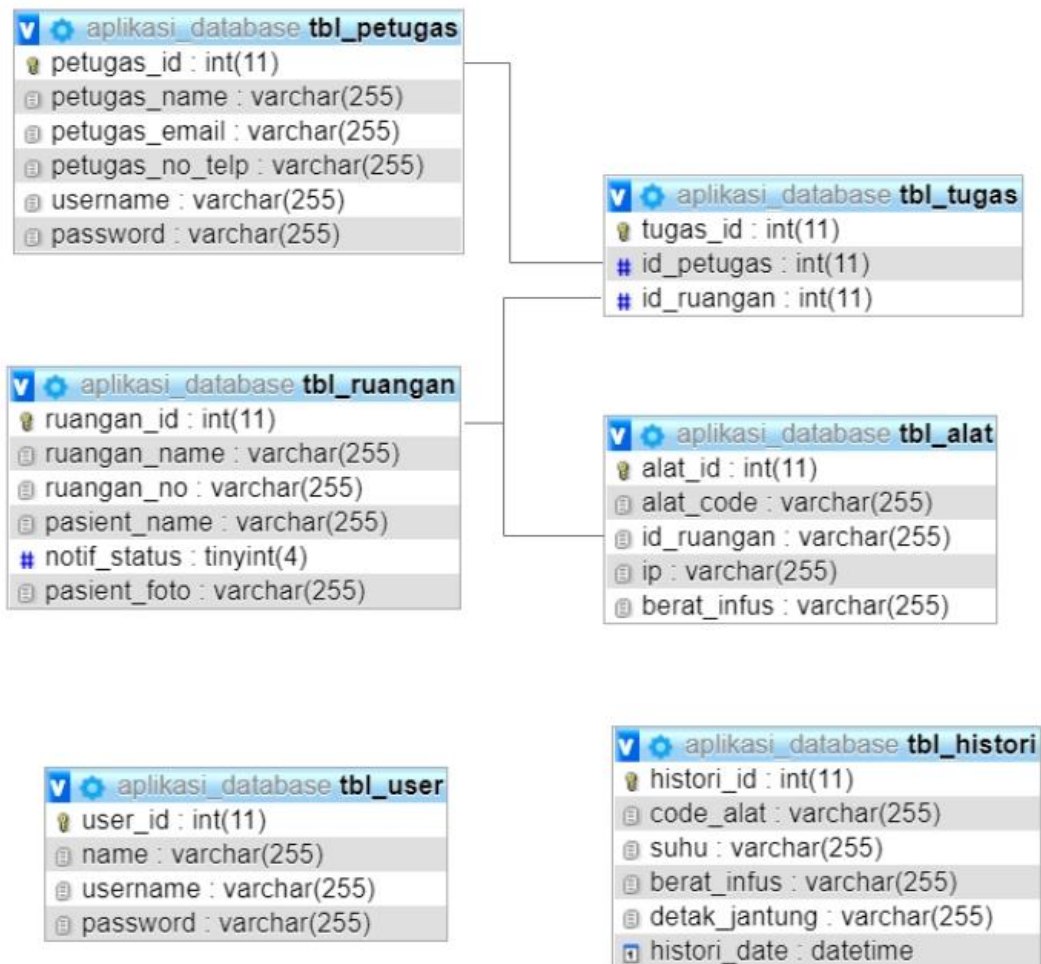
Tabel 3.5 Rancangan *Table* tbl_tugas

<i>Field</i>	<i>Tipe</i>	Keterangan
Tugas_id	Int(11)	Tugas number id (<i>primary key</i>)
Id_petugas	Int(11)	Id petugas (<i>foreign key</i>)
Id_ruangan	Int(11)	Id ruangan (<i>foreign key</i>)

Tabel 3.6 Rancangan *Table* tbl_histori

<i>Field</i>	<i>Tipe</i>	Keterangan
Histori_id	Int(11)	Id number untuk histori (<i>primary key</i>)
Code_alat	Varchar(255)	Alat code (<i>foreign key</i>)
Suhu	Varchar(255)	Data suhu tubuh
Berat_infus	Varchar(255)	Data berat infus
Detak_jantung	Varchar(255)	Data <i>heart rate</i>
Histori_date	datetime	Tanggalan untuk pemilihan waktu

Relasi antar *table* ditunjukkan oleh Gambar 3.3. relasi antar *table* digunakan untuk mengetahui *table* tbl_ruangan berelasi *one to many* dengan *table* tbl_tugas dan *table* tbl_alat. Terdapat relasi juga terhadap *table* tbl_petugas dimana petugas_id adalah *primary key* yang memiliki relasi dengan *table* tbl_tugas.



Gambar 3.19 Relasi Antar Tabel

3.8 Pengujian

Pada tahap ini dilakukan pengujian kerja sistem yang telah dibuat, pengujian ini dilakukan agar dapat menunjukkan bahwa perangkat lunak telah mampu bekerja sesuai dengan fungsi dan kebutuhannya pengujian yang dilakukan antara lain adalah:

- 1) Pengujian terhadap kualitas aplikasi yakni proses pengujian perangkat lunak dari segi fungsionalitas. Pengujian ini dilakukan untuk mengetahui apakah aplikasi dapat melakukan perintah sesuai dengan kebutuhan, untuk mengetahui bahwa aplikasi telah sesuai dengan kebutuhan pengujian ini aplikasi ini dilakukan berdasarkan *use case* yang ada.
- 2) Pengujian koneksi yakni proses pengujian perangkat lunak dari segi komunikasi. Pengujian ini dilakukan untuk mengetahui apakah koneksi antara mikrokontroler dengan wifi sudah tersambung. Pada tahap ini pengujian koneksi dilakukan pada aplikasi Arduino monitor untuk mengetahui bahwa mikrokontroler sudah mengirimkan data sensor ke server.
- 3) Pengujian pada pembuatan sebuah form laporan yang berisikan mengenai data *monitoring* berupa berat infus, *heart rate*, *temperature* dari pasien selama 5 menit sekali. Laporan dibuat untuk melihat histori *monitoring* pada pasien dalam satu hari penuh.
- 4) Pengujian *relation* antara *database* yang digunakan di dalam aplikasi, pengujian ini dilakukan untuk mengetahui apakah *relation primary key* dan *foreign key* sudah terhubung. Pengujian ini dilakukan dengan melakukan *add* pada *database* yang memiliki *relation*.

BAB 4 PENGUJIAN SISTEM

Pada bab ini akan dilakukan pengujian kerja sistem aplikasi yang telah dibuat agar dapat menunjukkan bahwa perangkat lunak sudah dapat melakukan fungsinya sesuai dengan spesifikasi dan kebutuhan. Pengujian pada sistem juga termasuk pada pengujian di dalam akuisisi data dari sensor ke mikrokontroler. Pengujian yang dilakukan antara lain adalah pengujian *black-box*, pengujian koneksi jaringan, pengujian *relation database*, pengujian akuisisi data pada mikrokontroler, pengujian pada *design form* laporan. Pengujian *black-box* adalah proses pengujian untuk aplikasi yang dibangun apakah sudah dapat melakukan fungsi masukan atau keluaran yang sesuai dengan spesifikasi dan kebutuhan yang diminta. Pada pengujian *black-box* didasarkan pada desain *use-case*.

Pengujian koneksi jaringan ditujukan untuk menguji apakah data yang dikirim oleh mikrokontroler telah berhasil dikirimkan secara wireless dan dapat diterima oleh komputer, bukan hanya itu saja data tersebut juga dapat ditampilkan pada aplikasi. Pengujian ini didasarkan oleh informasi dari *serial monitor* yang dihasilkan dan membandingkan kedua data yang ada pada *serial monitor* dan data yang ada pada aplikasi. Pengujian *relation database* ditujukan untuk mengetahui apakah hubungan antar *database* sudah berjalan dengan baik, pengujian ini dilakukan dengan cara meng-*input* data pada tabel *database* yang memiliki relasi dengan tabel *database* lainnya. Pengujian *form* laporan ditujukan untuk menguji apakah laporan yang dibuat secara otomatis memiliki isi yang sesuai dengan apa yang dibutuhkan, cara untuk melakukan pengujian tersebut adalah dengan membandingkan hasil dari data yang ada pada aplikasi dengan *form* laporan yang sudah dibuat secara otomatis dengan begitu dapat diketahui kebenaran dari laporan *monitoring* tersebut.

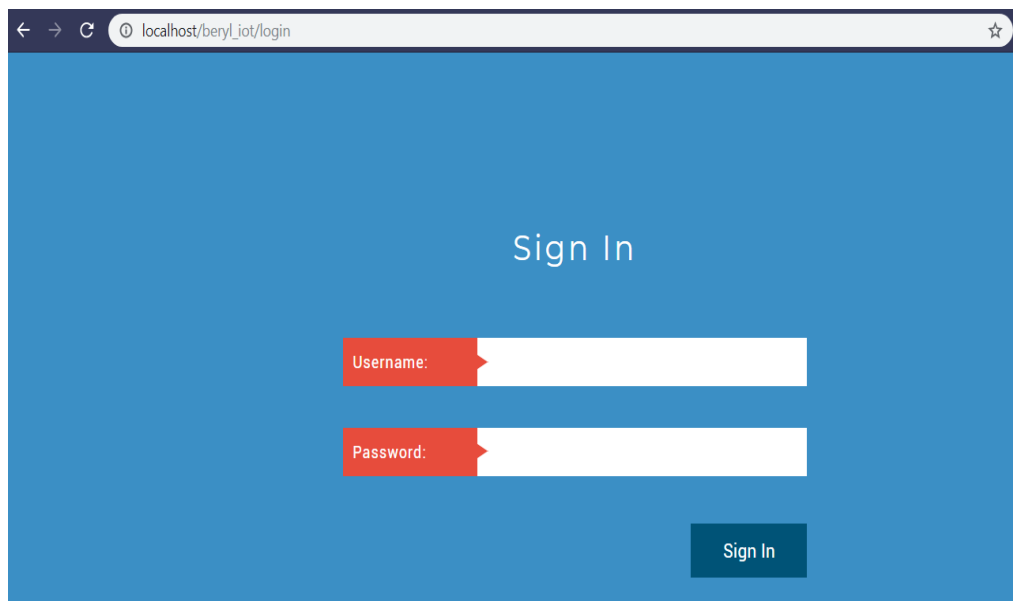
4.1 Pengujian Black Box

Dalam melakukan pengujian *black box* terdapat 6 *use case* yang diujikan pada bagian pengujian ini, dalam melakukan pengujian terhadap *use case* pertama yang dilakukan adalah memasukan data pada *system use case* tertentu dan selanjutnya akan dilihat hasil atau respon dari sistem terhadap *input* yang diberikan. Apabila hasil respon sesuai dengan fungsionalitas *use case* maka dapat disimpulkan bahwa hasil dari pengujian sesuai.

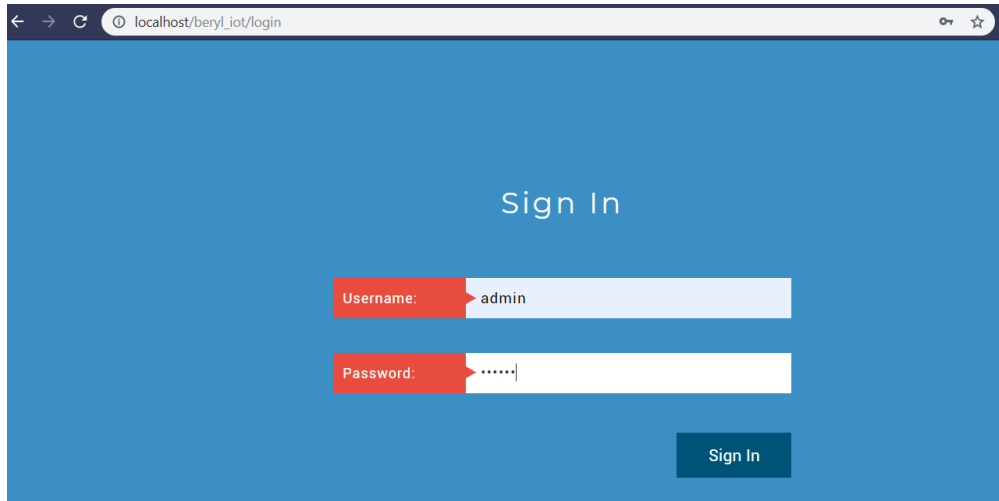
4.1.1 Use Case Login

Pada pengujian *use case login*, hal yang pertama dilakukan adalah masuk kedalam *website* dari aplikasi yang sudah dibuat disini menggunakan URL `localhost/beryl_iots`. Setelah memasukan URL tersebut seharusnya halaman pertama yang muncul adalah *menu login* seperti yang ditunjukkan oleh Gambar 4.1.

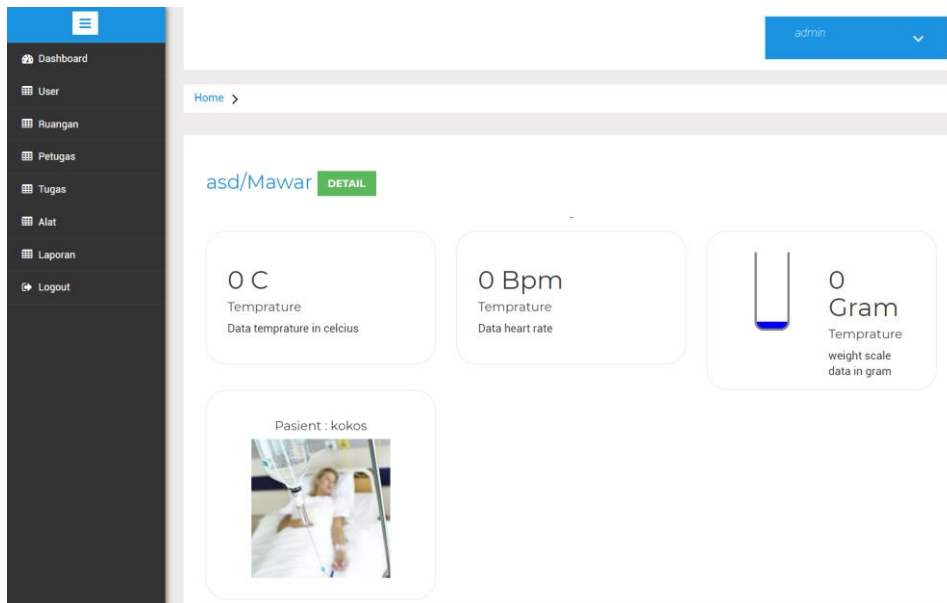
untuk pengujian pertama adalah masuk dengan akun yang memiliki hak akses admin seperti pada Gambar 4.2 yang hasilnya setelah mengisi *username* dan *password* seharusnya langsung masuk kepada *menu dashboard* seperti pada Gambar 4.3. Pengujian *login* berikutnya menggunakan akun yang memiliki hak akses petugas seperti pada Gambar 4.4. Setelah memasukkan *username* dan *password* seharusnya langsung masuk di dalam *menu dashboard* seperti pada Gambar 4.5 untuk membedakan hak akses dilihat dari Gambar 4.3 untuk hak akses admin *menu* yang dapat dipilih berupa *dashboard*, *user*, ruangan, petugas, tugas, alat, laporan dan *logout* tapi untuk hak akses petugas dapat dilihat pada Gambar 4.5 bahwa *menu* yang ada hanya *dashboard*, ruangan, tugas, alat, laporan dan *logout* dengan ini dapat diketahui bahwa aplikasi *use case login* sudah berjalan sesuai dengan kebutuhan.



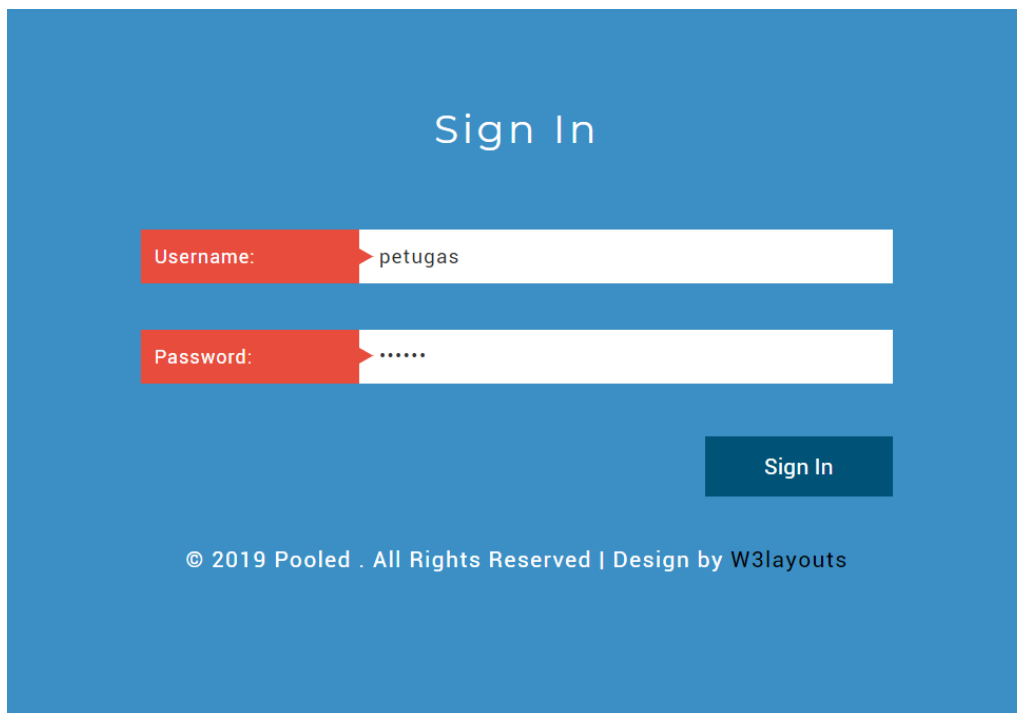
Gambar 4.1 Halaman *Login*



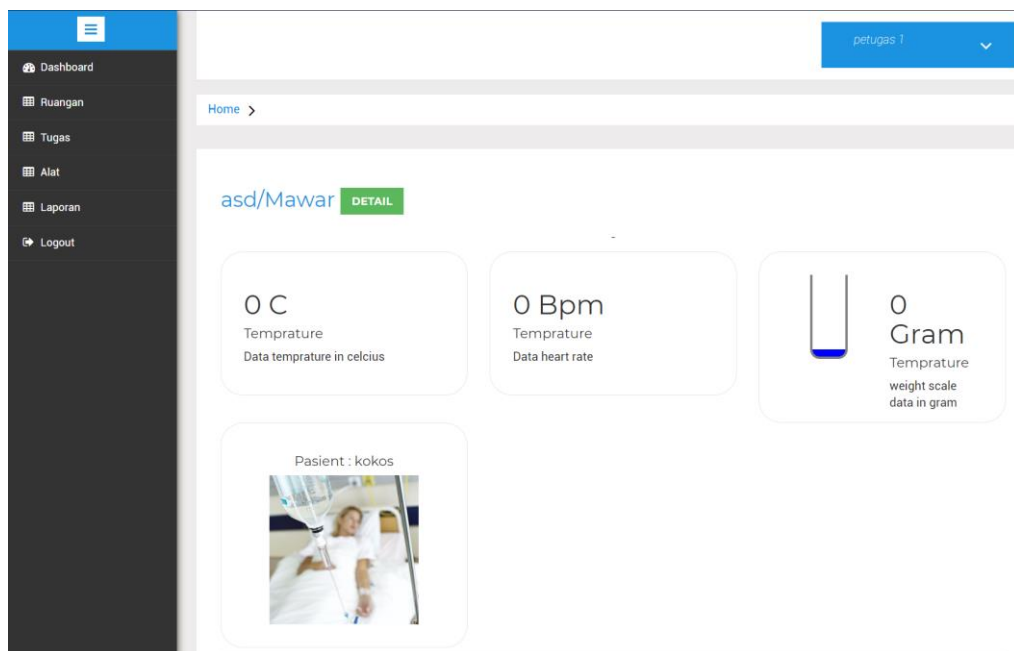
Gambar 4.2 *Login Admin*



Gambar 4.3 *Menu Dashboard*



Gambar 4.4 *Login Petugas*



Gambar 4.5 *Menu Dashboard*

4.1.2 Use Case User

Pada Pengujian *use case user* hal yang dilakukan adalah melakukan penambahan data akun pada *database tbl_user*. Dalam melakukan penambahan akun hal perbandingan yang akan diperlihatkan adalah tabel *database user* sebelum ditambahkan data *user* baru pada Gambar 4.6, setelah itu akan dilakukan *input* data *user* baru pada tabel *database user* pada Gambar 4.7 setelah ditambahkan *user* baru maka akan dilihat apakah data *user* bertambah pada *database user* dapat dilihat pada Gambar 4.8.

NAME	USERNAME
admin	admin
Beryl Sitorus	berylsitorus

Gambar 4.6 *Database Awal User*

Form Tambah user

Harap isi isian di bawah ini:

Name *

Username *

Password *

Gambar 4.7 *Input Data User*

Berhasil menambahkan data

ADD

NAME	USERNAME
admin	admin
Beryl Sitorus	berylsitorus
adam admin	adminadam

Gambar 4.8 Database User Terbaru

4.1.3 Use Case Ruangan

Pada pengujian *use case* ruangan hal yang dilakukan adalah melakukan penambahan jumlah ruangan yang ada, untuk membuktikan bahwa ruangan sudah berhasil ditambahkan pada *database* maka dilakukan perbandingan isi *database* sebelum dan sesudah melakukan *input* penambahan ruangan. *Database* sebelum *input* data dapat dilihat pada Gambar 4.9. untuk proses *input* data ruangan yang baru dapat dilihat pada Gambar 4.10 dan *database* setelah *input* data dilakukan dapat dilihat pada Gambar 4.11.

NO	RUANGAN NAME	RUANGAN NO	PASIENT NAME
1	Mawar	101	kokos
2	melati	102	dames

Gambar 4.9 Database Ruangan Awal

Form Tambah Ruangan

Harap isi isian di bawah ini:

Ruangan Name *	<input type="text" value="Teratai"/>
Ruangan No *	<input type="text" value="103"/>
Pasient Name	<input type="text" value="Jessica"/>
Pasient Photo	<input type="button" value="Choose File"/> x1.jpg.pages...gDexwbv.jpg
	<input type="button" value="✓ ADD"/> <input type="button" value="↻ RESET"/>

Gambar 4.10 *Input Data Ruangan*

Berhasil menambahkan data

Show entries

NO	RUANGAN NAME	RUANGAN NO	PASIENT NAME
1	Mawar	101	kokos
2	melati	102	dames
3	Teratai	103	Jessica

Gambar 4.11 *Database Ruangan Terbaru*

4.1.4 Use Case Petugas

Pada Pengujian use case petugas hal yang dilakukan adalah melakukan penambahan data akun pada database *tbl_petugas*. Dalam melakukan penambahan akun hal pembanding yang akan diperlihatkan adalah tabel database petugas sebelum ditambahkan data *user* baru pada Gambar 4.12. Setelah itu akan dilakukan *input* data *user* baru pada tabel database petugas pada Gambar 4.13. Setelah ditambahkan *user* baru maka akan dilihat apakah data *user* bertambah pada database petugas dapat dilihat pada Gambar 4.14.

NAME	USERNAME	EMAIL	NO TELP
petugas 1	petugas	petugas@email.com	08123456789

Showing 1 to 1 of 1 entries

Gambar 4.12 Database Petugas Awal

Form Tambah Petugas

Harap isi isian di bawah ini:

Petugas Name *

Petugas Email *

Petugas No Telp *

Username *

Password *

Gambar 4.13 Form Input Petugas

Berhasil menambahkan data

Show entries

NAME	USERNAME	EMAIL	NO TELP
petugas 1	petugas	petugas@email.com	08123456789
petugas2	petugas2	petugas2@gmail.com	081245786541

Gambar 4.14 Update Database Petugas

4.1.5 Use Case Tugas

Pada pengujian *use case* tugas hal yang dilakukan adalah melakukan penggabungan petugas dengan ruang yang pasien di dalamnya adalah tanggung jawabnya. Dalam menambahkan tabel *tbl_tugas* hal pertama yang harus dipilih adalah akun petugas setelah itu memilih nomer ruang yang di dalamnya merupakan pasien yang berhubungan dengan tugas petugas tersebut. *Database* awal dari *tbl_tugas* dapat dilihat pada Gambar 4.15. penambahan pada *database* petugas dengan mengisi *form* tugas dapat dilihat pada Gambar 4.16. hasil penambahan *database* dapat dilihat pada Gambar 4.17.

Show entries

NO	PETUGAS NAMA	RUANGAN NO
1	petugas 1	101

Gambar 4.15 *Database* Tugas Awal

Form Tambah Tugas

Harap isi isian di bawah ini:

Petugas *

Ruangan No *

Gambar 4.16 *Form Input* Tugas

Berhasil menambahkan data

ADD

Show 10 entries

NO	PETUGAS NAMA	RUANGAN NO
1	petugas 1	101
2	petugas2	102

Gambar 4.17 Database Tugas Terbaru

4.1.6 Use Case Alat

Pada pengujian *use case* alat hal yang dilakukan dalam pengujian ini adalah menguji *database* alat dengan cara melakukan penambahan unit alat pada *database* yang ada. Data *database* alat awal dapat dilihat pada Gambar 4.18. langkah awal dalam pengujian ini adalah melakukan *add* unit dengan mengisi *form* penambahan unit yang ada pada aplikasi, di dalam *form* tersebut yang harus diisi adalah *code* alat, nomer ruangan, *IP* dan berat infus. Untuk *code* alat merupakan penomoran atau penamaan yang dilakukan untuk membedakan alat satu dengan alat yang lainnya. Untuk nomor ruangan adalah hal yang dilakukan untuk mengetahui alat tersebut berada pada ruang yang mana. Untuk *IP* merupakan *IP address* dari *hardware* yang terhubung dengan *database* alat dan yang terakhir adalah berat infus, fungsi dari berat infus adalah untuk mengetahui nilai maksimal dari berat infus pada animasi tabung infus pada *dashboard* dengan *input* berat infus, animasi tabung dapat mengetahui berat maksimal pada kondisi penuh. Pengisian *form* penambahan alat dapat dilihat pada Gambar 4.19. untuk hasil akhir *database* setelah penambahan unit alat dapat dilihat pada *database* alat telah berhasil ditambah unit alat yang baru, dapat dilihat pada Gambar 4.20.

NO	ALAT CODE	RUANGAN NO	IP	BERAT INFUS
1	asd	101	12412431353	1000
2	MWR	102	395239457234	1000

Gambar 4.18 Database Alat Awal

Form Tambah Alat

Harap isi isian di bawah ini:

Alat Code *

Ruangan No *

IP *

Berat Infus *

Gambar 4.19 Form Tambah Alat

Table Alat

Berhasil menambahkan data

ADD

Show entries

NO	ALAT CODE	RUANGAN NO	IP	BERAT INFUS
1	MWR	102	395239457234	1000
2	asd	101	12412431353	1000
3	MLT	103	1654135438	1000

Gambar 4.20 Database Alat Terbaru

4.2 Pengujian Laporan Otomatis

Pengujian laporan otomatis merupakan pengujian yang dilakukan pada fungsi aplikasi *database* *tbl_histori*, fungsi dari *tbl_histori* adalah menyimpan data sensor yang ditangkap. Data sensor yang disimpan pada *tbl_histori* adalah data berat infus, data *heart rate* dan data *temperature*. Dari situ dibuatlah sebuah fungsi untuk membuat laporan dalam bentuk *file* pdf yang berguna untuk menyimpan histori data sensor di dalam *file* agar dapat memudahkan *user* atau dokter untuk melihat kondisi pasien yang sedang dirawat. Pengujian ini dimulai dari pembuatan *form* laporan pada aplikasi, di dalam *form* tersebut terdapat pilihan tanggal dan tahun yang difungsikan untuk memilih pada tanggal berapa data yang harus dimasukkan ke dalam laporan, setelah itu pemilihan nomer ruang dibuat untuk mengetahui data alat yang dilaporkan di dalam laporan agar pada laporan diketahui data yang disimpan adalah data dari alat dengan nomor ruangnya. Pengisian *form* untuk pembuatan laporan otomatis dapat dilihat pada Gambar 4.21. hasil laporan yang telah dibuat dapat dilihat pada Gambar 4.22 dan pembandingan dari data laporan terdapat pada Gambar 4.23.

Form Report Alat

Harap isi isian di bawah ini:

Tanggal *

Ruangan No *

✓ LIHAT

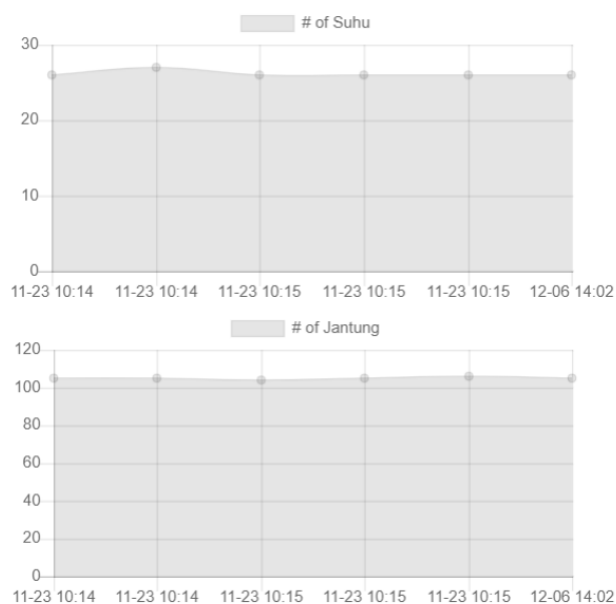
Gambar 4.21 *Form Report Alat*

Tanggal 2019-11-23

No	Alat	Ruangan	Suhu	Berat	Detak Jantung
1	MWR	Mawar	26 C	1000 Gr	105 Bpm
2	MWR	Mawar	27 C	1000 Gr	105 Bpm
3	MWR	Mawar	26 C	999 Gr	104 Bpm
4	MWR	Mawar	26 C	999 Gr	105 Bpm
5	MWR	Mawar	26 C	998 Gr	106 Bpm
6	MWR	Mawar	26 C	998 Gr	105 Bpm

Gambar 4.22 Laporan

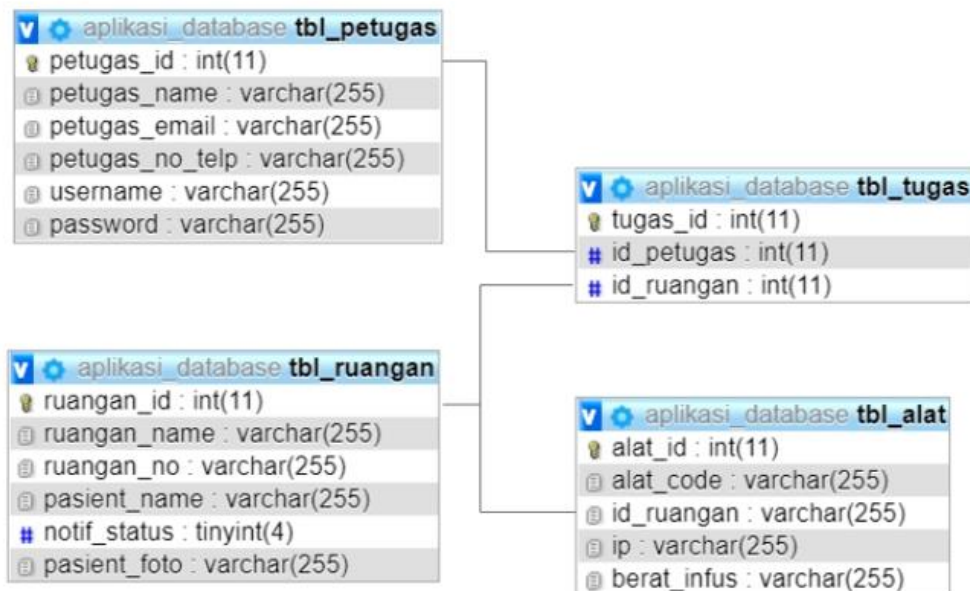
Dapat dilihat pada Gambar 4.21. Laporan yang dibuat sudah lengkap dengan tabel yang berisikan *code* alat, ruangan, data suhu, data berat dan data detak jantung atau *heart rate*. Data tersebut semuanya diambil pada tanggal tertentu yang diperlihatkan di atas laporan pada Gambar 4.21 tanggal data pada laporan adalah pada tanggal 23 November 2019. Data tersebut akan dibandingkan dengan data yang ada di dalam aplikasi untuk menjadi pembanding kebenaran dari data yang ada pada laporan. Sebagai pembanding laporan dapat dilihat pada Gambar 4.22.



Gambar 4.23 Data Sensor

4.3 Pengujian *Relation Database*

Dalam melakukan pengujian *relation database* terdapat beberapa hal yang harus diperhatikan yaitu pada saat penambahan data pada *database* yang memiliki *relation* dengan *database* lain terdapat *form input* yang diisi dengan pilihan dari data yang diambil dari *database* yang berhubungan dengan *database* tersebut. Maka dari itu langkah pertama dalam pengujian ini adalah mengetahui *database* mana yang memiliki *relation* dengan *database* lainnya, hal tersebut dapat dilihat pada Gambar 4.24 jika dilihat *database* yang memiliki *relation* adalah *database* ruangan yang memiliki *relation* dengan *database* tugas dan *database* alat, setelah itu *database* petugas juga memiliki *relation* dengan *database* tugas. Dilihat dari garis *relation* database tugas memiliki *relation* dengan dua *database* yang mewajibkan melakukan pengisian terhadap dua *database* yang memiliki *relation* dengan *database* tugas agar *database* tugas dapat ditambahkan.



Gambar 4.24 *Relation Database*

4.3.1 Relation Database Tugas

Dalam melakukan pengujian relasi *database* tugas dengan *database* lain dilakukan dengan cara melihat *input* data pada saat mengisi *form* dari *database* tugas. Jika dilihat dari Gambar 4.24 diketahui bahwa *database* tugas memiliki relasi dengan *database* ruangan dan *database* petugas maka dari itu sebelum mengisi *database* tugas seharusnya melakukan pengisian data terlebih dahulu pada *database* ruangan dan *database* petugas dikarenakan dalam *form* tugas yang dimasukan adalah data ruangan dan petugas dari kedua *database* yang memiliki relasi tersebut. *Form* pengisian *database* tugas dapat dilihat pada Gambar 4.25. *Input* data dari *form* tersebut berasal dari isi *database* ruangan dapat dilihat pada Gambar 4.26 dan isi dari *database* petugas yang dapat dilihat pada Gambar 4.27.

Form Tambah Tugas

Harap isi isian di bawah ini:

Petugas *

Ruangan No *

- 101/MAWAR
- 102/MELATI
- 103/TERATAI

Gambar 4.25 Form Database Tugas

NO	RUANGAN NAME	RUANGAN NO	PASIENT NAME
1	Mawar	101	kokos
2	melati	102	dames
3	Teratai	103	Jessica

Gambar 4.26 Database Ruangan

NAME	USERNAME	EMAIL	NO TELP
petugas 1	petugas	petugas@email.com	08123456789
petugas2	petugas2	petugas2@gmail.com	081245786541

Gambar 4.27 Database Petugas

4.3.2 Relation Database Alat

Dalam melakukan pengujian Relasi *database* alat dengan *database* lain, dilakukan dengan cara mengisi *input form* pada *database* alat. Jika dilihat pada Gambar 4.24 dapat diketahui bahwa relasi yang dimiliki oleh *database* alat adalah dengan *database* ruangan. Maka sebelum melakukan pengisian *database* alat, *database* ruangan sudah harus memiliki data pada tabel *database* tersebut agar *database* alat dapat ditambahkan. *Form* pengisian data pada *database* alat dapat dilihat pada Gambar 4.28. *form* pengisian *database* alat pada bagian ruangan pengisiannya berupa isi data dari *database* ruangan maka dari itu *database* alat memiliki relasi dengan *database* ruangan. *Database* ruangan dapat dilihat pada Gambar 4.29.

Form Tambah Alat

Harap isi isian di bawah ini:

Alat Code *

Ruangan No *

101/MAWAR

102/MELATI

103/TERATAI

IP *

Berat Infus *

✓ ADD
↺ RESET

Gambar 4.28 Form Tambah Alat

Table Ruangan

ADD

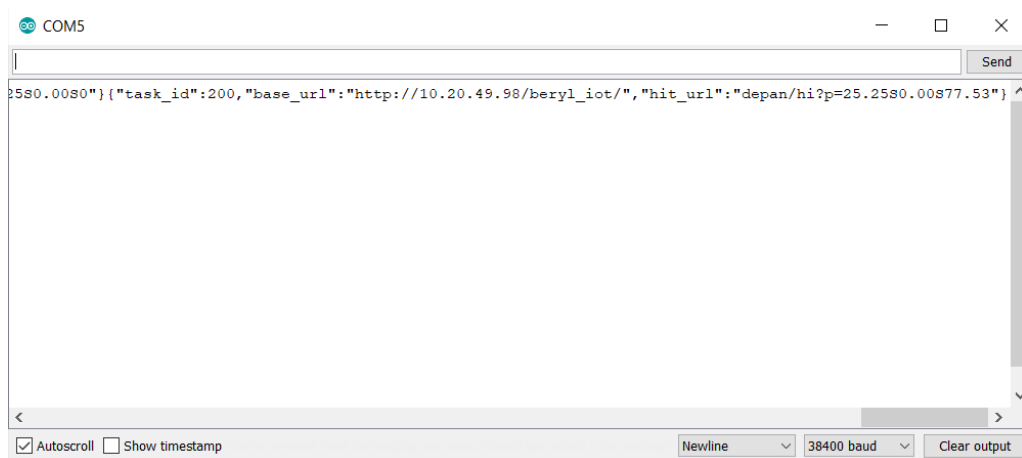
Show entries

NO	RUANGAN NAME	RUANGAN NO	PASIENT NAME
1	Mawar	101	kokos
2	melati	102	dames
3	Teratai	103	Jessica

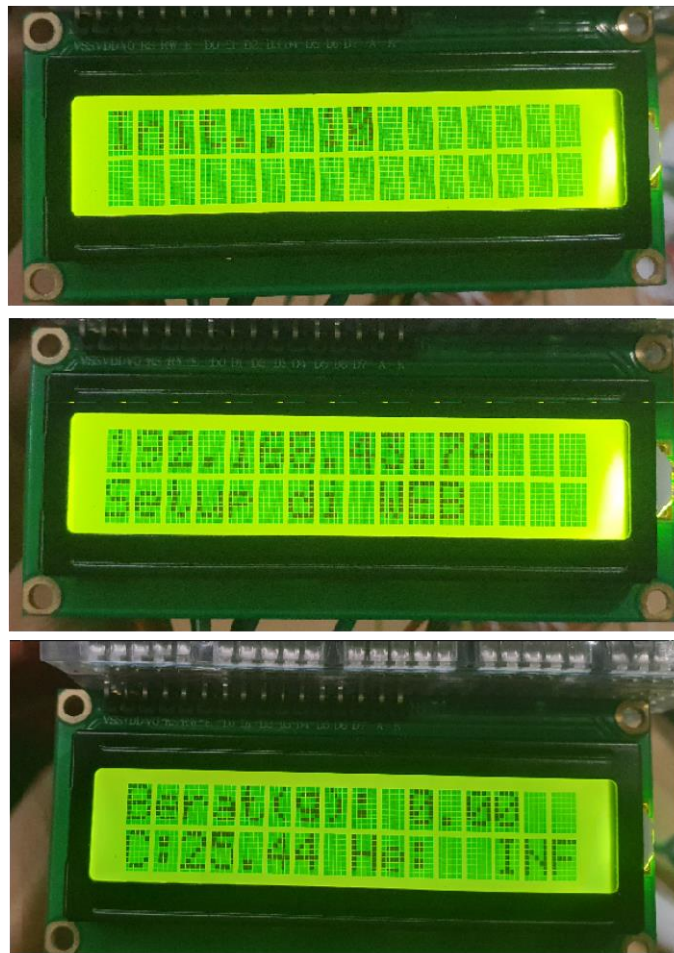
Gambar 4.29 Database Ruangan

4.4 Pengujian Komunikasi Sistem

pengujian yang dilakukan adalah melakukan koneksi dari awal hingga data dapat diterima oleh aplikasi, hal pertama yang dilakukan adalah melakukan *setting* nama wifi dan *password* wifi yang dimasukan pada mikrokontroler untuk mengkonfigurasi ESP agar dapat langsung terkoneksi dengan wifi yang sesuai dengan keinginan. Dapat dilihat pada Gambar 4.31 yang merupakan proses inisiasi koneksi terhadap wifi, menginfokan *IP Address* dan pada akhirnya menampilkan hasil dari data sensor pada *LCD display*.



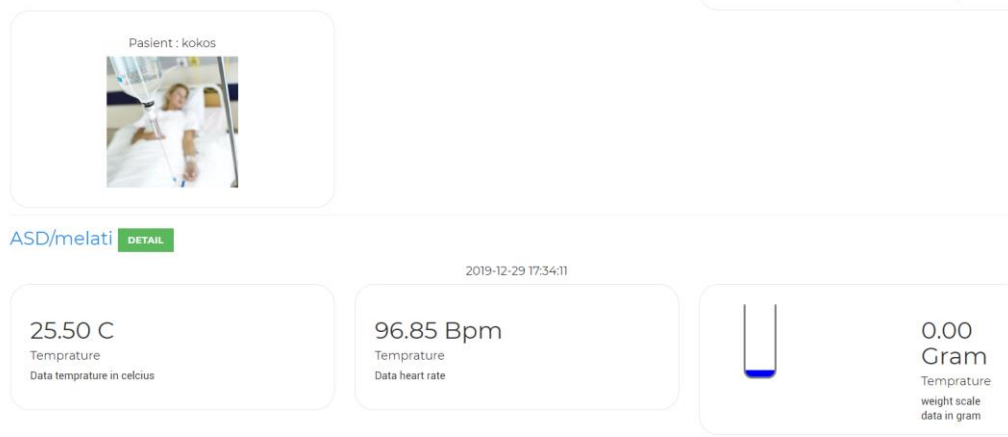
Gambar 4.30 Serial Monitor Arduino



Gambar 4.31 Inisialisasi Komunikasi *Client-Server*

	histori_id	id_alat	suhu	berat_infus	detak_jantung	histori_date	ip
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	251	4	25.50	0.00	96.85	2019-12-29 17:34:11	192.168.43.74
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	250	4	25.50	0.00	96.85	2019-12-29 17:34:10	192.168.43.74
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	249	4	25.50	0.00	96.85	2019-12-29 17:34:08	192.168.43.74
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	248	4	25.50	0.00	96.85	2019-12-29 17:34:06	192.168.43.74
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	247	4	25.50	0.00	96.85	2019-12-29 17:34:05	192.168.43.74
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	246	4	25.50	0.00	0	2019-12-29 17:34:02	192.168.43.74
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	245	4	25.44	0.00	0	2019-12-29 17:34:00	192.168.43.74
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	244	4	25.50	0.00	0	2019-12-29 17:33:59	192.168.43.74
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	243	4	25.50	0.00	0	2019-12-29 17:33:57	192.168.43.74

Gambar 4.32 *Update Database*



Gambar 4.33 *Dashboard* Aplikasi

Setelah tahap *client* mengirim data kepada *server* yang dapat dibuktikan pada Gambar 4.30, dimana pada Gambar 4.30 *serial monitor* arduino memperlihatkan proses pengiriman data dari mikrokontroler ke *server* dan pada Gambar 4.32, dapat dilihat bahwa tabel histori bertambah setelah mikrokontroler mengirimkan data ke *server*. Pada Gambar 4.33 merupakan hasil data yang ditampilkan pada aplikasi sebagai pembandingan bahwa *database* yang tersimpan dengan tampilan data yang ada pada aplikasi sama adanya.

BAB 5 PEMBAHASAN

5.1 Pengujian *Black Box*

Pada pengujian *black box* pengujian yang dilakukan adalah pengujian pada *use case* pada setiap bagian dari aplikasi dimana *use case* yang digunakan sebagai pengujian disini berjumlah 6 *use case*. 6 *use case* itu adalah *use case login*, *use case user*, *use case ruangan*, *use case petugas*, *use case tugas*, *use case alat*. Pada bab sebelumnya setiap *use case* diuji dengan diperlihatkan gambar *capture layer* pada setiap pengujian dengan kondisi cara pengujian masing-masing. Hasil yang diperlihatkan pada gambar pengujian menunjukkan respon aplikasi terhadap perintah yang dilakukan oleh *user* atau pengguna. *Table 5.1* sampai *5.6* merupakan hasil dari pengujian terhadap setiap *use case* yang diujikan.

Table 5.1 Use Case Login

Pengujian	Respon	Kesimpulan
<i>Login (Admin)</i>	Masuk kedalam aplikasi sebagai Admin	Sesuai
<i>Login (Petugas)</i>	Masuk kedalam aplikasi sebagai Petugas	Sesuai

Table 5.2 Use Case User

Pengujian	Respon	Kesimpulan
<i>User (database)</i>	Halaman aplikasi memperlihatkan <i>database user</i>	Sesuai
<i>User (add form)</i>	<i>Form</i> pengisian <i>user database</i>	Sesuai
<i>User (update)</i>	<i>Database ter-update</i> setelah <i>add form</i>	Sesuai

Table 5.3 Use Case Ruangan

Pengujian	Respon	Kesimpulan
<i>Ruangan (database)</i>	Halaman aplikasi memperlihatkan <i>database Ruangan</i>	Sesuai
<i>Ruangan (add form)</i>	<i>Form</i> pengisian <i>ruangan database</i>	Sesuai
<i>Ruangan (update)</i>	<i>Database ter-update</i> setelah <i>add form</i>	Sesuai

Table 5.4 Use Case Petugas

Pengujian	Respon	Kesimpulan
Petugas (database)	Halaman aplikasi memperlihatkan <i>database</i> petugas	Sesuai
Petugas (add form)	<i>Form</i> pengisian petugas <i>database</i>	Sesuai
Petugas (update)	<i>Database</i> ter-update setelah <i>add form</i>	Sesuai

Table 5.5 Use Case Tugas

Pengujian	Respon	Kesimpulan
Tugas (database)	Halaman aplikasi memperlihatkan <i>database</i> tugas	Sesuai
Tugas (add form)	<i>Form</i> pengisian tugas <i>database</i>	Sesuai
Tugas (update)	<i>Database</i> ter-update setelah <i>add form</i>	Sesuai

Table 5.6 Use Case Alat

Pengujian	Respon	Kesimpulan
Alat (database)	Halaman aplikasi memperlihatkan <i>database</i> alat	Sesuai
Alat (add form)	<i>Form</i> pengisian alat <i>database</i>	Sesuai
Alat (update)	<i>Database</i> ter-update setelah <i>add form</i>	Sesuai

Berdasarkan keseluruhan hasil pengujian dapat ditarik kesimpulan bahwa perangkat lunak bebas dari kesalahan dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.

5.2 Pengujian Laporan Otomatis

Pada pengujian laporan otomatis yang dilakukan adalah pengujian kebenaran dari laporan yang dibuat dengan membandingkan laporan dan hasil data sensor yang ada pada *dashboard* aplikasi. Pengujian berikutnya adalah pengujian pembuatan laporan sesuai dengan tanggal yang diinginkan jadi, laporan yang dibuat

harus memiliki data yang sesuai dengan tanggal yang diminta, untuk menyesuaikan dapat disamakan dengan data yang diterima oleh aplikasi pada *dashboard*. Pengujian di sini juga dilihat bahwa apakah tabel histori menyimpan data yang tepat karena jika data sensor tidak sesuai maka hasil dari laporan akan tidak sama dengan *dashboard* yang berada pada aplikasi.

Table 5.7 Laporan Otomatis

Pengujian	Respon	Kesimpulan
Laporan (database)	Aplikasi terhubung dengan <i>database</i> histori	Sesuai
Laporan (add form)	<i>Form</i> pengisian tanggal dan nomer ruangan Laporan	Sesuai
Laporan (create)	Laporan berhasil dibuat dengan otomatis	Sesuai

Pada saat melakukan pembuatan laporan untuk memudahkan *user* mendapatkan data monitoring yang dibutuhkan, hal yang harus diperhatikan adalah ketepatan data yang dimiliki oleh laporan tersebut. Jika hasil yang ditunjukkan oleh laporan tidak sesuai dengan data *monitoring* yang sebenarnya akan menyebabkan kesalahan dalam mengambil keputusan tindakan. Maka dari itu dibuat pengujian laporan otomatis agar dipastikan bahwa laporan yang dibuat memiliki isi data yang sesuai dengan data yang ditangkap oleh sensor.

5.3 Pengujian *Relation Database*

Pada pengujian *relation database* yang dilakukan adalah pengujian hubungan antara *database* dengan *database* yang lain. Isi pengujian ini adalah pembuktian bahwa terdapat tabel *database* yang memiliki relasi dengan *database* lain. Diketahui bahwa *database* yang memiliki relasi adalah *database* tugas dan *database* alat. Penggunaan relasi digunakan untuk pengisian data agar penyimpanan data dapat optimal dan tidak ada data yang terduplikat, relasi *database* ini juga dilakukan untuk mengurangi *error* pada aplikasi yang disebabkan oleh penambahan data pada *database*.

Table 5.8 Relation Database

Pengujian	Respon	Kesimpulan
Database Tugas (Form Relation)	<i>Form input database</i> tugas pada data <i>relation</i> sudah dapat menampilkan pilihan.	Sesuai
Database Alat (Form Relation)	<i>Form input database</i> alat pada data <i>relation</i> sudah dapat menampilkan pilihan.	Sesuai

5.4 Pengujian Komunikasi Sistem

Pada pengujian komunikasi sistem yang dilakukan adalah menghubungkan koneksi dari inisiasi awal hingga data dapat diterima oleh aplikasi. Isi pengujian ini adalah pembuktian bahwa komunikasi antara *client* dan *server* telah berhasil dan data yang dikirimkan oleh *client* dapat diterima oleh *server* dengan baik. Pada pengujian ini juga membuktikan bahwa *database* *tb_histori* bertambah pada saat data dari *client* diterima oleh *server*.

Table 5.9 Komunikasi Sistem

Pengujian	Respon	Kesimpulan
Pengiriman data (client – server)	Pengiriman data dari <i>client</i> sudah dapat diterima dengan tepat oleh <i>server</i> .	Sesuai
Database (Data Sensor)	Data yang diterima oleh <i>server</i> sudah berhasil masuk dan tersimpan pada database.	Sesuai

BAB 6 KESIMPULAN DAN SARAN

6.1 Kesimpulan

Pada penelitian tugas akhir ini dirancang sebuah sistem *wireless monitoring* dengan aplikasi berbasis web dan *database* medis. Berdasarkan hasil penelitian, *multiple wireless monitoring* berhasil diimplementasikan dalam sistem berbasis web dengan menggunakan PHP dan database yang dibuat di dalam MySQL. Dalam tugas akhir ini aplikasi berhasil mengimplementasikan level akses dalam aplikasi yang dibedakan menjadi dua *level* akses yaitu akses admin dan akses perawat, dapat melakukan fungsinya sesuai dengan *use case* yang sudah diujikan yaitu *user* dapat melakukan *update*, *add*, *edit* dan *delete* pada setiap tabel yang disediakan pada aplikasi. User juga dapat melakukan pembuatan laporan *monitoring* dalam bentuk pdf secara otomatis sesuai dengan hari yang ditentukan.

Pada penelitian tugas akhir ini juga berhasil mengimplementasikan sistem pengiriman data secara *wireless* yang berbasis *TCP/IP* dan setiap data yang dikirimkan oleh *wireless sensor network* dapat diterima oleh *server* dan disimpan pada *database* tanpa ada satupun data yang hilang atau memiliki *success rate* pada transfer data sebesar 100 % dan dapat membuat *report* harian secara otomatis dalam bentuk *file* pdf yang berisikan data dari setiap sensor *monitoring*.

6.2 Saran

Saran untuk penelitian ini adalah Penggunaan *hardware* dengan *filtering* yang lebih baik agar aplikasi menangkap data yang memiliki nilai *error* kecil atau dapat diabaikan dan penambahan *system software* untuk melakukan *decisions support systems*. Penambahan sistem *security* pada pengiriman data secara *wireless* agar data yang dikirim dapat lebih *secure*, penambahan sistem *security* dapat berupa enkripsi pada data yang dikirimkan atau penggunaan protokol seperti MQTT atau HTTPS yang mempunyai sistem *security* data yang lebih baik.

DAFTAR PUSTAKA

- [1] M. Elliott, "Critical care: The eight vital signs of patient monitoring," *British journal of nursing*, vol. 21 no 10, p. 621, 2012.
- [2] Nindithia Putri Windryani, " Comparison Analysis Between MQTT and HTTP Protocol in lot Platform," e-Proceeding of Engineering, Vol.6, No.2 Agustus 2019.
- [3] H. S.Ahmed, "Wireless Sensor Network for Medical," in *2014 International conference*, iraq, 2015.
- [4] A. S. A. F. S. Nilanjan Dey, "Developing Residential Wireless Sensor," *IEEE Transactions on Consumer Electronics*, vol. 63 no 4, p. 432, 2017.
- [5] M. S. J. R. Paulo Neves, "Application of Wireless Sensor Networks to Healthcare," *JOURNAL OF COMMUNICATIONS SOFTWARE AND SYSTEMS*, vol. 4 no 3, p. 181, 2008.
- [6] M.-T. Vo, "wireless sensor network for real time healthcare monitoring: network design and performance evaluation simulation," in *international conference on biomedical engineering in vietnam*, pp, vietnam, 2015.
- [7] J. K. S. I. N. Alsalaet, "Programmable low-power wireless sensor network for vibration monitoring and analysis," *International Journal of Medical Engineering and Informatics*, vol. 5 no 2, pp. 163-176, 2013.

BIOGRAFI PENULIS



Beryl Sitorus adalah seorang mahasiswa Teknik Biomedik, Institut Teknologi Sepuluh Nopember 2015. Ia lahir di Kota Bandung pada tanggal 01 April 1997. Selama perkuliahan penulis aktif dalam beberapa kegiatan kemahasiswaan baik di dalam jurusan maupun di luar jurusan. Penulis juga melakukan penelitian tugas akhir di laboratorium elektronika biomedika.