



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TJ141502**

**PEMANTAUAN *VITAL SIGN* UNTUK PUSAT KESEHATAN  
MASYARAKAT MENGGUNAKAN *SINGLE BOARD  
COMPUTER* DAN MIKROKONTROLER**

Ahmad Aris Habibi  
NRP 2913 100 009

Dosen Pembimbing  
Dr. I Ketut Eddy Purnama, ST., MT.  
Arief Kurniawan, ST., MT.

DEPARTEMEN TEKNIK KOMPUTER  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TJ141502**

**PEMANTAUAN *VITAL SIGN* UNTUK PUSAT  
KESEHATAN MASYARAKAT MENGGUNAKAN *SINGLE  
BOARD COMPUTER* DAN MIKROKONTROLER**

Ahmad Aris Habibi  
NRP 2913 100 009

Dosen Pembimbing  
Dr. I Ketut Eddy Purnama, ST., MT.  
Arief Kurniawan, ST., MT.

DEPARTEMEN TEKNIK KOMPUTER  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017



**FINAL PROJECT - TJ141502**

**VITAL SIGN MONITORING FOR SOCIETY HEALTH  
CENTER USING SINGLE BOARD COMPUTER AND  
MICRO-CONTROLLER**

Ahmad Aris Habibi  
NRP 2913 100 009

Advisor  
Dr. I Ketut Eddy Purnama, ST., MT.  
Arief Kurniawan, ST., MT.

Department of Computer Engineering  
Faculty of Electrical Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2017

# PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Pemantauan *Vital Sign* untuk Pusat Kesehatan Masyarakat Menggunakan *Single Board Computer* dan Mikrokontroler**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2017

Ahmad Aris Habibi

NRP. 2913100009

# LEMBAR PENGESAHAN

Pemantauan *Vital Sign* untuk Pusat Kesehatan Masyarakat  
Menggunakan *Single Board Computer* dan Mikrokontroler

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh  
gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh : Ahmad Aris Habibi (NRP: 2913100009)

Tanggal Ujian : 11 Juli 2017

Periode Wisuda : September 2017

Disetujui oleh:

Dr. I Ketut Eddy Purnama, ST., MT.  
NIP: 196907301995121001

(Pembimbing I)

Arief Kurniawan, ST., MT.  
NIP: 197409072002121001

(Pembimbing II)

Prof. Dr. Ir. Mauridhi Hery P. M.Eng  
NIP: 195809161986011001

(Penguji I)

Dr. Surya Sumpeno, ST., M.Sc.  
NIP: 196906131997021003

(Penguji II)

Muhtadin, ST., M.Sc.  
NIP: 198106092009121003

(Penguji III)

Mengetahui  
Kepala Departemen Teknik Komputer

Dr. I Ketut Eddy Purnama, ST., MT.  
NIP: 196907301995121001

# ABSTRAK

Nama Mahasiswa : Ahmad Aris Habibi  
Judul Tugas Akhir : Pemantauan *Vital Sign* Untuk Pusat Kesehatan Masyarakat Menggunakan *Single Board Computer* dan Mikrokontroler  
Pembimbing : 1. Dr. I Ketut Eddy Purnama, ST., MT.  
2. Arief Kurniawan, ST., MT.

Banyaknya puskesmas (Pusat Kesehatan Masyarakat) yang telah membuka layanan rawat inap, diharuskan puskesmas memiliki peralatan medis yang memadai. Di sisi lain, kondisi pasien yang dirawat di puskesmas merupakan hal yang harus dipantau oleh paramedis secara berkala. Kondisi pasien ditentukan dari hasil analisa data *vital sign*. *Vital sign* seorang pasien terlihat hanya ketika alat pemantau dalam keadaan bekerja. Data yang telah terekam tidak dapat diakses kembali oleh paramedis. Akibatnya dokter hanya dapat menganalisa pasien pada saat itu. Selain itu, terkadang dokter tidak berada di tempat saat ada laporan. Sehingga dibutuhkan sistem pemantauan yang dapat menyimpan data ke *cloud storage* guna laporan *vital sign*. Sistem dibuat menggunakan *single board computer* dan mikrokontroler dan terhubung ke *cloud storage*. Hasil dari sistem yang dibuat menunjukkan bahwa data yang divisualisasi merupakan data valid dari sensor *node*. Rata-rata tingkat akurasi masing-masing sensor jika dipersentase yaitu sensor ECG sebesar 83.47%, sensor suhu sebesar 97.36%, dan sensor nafas sebesar 76.42%. Pada proses penyimpanan data *vital sign* ke *cloud storage* memiliki rata-rata *throughput uplink* dan *downlink* masing-masing sebesar 572.2 *Bit/s* dan 3.54 *kBit/s*. Sistem yang dibuat dapat memantau *vital sign* secara *realtime*. *User interface vital sign* dan laporan dapat berjalan dengan baik. Sehingga dapat disimpulkan bahwa sistem dapat memantau *vital sign* dan memvisualisasikan *vital sign*. Lalu data tersebut disimpan pada *cloud storage* dan dibuat laporan yang dapat diakses oleh dokter.

Kata Kunci : *Vital Sign*, Mikrokontroler, *Single Board Computer*, *Cloud Storage*,

*Halaman ini sengaja dikosongkan*

# ABSTRACT

*Name* : Ahmad Aris Habibi  
*Title* : *Vital Sign Monitoring for Society Health Center Using Single Board Computer and Micro-controller*  
*Advisors* : 1. Dr. I Ketut Eddy Purnama, ST., MT.  
2. Arief Kurniawan, ST., MT.

*The number of puskesmas (Society Health Center) that have opened inpatient services, are required to have adequate medical equipment. On the other hand, the condition of the patients treated at the puskesmas is something that should be monitored by paramedics on a regular basis. The patient's condition is determined from the analysis of vital sign data. Patient's vital sign is seen only when the monitoring device is working. Recorded data can not be re-accessed by paramedics. As a result doctors can only analyze patients at that time. In addition, sometimes doctors are not in place when there are reports. So it takes a monitoring system that can store data to cloud storage for vital sign reports. The system is built using single board computer and micro-controller and connected to cloud storage. The result of system indicates that the visualized data is a valid data from the sensor node. Average accuracy level of each sensor for ECG sensor is 83.47%, temperature sensor is 97.36%, and breath sensor is 76.42%. In the process of storing vital sign data to cloud storage has an average uplink and downlink throughput of 572.2 Bit/s and 3.54 kBit/s. The system can monitor the vital sign in real time. The vital sign user interface and reports can run well. So it can be concluded that the system can monitor the vital sign and visualize the vital sign. Then the data is stored in the cloud storage and made reports that can be accessed by doctors.*

*Keywords* : *Vital Sign, Single Board Computer, Micro-controller, Cloud Storage*

*Halaman ini sengaja dikosongkan*

# KATA PENGANTAR

Puji dan syukur kehadiran Allah SWT atas segala limpahan berkah, rahmat, serta hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul **Pemantauan *Vital Sign* untuk Pusat Kesehatan Masyarakat Menggunakan *Single Board Computer* dan Mikrokontroler.**

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer ITS, Bidang Studi Telematika, serta digunakan sebagai persyaratan menyelesaikan pendidikan S1 Departemen Teknik Komputer ITS. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, khususnya bapak, ibu, dan adik yang telah memberikan dorongan spiritual dan material dalam penyelesaian buku penelitian ini.
2. Bapak Kepala Departemen Teknik Komputer Dr. I Ketut Eddy Purnama, ST., MT.
3. Bapak Dr. I Ketut Eddy Purnama, ST., MT. selaku dosen pembimbing I dan Bapak Arief Kurniawan, ST., MT. selaku dosen pembimbing II atas arahan dan bimbingan selama mengerjakan penelitian.
4. Bapak-ibu dosen pengajar Departemen Teknik Komputer atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis.
5. Guru-guru penulis yang telah mengajarkan ilmunya sehingga penulis dapat studi di tingkat S1 dan menyelesaikan buku penelitian ini.
6. Seluruh teman-teman Lab B401 Laboratorium Komputasi Multimedia yang telah banyak menghibur dan membantu menyelesaikan buku ini.
7. Teman seperjuangan Gafur HZ Bahari dan seluruh teman-teman e-53 yang selalu memberikan semangat penulis dan membantu permasalahan dalam penulisan buku ini.
8. Seluruh orang yang terlibat dalam pengerjaan penelitian ini sehingga proses penelitian berjalan dengan baik.

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, Juli 2017

Penulis

# DAFTAR ISI

<b>ABSTRAK</b>	<b>i</b>
<b>ABSTRACT</b>	<b>iii</b>
<b>KATA PENGANTAR</b>	<b>v</b>
<b>DAFTAR ISI</b>	<b>vii</b>
<b>DAFTAR GAMBAR</b>	<b>ix</b>
<b>DAFTAR TABEL</b>	<b>xi</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Perumusan Masalah . . . . .	2
1.3 Tujuan dan Manfaat . . . . .	3
1.4 Batasan Masalah . . . . .	3
1.5 Sistematika Penulisan . . . . .	3
<b>2 TINJAUAN PUSTAKA</b>	<b>5</b>
2.1 Puskesmas . . . . .	5
2.2 <i>Vital Sign</i> . . . . .	6
2.2.1 Detak Jantung . . . . .	6
2.2.2 Suhu Tubuh . . . . .	7
2.2.3 Jumlah Nafas . . . . .	9
2.2.4 Tekanan Darah . . . . .	9
2.2.5 Saturasi Oksigen . . . . .	10
2.2.6 Rasa Sakit . . . . .	11
2.2.7 Tingkat Kesadaran . . . . .	11
2.2.8 Ekskresi Urin . . . . .	11
2.3 Arsitektur <i>Monitoring E-Health</i> . . . . .	11
2.4 <i>Single Board Computer</i> Raspberry Pi . . . . .	12
2.5 Mikrokontroler Arduino . . . . .	14
2.6 Komunikasi Serial . . . . .	15
2.7 E-Health Shield . . . . .	16
2.7.1 Sensor ECG . . . . .	16

2.7.2	Sensor Hembusan Nafas . . . . .	17
2.7.3	Sensor Tekanan Darah . . . . .	17
2.7.4	Sensor Suhu Tubuh . . . . .	17
2.8	Cloud Storage . . . . .	18
2.9	Protokol HTTP . . . . .	19
<b>3</b>	<b>DESAIN DAN IMPLEMENTASI SISTEM</b>	<b>23</b>
3.1	Desain Sistem . . . . .	23
3.2	Akuisisi dan Transmisi Data pada Sensor <i>Node</i> . . . . .	24
3.3	Implementasi Komunikasi Serial UART . . . . .	26
3.4	Program <i>Receiver</i> dan Penyimpanan Lokal . . . . .	27
3.5	Penyimpanan ke <i>Cloud Server</i> Berbasis Protokol HTTP . . . . .	28
3.6	Desain <i>Database</i> Pada <i>Server</i> . . . . .	30
3.7	<i>User Interface</i> Berbasis Desktop . . . . .	32
3.8	Laporan <i>Vital Sign</i> . . . . .	34
<b>4</b>	<b>PENGUJIAN DAN ANALISA</b>	<b>37</b>
4.1	Pengujian <i>Receiver</i> . . . . .	37
4.2	Validasi Data <i>Vital Sign</i> . . . . .	39
4.3	Perbandingan Sensor . . . . .	42
4.3.1	Sensor ECG . . . . .	42
4.3.2	Sensor Suhu . . . . .	42
4.3.3	Sensor Nafas . . . . .	43
4.3.4	Sensor Tekanan Darah . . . . .	44
4.4	Pengujian Penyimpanan ke <i>Cloud Storage</i> . . . . .	44
4.5	Pengujian Throughput . . . . .	45
4.5.1	<i>Throughput</i> rata-rata . . . . .	46
4.5.2	<i>Throughput</i> maksimal . . . . .	46
4.6	Pengujian <i>User Interface</i> Pemantauan . . . . .	47
<b>5</b>	<b>PENUTUP</b>	<b>53</b>
5.1	Kesimpulan . . . . .	53
	<b>DAFTAR PUSTAKA</b>	<b>55</b>
	<b>LAMPIRAN</b>	<b>59</b>
	<b>BIOGRAFI PENULIS</b>	<b>61</b>

# DAFTAR GAMBAR

2.1	Raspberry Pi 3 Model B <i>pinout</i> . . . . .	13
2.2	Arduino Uno <i>pinout</i> . . . . .	14
3.1	Gambaran umum kerja sistem . . . . .	23
3.2	Rancangan alat dan integrasi sensor . . . . .	24
3.3	<i>Packing</i> alat dan sensor . . . . .	25
3.4	Diagram alir akuisisi data dan transmisi data ke SBC . . . . .	26
3.5	Diagram alir <i>receive</i> data dan penyimpanan data oleh SBC . . . . .	28
3.6	Diagram alir penyimpanan data ke <i>cloud storage</i> . . . . .	29
3.7	HTTP GET <i>request</i> . . . . .	30
3.8	Desain <i>Entity Relation Database</i> . . . . .	31
3.9	Diagram alir visualisasi data . . . . .	32
3.10	Tampilan utama <i>user interface monitoring vital sign</i> . . . . .	33
3.11	Desain <i>user interface</i> laporan . . . . .	34
3.12	Implementasi laporan berbasis web . . . . .	34
3.13	Diagram alir visualisasi laporan . . . . .	35
4.1	Pemasangan sensor-sensor kesehatan . . . . .	38
4.2	Data yang berhasil terkirim dalam selang waktu satu detik . . . . .	39
4.3	Data terkirim pengujian <i>labeling</i> data . . . . .	40
4.4	Perbandingan <i>log</i> data sebelum dikirim dan sesudah terkirim . . . . .	41
4.5	Proses penyimpanan data ke <i>cloud storage</i> . . . . .	45
4.6	Visualisasi <i>vital sign</i> keadaan normal . . . . .	48
4.7	Menu <i>setup</i> identitas pasien . . . . .	49
4.8	Menu <i>set</i> normal <i>vital sign</i> pasien . . . . .	49
4.9	Visualisasi <i>vital sign</i> keadaan abnormal . . . . .	50
4.10	Blok pada elemen <i>user interface</i> yang diujikan . . . . .	51
4.11	Hasil pengujian elemen tampilan utama . . . . .	52

*Halaman ini sengaja dikosongkan*

## DAFTAR TABEL

2.1	Tabel jumlah detak jantung per menit terhadap usia pasien . . . . .	7
2.2	Tabel suhu tubuh terhadap usia pasien . . . . .	8
2.3	Tabel jumlah nafas terhadap usia pasien . . . . .	10
2.4	Tabel tekanan darah terhadap usia pasien . . . . .	10
2.5	Tabel Konstanta $K_1$ dan $K_2$ . . . . .	18
4.1	Tabel pengujian data terkirim terhadap selang waktu	39
4.2	Tabel pengujian ECG . . . . .	42
4.3	Tabel pengujian sensor suhu . . . . .	43
4.4	Tabel pengujian sensor nafas . . . . .	43
4.5	Tabel pengujian <i>throughput</i> rata-rata . . . . .	46
4.6	Tabel pengujian <i>throughput</i> maksimal . . . . .	47
4.7	Hasil pengujian kejelasan <i>user interface</i> . . . . .	52

*Halaman ini sengaja dikosongkan*

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Puskesmas(Pusat Kesehatan Masyarakat) merupakan unit pelaksana teknis dinas kesehatan kabupaten/kota yang bertanggung jawab menyelenggarakan pembangunan kesehatan di suatu wilayah kerja [1]. Saat ini puskesmas telah membuka layanan rawat inap selama 24 jam yang mana, pasien dirawat dan menginap di puskesmas hingga pasien tersebut sembuh dari sakit. Data pada tahun 2015 menunjukkan bahwa sebanyak 3396 puskesmas telah membuka layanan rawat inap 24 jam [2]. Sehingga peralatan medis juga ditingkatkan guna perawatan pasien yang baik.

Disisi lain kesehatan pasien ketika dirawat di puskesmas merupakan hal yang harus dipantau oleh paramedis secara berkala. Pemantauan dilakukan guna menghindari adanya hal-hal yang tidak diinginkan terhadap pasien. Untuk menangani hal tersebut, paramedis memeriksa data-data pemantauan pasien yang sedang berjalan. Paramedis membuat laporan untuk dianalisa oleh dokter. Analisis dokter mempengaruhi cara penanganan pasien. Oleh sebab itu dibutuhkan data yang cukup absah untuk menganalisis pasien agar hasil benar-benar tepat.

Paramedis melakukan pemeriksaan dan pengambilan data pasien hanya pada waktu yang telah terjadwal. Diluar jadwal pemeriksaan, data yang terekam oleh alat pemantau kesehatan tidak tersimpan secara otomatis. Hal ini berdampak pada hasil analisi yang kurang tepat. Jika hasil analisis kurang tepat maka, penanganan pasien juga kurang tepat.

Penelitian tentang pemantauan *vital sign* pada rumah sakit telah dilakukan oleh Ismail [3]. Pada penelitiannya dilakukan pemantauan terhadap *vital sign* pasien yang dirawat di rumah sakit berbasis mikrokontroler dan TCP/IP (*Transmission Control Protocol/Internet Protocol*). Penelitian lain yaitu tentang penyimpanan data pantauan kesehatan ke *cloud* telah dilakukan oleh Mohammed et al. [4]. Pada penelitiannya dilakukan pengambilan data oleh sensor yang kemudian diunggah ke cloud dan diakses menggunakan *smartphone*. Sensor yang digunakan adalah ECG (*Electroca-*

*rdiogram*). Pengunggahan data dilakukan menggunakan FTP(*file transfer protocol*).

Dalam tugas akhir ini akan dibuat suatu alat untuk menampilkan *vital sign* pasien sekaligus menyimpan data ke *cloud storage*. Dengan *vital sign monitoring*, data *vital sign* pasien dapat dipantau dan tersimpan secara otomatis. *Vital sign* didapat dengan mengintegrasikan sensor-sensor kesehatan. Data yang didapat dari sensor masuk ke mikrokontroler lalu dikirim ke *single board computer* untuk ditampilkan disisi pasien. Data disimpan pada penyimpanan lokal kemudian dilakukan sinkronisasi ke *cloud storage*. Data yang ditampilkan pada sistem pemantauan berupa detak jantung, jumlah nafas, tekanan darah, dan suhu tubuh. *Vital sign monitoring* ini diaplikasikan pada puskesmas yang membuka layanan rawat inap, sehingga diharapkan penelitian tugas akhir ini mampu membantu penanganan pasien pada puskesmas tersebut.

## 1.2 Perumusan Masalah

Puskesmas yang telah membuka layanan rawat inap memiliki peralatan medis yang terbatas. Kesehatan pasien yang dirawat di puskesmas harus selalu dalam pemantauan paramedis. Sehingga pihak puskesmas diharuskan menyediakan alat yang dapat memantau *vital sign* pasien.

Selain itu pemeriksaan pasien yang dilakukan di puskesmas hanya pada waktu yang telah dijadwalkan. Data yang terekam oleh alat pemantau pasien bersifat sekali pengambilan. Sehingga data yang berjalan secara *realtime* tidak tersimpan secara otomatis. Akibatnya tidak ada data untuk dijadikan referensi dengan data yang baru diambil.

Saat ini belum ada sistem yang digunakan untuk menyimpan histori data pasien yang telah terekam. Padahal dengan analisa histori *vital sign*, seorang dokter bisa menganalisa maupun mendiagnosa penyakit yang dialami pasien. Banyak penanganan pasien yang terlambat atau kurang tepat dikarenakan data *vital sign* pasien tidak absah untuk dianalisa. Hal ini dapat membahayakan kondisi pasien yang tidak diinginkan.

### 1.3 Tujuan dan Manfaat

Tujuan dari tugas akhir ini yaitu untuk mengembangkan sistem yang dapat mengakuisisi data *vital sign* pasien secara *realtime* lalu memvisualisasikan pada layar *display* dan melakukan penyimpanan data *vital sign* ke penyimpanan cloud. Dalam tugas akhir ini diimplementasikan beberapa ruang lingkup seperti *emmbdedded system*, jaringan komputer, dan algoritma pemrograman.

Diharapkan dari pembuatan sistem *vital sign monitoring* ini dapat melengkapi peralatan medis yang ada di puskesmas rawat inap. Dokter dengan mudah dapat mengakses histori pasien yang telah terekam oleh alat. Selain itu, diharapkan alat ini dapat digunakan untuk proses analisa dan diagnosa terhadap pasien dengan baik dan tepat.

### 1.4 Batasan Masalah

Batasan masalah pada tugas akhir ini yaitu:

1. Sensor yang digunakan adalah sensor ECG, tekanan darah, suhu tubuh, dan hembusan nafas.
2. Mikrokontroler yang digunakan adalah ATmega328 pada *board* Arduino Uno dan e-Health Shield.
3. *Single Board Computer* yang digunakan adalah Raspberry Pi 3 Model B.
4. Puskesmas tersedia akses internet.
5. Parameter *vital sign* mengacu pada dokumentasi sensor-sensor yang digunakan.
6. Pengujian sitem tidak dilakukan di lapangan.

### 1.5 Sistematika Penulisan

Laporan penelitian tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu :

1. BAB I Pendahuluan

Bab ini berisi uraian tentang latar belakang permasalahan, rumusan permasalahan, tujuan dan manfaat penelitian, batasan masalah, dan sistematika penulisan.

## 2. BAB II Tinjauan Pustaka

Pada bab ini berisi tentang uraian secara sistematis dasar teori yang berhubungan dengan permasalahan yang dibahas pada penelitian tugas akhir ini. Metode maupun referensi yang digunakan sebagai dasar dalam penelitian, yaitu informasi tentang puskesmas, teori *vital sign*, dasar *single board computer* & mikrokontroler, penyimpanan ke *cloud server*, dan komunikasi serial.

## 3. BAB III Desain Sistem dan Implementasi

Bab ini berisi tentang penjelasan-penjelasan terkait eksperimen perancangan alat yang akan dilakukan dan langkah-langkah data diolah hingga menghasilkan visualisasi. Untuk melakukan perancangan sistem maka, digunakan blok diagram atau *work flow* agar sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk diimplementasikan pada pelaksanaannya.

## 4. BAB IV Pengujian dan Analisa

Bab ini menjelaskan tentang pengujian eksperimen yang dilakukan terhadap data dan analisisnya. Data *vital sign* yang diakuisisi oleh sensor akan diuji pada pengiriman dan penyimpanannya. Selain itu visualisasi pada tampilan pada layar juga dilakukan pengujian.

## 5. BAB V Penutup

Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk pengembangan lebih lanjut juga dituliskan pada bab ini.

# BAB 2

## TINJAUAN PUSTAKA

Penelitian ini dilakukan berdasarkan beberapa teori penunjang sebagai bahan acuan dan referensi. Teori penunjang digunakan sebagai landasan pembuatan ataupun pengujian dalam penelitian ini. Dengan demikian pengerjaan penelitian ini menjadi lebih terarah.

### 2.1 Puskesmas

Puskesmas(Pusat Kesehatan Masyarakat) merupakan unit pelaksana teknis dinas kesehatan kabupaten atau kota yang bertanggungjawab menyelenggarakan pembangunan kesehatan di suatu wilayah kerja. [1]

1. Unit Pelaksana Teknis

Sebagai Unit Pelaksana Teknis Dinas(UPTD) kesehatan kabupaten atau kota, puskesmas berperan menyelenggarakan sebagian dari tugas teknis operasional dinas kesehatan kabupaten atau kota dan merupakan unit pelaksana tingkat pertama serta ujung tombak pembangunan kesehatan di Indonesia.

2. Pembangunan Kesehatan

Pembangunan kesehatan adalah penyelenggaraan upaya kesehatan oleh bangsa Indonesia untuk meningkatkan kesadaran, kemauan dan kemampuan hidup sehat bagi setiap orang agar terwujud derajat kesehatan masyarakat yang optimal.

3. Penanggungjawab Penyelenggaraan

Penanggungjawab utama penyelenggaraan seluruh upaya pembangunan kesehatan di wilayah kabupaten atau kota adalah dinas kesehatan kabupaten atau kota, sedangkan puskesmas bertanggungjawab hanya sebagian upaya pembangunan kesehatan yang dibebankan oleh dinas kesehatan kabupaten atau kota sesuai dengan kemampuannya.

4. Wilayah Kerja

Secara nasional, standar wilayah kerja puskesmas adalah satu kecamatan, tetapi apabila di satu kecamatan terdapat lebih dari satu puskesmas, maka tanggungjawab wilayah kerja dibagi antar puskesmas, dengan memperhatikan keutuh-

an konsep wilayah (desa atau kelurahan atau RW). Masing-masing puskesmas tersebut secara operasional bertanggungjawab langsung kepada dinas kesehatan kabupaten atau kota.

Pada dasarnya, puskesmas memberikan pelayanan preventif, promotif, kuratif hingga rehabilitatif baik melalui upaya kesehatan perorangan atau upaya kesehatan masyarakat. Puskesmas yang memberikan pelayanan rawat inap selain pelayanan rawat jalan disepakati oleh puskesmas dan dinas kesehatan yang bersangkutan. Perawat memberikan pelayanan di masyarakat seperti puskesmas pembantu, puskesmas keliling, posyandu, pos kesehatan desa maupun pos bersalin desa (polindes).

## 2.2 *Vital Sign*

Dalam dunia medis, *vital sign* merupakan suatu nilai yang digunakan untuk mengukur fungsi dasar tubuh. *Vital sign* pasien sangat penting untuk dilakukan pemantauan. Pemantauan *vital sign* dilakukan guna mengetahui atau menganalisa kesehatan fisik seseorang secara umum, menunjukkan penyakit yang mungkin diidap seseorang, dan menunjukkan kemajuan kesehatan seseorang. Biasanya perbedaan kondisi normal *vital sign* pasien didasarkan perbedaan usia, jenis kelamin, atau berat tubuh. *Vital sign* pasien dilakukan observasi secara berkala oleh paramedis setiap satu jam sekali.

Paramedis membuat laporan *vital sign* yang sedang berjalan pada saat observasi. *Vital sign* dilaporkan kepada dokter untuk proses analisa lebih lanjut. Elliott dan Coventry menjelaskan bahwa terdapat delapan buah vital sign yang dapat digunakan untuk memantau kesehatan pasien [5], namun ada empat *vital sign* yang terpenting yaitu suhu tubuh, tekanan darah, dan hembusan nafas.

### 2.2.1 Detak Jantung

Detak jantung atau disebut dengan *pulse* merupakan jumlah detak jantung per satuan waktu, biasanya per menit. Detak jantung didasarkan pada jumlah kontraksi ventrikel (bilik bawah jantung). Detak jantung memungkinkan terlalu cepat (takikardia) atau terlalu lambat (bradikardia). Untuk mengetahui BPM (*Beat Per Minute*) pasien digunakan alat ECG ataupun secara manual. Proses pemeriksaan detak jantung dilakukan selama minimal 30 detik atau

lebih. Jumlah detak jantung terhadap usia pasien dijelaskan pada tabel 2.1.

**Tabel 2.1:** Tabel jumlah detak jantung per menit terhadap usia pasien [6]

Usia	Minimal (BPM)	Maksimal (BPM)	Rata-rata (BPM)
Baru lahir	100	180	140
1 bulan - 1 tahun	80	160	120
1 - 3 tahun	80	130	105
3 - 6 tahun	80	120	100
6 - 12 tahun	65	100	83
12 - 18 tahun	60	90	85
19 - 69 tahun	60	100	80
>70 tahun	60	100	80

### 2.2.2 Suhu Tubuh

Suhu tubuh adalah indikasi adanya metabolisme di tubuh manusia. Untuk membuang sisa-sisa metabolisme, tubuh manusia membuang panas dalam tubuh melalui proses *thermoregulation*. Suhu tubuh dikontrol oleh hipotalamus. Hal yang mempengaruhi suhu tubuh antara lain seperti patofisiologi, keadaan kulit, atau faktor usia. Suhu tubuh dapat diukur menggunakan thermometer analog maupun digital. Suhu tubuh orang dewasa yang normal berkisar antara  $36^{\circ}\text{C}$  -  $37^{\circ}\text{C}$ . Suhu tubuh seseorang dipengaruhi oleh usia, aktifitas, hormon, rasa stres atau sakit yang diderita, dan obat-obatan yang diminum. Tabel suhu tubuh seseorang terhadap usia dijelaskan pada tabel 2.2. Dari tabel tersebut rata-rata suhu seseorang sejak umur satu tahun hingga dewasa mengalami penurunan.

Suhu pasien yang diukur yakni suhu utama. Suhu utama ini merupakan suhu yang keluar dari tubuh dan digunakan sebagai parameter *vital sign*. Untuk mendapatkan suhu tubuh pasien dilakukan pengukuran terhadap pasien. Ada beberapa cara untuk melakukan pengukuran suhu. Perbedaan cara pengukuran berdasarkan pada faktor usia, keadaan sakit pasien, dan kebijakan puskesmas. Hasil dari masing-masing pengukuran memiliki nilai yang berbeda-beda. Ada lima cara pengukuran suhu tubuh pasien:

**Tabel 2.2:** Tabel suhu tubuh terhadap usia pasien [6]

Usia	Minimal (C)	Maksimal (C)	Rata-rata (C)
Baru lahir	35.5	37.5	36.5
1 bulan - 1 tahun	37.4	37.6	37.5
1 - 3 tahun	37.4	37.6	37.5
3 - 6 tahun	37	37.2	37
6 - 12 tahun	37	37	37
12 - 18 tahun	36.1	37.2	36.65
19 - 69 tahun	36.1	37.2	36.65
>70 tahun	35	37.2	36.1

1. Pengukuran *Oral Cavity*.

Pengukuran suhu dilakukan dengan cara meletakkan alat ukur pada bawah lidah. Lalu pasien menutup mulut sampai nilai suhu keluar. Cara pengukuran ini banyak digunakan pada anak-anak maupun orang dewasa. Namun ketika pasien sakit bagian mulut, cara ini tidak disarankan. Jika pasien sehabis makan, minum, merokok, atau aktifitas yang memungkinkan menambah suhu maka, pengukuran dilakukan pada selang waktu antara 20 sampai 30 menit.

2. Pengukuran *Tympanic*.

Pengukuran ini menggunakan termometer timpani dengan teknologi infra merah untuk mengukur suhu tubuh dari panas gendang telinga (membran timpani) dan jaringan sekitarnya. Panas itu kemudian diubah menjadi pembacaan digital. Kebanyakan perangkat timpani menghasilkan tampilan digital yang mudah dibaca dengan cepat. Banyak termometer timpani memberi konversi Celsius dan Fahrenheit dan setara membaca untuk suhu oral dan rektal.

3. Pengukuran *Axillary*.

Pengukuran ini dilakukan pada bagian ketiak pasien. Pembacaan suhu umumnya 0.9°F (0.5°C) lebih rendah dari pada mulut atau telinga karena pengukuran *axillary* berada di bagian luar tubuh. Pembacaan normal untuk suhu aksilaris antara 96.6°F (35.9°C) dan 98°F (36.7°C). Faktor-faktor yang mempengaruhi suhu *axillary* adalah waktu saat pengukuran dan tingkat aktivitas pasien sebelum pengukuran suhu.

#### 4. Pengukuran *Rectal*.

Pengukuran *rectal* dilakukan pada lubang anus pasien. Pengukuran dengan cara ini paling tidak disukai untuk mengukur suhu karena ketidaknyamanan bagi pasien dan sifat prosedur yang invasif. Terkadang cara ini diprioritaskan pada pasien yang luka parah, cacat wajah, atau pasien yang koma.

#### 5. Pengukuran *Temporal*.

Pengukuran pada *arteri temporal*(dahi) adalah lokasi yang sangat baik untuk mengukur suhu karena cocok untuk semua umur dan tidak menimbulkan risiko cedera pada pasien atau untuk klinisi. Pengukuran dilakukan dengan cara pemindaian di dahi pasien. Pengukuran dengan cara ini menghasilkan nilai 1°C lebih tinggi dari pada pengukuran *axillary* dan 0.5°C lebih tinggi dari pada pengukuran *rectal*.

### 2.2.3 Jumlah Nafas

Nafas manusia merupakan indikator sehat atau tidaknya pernafasan manusia. Pernapasan manusia dibagi menjadi dua proses yaitu inspirasi atau proses menarik nafas dan ekspirasi. Proses inspirasi terjadi ketika otot diafragma berkontraksi. Saat otot diafragma berkontraksi, rongga dada akan mengembang sehingga tekanan di dalam rongga dada akan berkurang dan mengakibatkan udara masuk ke paru-paru. Pada saat mengeluarkan nafas otot diafragma akan melemas yang mengakibatkan tekanan di dalam rongga dada menjadi naik dan udara akan tertekan keluar. Jumlah nafas manusia per menit dipengaruhi oleh *hypercapnia*, *hypoxemia*, dan asidosis.

Jumlah nafas biasanya ditulis dalam satuan RPM (*Respiration Per Minute*). Untuk mengetahui RPM seseorang, dilakukan pemeriksaan selama satu menit. Jumlah nafas orang dewasa normal adalah antara 12 hingga 20 nafas per menit. Jumlah nafas seseorang terhadap usia dijelaskan pada tabel 2.3. Pada tabel tersebut nilai jumlah nafas seseorang semakin menurun ketika usia semakin tua.

### 2.2.4 Tekanan Darah

Tekanan darah adalah ukuran selisih tekanan darah saat jantung mengalami kontraksi (*systolic*) dan saat beristirahat (*diastolic*). Menurut National Heart, Lung, and Blood Institute (NHLBI), tekanan darah orang dewasa normal adalah dibawah atau sama dengan 120 mmHg sistole dan 80 mmHg diastole [7]. Tabel 2.4 menje-

**Tabel 2.3:** Tabel jumlah nafas terhadap usia pasien [6]

Usia	Minimal (RPM)	Maksimal (RPM)	Rata-rata (RPM)
Baru lahir	30	60	45
1 bulan - 1 tahun	30	60	45
1 - 3 tahun	25	40	32
3 - 6 tahun	20	35	27
6 - 12 tahun	20	30	25
12 - 18 tahun	12	20	16
19 - 69 tahun	12	20	16
>70 tahun	12	20	16

laskan tekanan darah terhadap bertambahnya usia seseorang. Dari tabel tersebut dapat disimpulkan bahwa semakin bertambah usia seseorang maka semakin bertambah pula ukuran normal tekanan darahnya. Untuk mengetahui tekanan darah seseorang dilakukan pengukuran dengan alat *sphygmomanometer* manual maupun digital. Hasil pengukuran tekanan darah dapat dipengaruhi oleh keadaan seseorang seperti stres, aktifitas keseharian, olahraga, obat-obatan, dan obesitas.

**Tabel 2.4:** Tabel tekanan darah terhadap usia pasien [6]

Umur	Sistole (mmHg)	Diastole (mmHg)
Baru lahir	60 - 90	20 - 60
1 bulan - 1 tahun	75 - 100	50 - 70
1 - 3 tahun	80 - 110	55 - 80
3 - 6 tahun	80 - 110	50 - 80
6 - 12 tahun	100 - 110	60 - 70
12 - 18 tahun	110 - 120	60 - 65
19 - 69 tahun	110 - 140	60 - 90
>70 tahun	120 - 140	70 - 90

### 2.2.5 Saturasi Oksigen

Saturasi oksigen adalah persentase untuk mengukur tingkat hemoglobin yang mengikat oksigen saat manusia bernafas. Saturasi oksigen biasa disebut SpO<sub>2</sub>. Untuk mengetahui saturasi oksigen digunakan alat pulsioximeter. Tubuh manusia setidaknya membutuhkan tingkat oksigen dalam darah sebesar 95% hingga 100%. Jika

dibawah 92% kemungkinan seseorang mengalami sakit pernapasan seperti hipoksemia. Hipoksemia ini merupakan penyakit yang disebabkan karena kelelahan, sesak, dan mungkin kebingungan.

### **2.2.6 Rasa Sakit**

Rasa sakit dapat digunakan sebagai *vital sign* yang menunjukkan tingkat ketahanan pasien terhadap rasa sakit dan tingkat keefektifan *analgesic*(obat bius). Seringnya pemantauan dapat dibuat sebuah penilaian rasa sakit terhadap pasien. Hasil penilaian tersebut dapat juga dimasukkan dalam pemantauan *vital sign*. Terkadang pasien harus mengungkapkan rasa sakit untuk mendapatkan penilaian.

### **2.2.7 Tingkat Kesadaran**

Tingkat kesadaran atau *Glasgow Coma Scale* adalah parameter yang digunakan untuk menilai tingkat kesadaran pasien. Untuk mengetahui tingkat kesadaran dilakukan dengan cara memberi rangsangan pada pasien. Respon pasien terhadap rangsangan yang diberikan akan berbeda-beda tergantung tingkat kesadaran pasien tersebut. Tingkat kesadaran dapat dipicu karena keadaan tertentu seperti kurangnya kadar oksigen dalam otak atau kurang aliran darah ke otak dan adanya tekanan yang berlebihan di dalam rongga tulang kepala.

### **2.2.8 Ekskresi Urin**

Urin yang diekskresikan oleh pasien dapat menjadi indikator tingkat keseimbangan cairan tubuh dan fungsi dari ginjal pasien tersebut. Kemungkinan terjadi kesulitan dalam pemantauan urin secara langsung jika pasien tidak memiliki kateter pada saluran kencing. Pada orang dewasa, ekskresi urin normal minimal 0.5 ml/kg/jam. Pada pasien yang mengalami ekskresi kurang dari 0.5 ml/kg/jam selama 2 jam berturut-turut merupakan tanda penurunan aliran darah pada ginjal (*renal hypoperfusion*).

## **2.3 Arsitektur *Monitoring E-Health***

Rancangan *vital sign monitoring* dengan penyimpanan *cloud storage*, didesain berdasarkan arsitektur *monitoring E-Health*. Menurut Mukherjee et al. [8] ada tiga lapisan arsitektur untuk membuat *vital sign monitoring*, yaitu:

1. *Perception Layer*

Lapisan ini merupakan lapisan paling bawah, yakni lapisan yang berhubungan dengan sensor-sensor kesehatan. Terdapat dua jenis sensor yaitu sensor medis dan sensor keadaan lingkungan sekitar. Pada lapisan ini data *vital sign* diakuisisi oleh sensor medis yang ditempelkan ke tubuh pasien. Untuk pengiriman data sensor ke *layer* selanjutnya dapat digunakan komunikasi melalui jaringan LAN(*Local Area Network*), *Zigbee*, maupun *Bluetooth*.

2. *Middleware and APIs Layer*

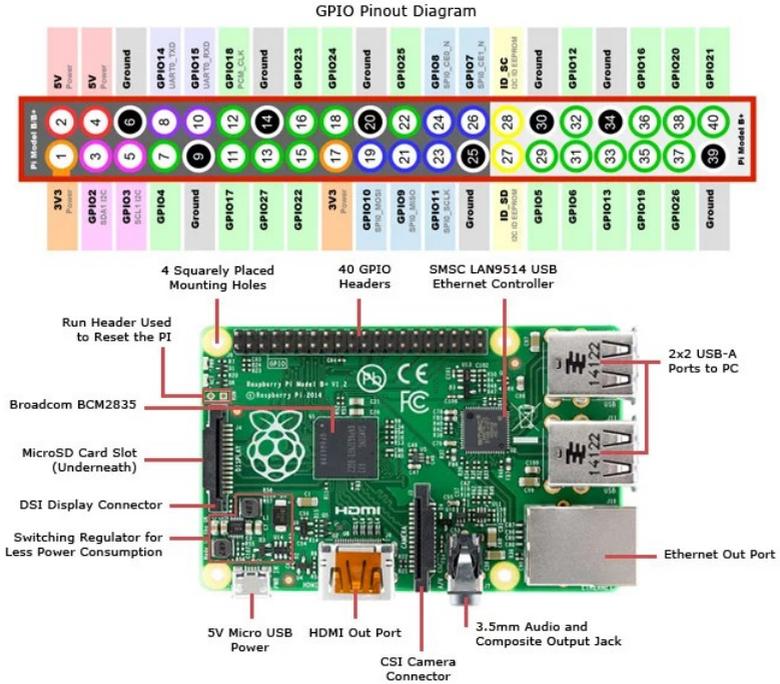
Pada *layer* ini terdapat sistem yang mengatur jalannya pengiriman data *vital sign*. Pada *layer* dibuat sebuah APIs(Application Programming Interfaces) yang menangani *request* dari *Perception layer* maupun dari *E-Health Application and Service Layer*. *Request* yang dibuat meliputi penanganan data masuk ke *database* maupun akses data dari *database*. Dari *request* tersebut sistem pada *APIs layer* melakukan *response* berupa informasi berhasil atau tidaknya pengiriman data. Secara singkatnya *layer* ini merupakan gerbang penghubung antara *Perception layer* dan *Service layer*.

3. *E-Health Application and Service Layer*

Lapisan ini merupakan lapisan yang berkaitan dengan visualisasi data yang telah diakuisisi. Lapisan ini memuat data-data histori pasien untuk dianalisa dan dapat juga memberikan timbal balik berupa tindakan dari dokter terhadap pasien yang dipantau *vital sign*-nya. Data yang dianalisa merupakan data histori dan data yang termonitor. Dari data histori dapat diketahui waktu terjadi bahaya pada pasien dan dapat ditindaklanjuti segera mungkin.

## 2.4 *Single Board Computer Raspberry Pi*

Raspberry Pi merupakan *single board computer* yang dapat digunakan untuk menjalankan perangkat lunak maupun proses pengolahan data. Selain itu Raspberry Pi dapat melakukan proses visualisasi data seperti halnya pada komputer personal. Raspberry Pi 3 Model B berbasis sistem operasi Raspbian, yaitu sistem operasi yang dikembangkan oleh perusahaan Raspberry Pi yang merupakan turunan dari sistem operasi Debian.



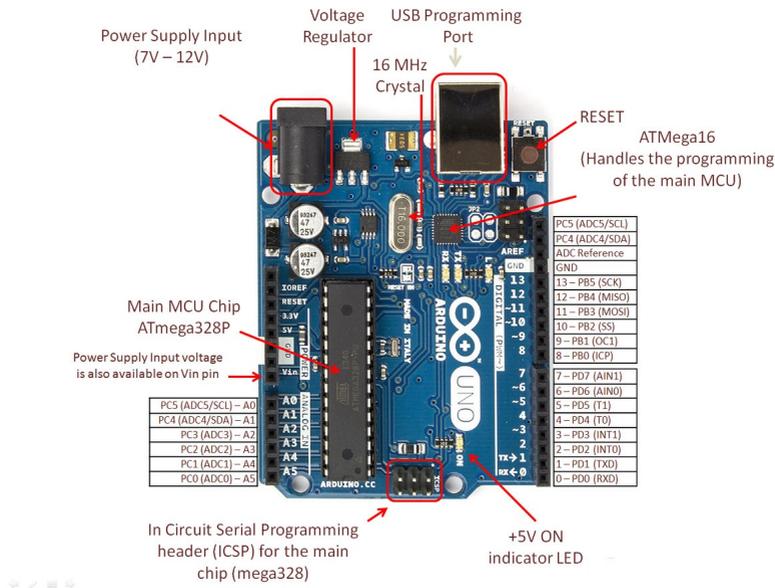
**Gambar 2.1:** Raspberry Pi 3 Model B *pinout* [9]

Gambar 2.1 merupakan *pinout* dari Raspberry Pi 3 Model B. Pada gambar tersebut Raspberry Pi 3 Model B menggunakan CPU(*Central Processing Unit*) Broadcom BCM2837 64bit dengan kecepatan 1.2Ghz dan GPU(*Graphic Processing Unit*) jenis Broadcom VideoCore IV. Raspberry Pi 3 Model B memiliki kapasitas RAM(Random Access Memory) sebesar 1GB dan menggunakan *microSD* sebagai *storage*, sehingga *operating system* dan *file* tersimpan pada *microSD* tersebut. Selain itu Raspberry Pi 3 Model B dilengkapi dengan *ethernet port*, *wireless fidelity* 802.11n 2.4Ghz, dan *bluetooth 4.1*. Pada *input/output*, Raspberry Pi 3 Model B mempunyai GPIO(*General Purpose Input Output*) sebanyak 40 pin selain itu terdapat beberapa *port* seperti *port* HDMI(*High Definition Multimedia Interface*), *audio jack* 3.5mm, empat *port* USB(*Universal*

Serial Bus), Camera Serial Interface(CSI), dan Display Serial Interface(DSI) [10].

## 2.5 Mikrokontroler Arduino

Arduino merupakan papan mikrokontroler yang dapat melakukan proses input maupun output dengan cara *upload* program tertentu, sebagai contoh program akuisisi data oleh sensor suhu. Pada Arduino Uno mikrokontroler yang digunakan yakni ATmega328. Arduino Uno beroperasi pada tegangan lima volt dan tegangan yang direkomendasikan sebesar tujuh sampai 12 volt.



**Gambar 2.2:** Arduino Uno *pinout* [11]

Gambar 2.2 merupakan pinout diagram pada Arduino Uno. Pada gambar tersebut Arduino Uno memiliki 14 pin *input/output* yang mana enam pin diantaranya menyediakan PWM (*Pulse Width Modulation*) dan terdapat enam pin analog *input*. Untuk penggunaan arus pada setiap pin membutuhkan arus sebesar 20 mA dan pada

*output* 3.3 volt terdapat arus 50 mA. *Clock speed* pada Arduino yaitu sebesar 16 Mhz. *File binary* program yang dapat diupload pada Arduino harus kurang dari 31.5 KB karena pada Arduino Uno hanya menyediakan kapasitas *flash memory* sebesar 32 KB dan 0.5 KB digunakan sebagai *bootloader*. Arduino mempunyai SRAM sebesar 2 KB dan EEPROM 1 KB. Pada Arduino Uno terdapat *In Circuit Serial Programming* yakni pemrograman pada *chip* mikrokontroler juga dapat dilakukan langsung melalui pin tersebut. Selain itu Arduino Uno dilengkapi sebuah tombol *reset*, led (*light-emitting diode*) indikator *on/off*, *power supply input*, dan led *digital* pin 13. Pemrograman Arduino dapat dilakukan melalui komputer dengan *software* Arduino untuk memprogram [12].

## 2.6 Komunikasi Serial

Komunikasi Serial UART (*Universal Asynchronous Receiver Transmitter*) mempresentasikan sebuah IC (*integrated circuit*) yang berfungsi untuk menerjemahkan antara bit-bit paralel data dan bit-bit serial. IC tersebut termasuk bagian dari sebuah mikrokontroler. Dalam proses komunikasi, antara *receiver* dan *transmitter* harus mempunyai pengaturan yang sama. Salah satu pengaturan yang paling penting pada komunikasi serial yaitu besarnya *baud rate* [13]. *Baud rate* merupakan kecepatan aliran data pada saat proses komunikasi, biasanya satuan *baud rate* dituliskan dalam bps (*bit per second*).

Komunikasi serial pada dua jalur *Rx* dan *Tx* dapat dilakukan komunikasi secara:

1. *Simplex* – Satu arah pengiriman data, *transmitter* ke *receiver*
2. *Half Duplex* – Pengiriman data secara bergantian antara perangkat satu dengan lainnya
3. *Full Duplex* – Antara dua perangkat dapat menerima dan mengirim data secara bersamaan

Secara teknis pemasangan komunikasi serial UART antara *receiver* dan *transmitter* dihubungkan saling bersilang antara *Rx* dan *Tx* lainnya. Hal ini dikarenakan data yang dikirimkan oleh *transmitter* melalui *Tx* akan diterima oleh *receiver* melalui *Rx*. Begitu pun sebaliknya jika ada respon atau pengiriman data dari *receiver*

maka akan diterima oleh *transmitter*. Pada komunikasi serial hal dapat mempengaruhi pengiriman data yaitu *power supply*, sumber elektromagnetik, dan *resonant frequency*.

## 2.7 E-Health Shield

E-health Shield merupakan papan *peripheral* tambahan yang ditujukan pada *board* Arduino dan Raspberry Pi [14]. E-health Shield berfungsi untuk pengambilan data *vital sign*. E-health Shield mampu mengakuisisi data *vital sign* dengan menggunakan sensor-sensor kesehatan. Ada sembilan sensor kesehatan yang dapat digunakan untuk pemantauan *vital sign*, yaitu:

1. Sensor posisi tubuh.
2. Sensor suhu tubuh.
3. Sensor EMG (*Electromyography*)
4. Sensor EMG
5. Sensor hembusan nafas
6. Sensor GSR (*Galvanic Skin Response*).
7. Sensor tekanan darah.
8. Sensor *Pulsioximeter*
9. Sensor *Glucometer*

Pada tugas akhir ini digunakan empat buah sensor kesehatan yaitu sensor ECG, sensor suhu tubuh, sensor hembusan nafas, dan tekanan darah. Masing-masing sensor tersebut memiliki karakteristik yang berbeda-beda. Dalam proses pengambilan data dan pengolahan data, e-Health Shield mempunyai *library* tersendiri yang ditambahkan pada pemrograman Arduino. *Library* tersebut bertujuan untuk mempermudah pengolahan *raw* data yang diakuisisi oleh sensor menjadi data yang dapat digunakan secara umum. Penggunaan *library* cukup dengan memanggil fungsi yang terdapat pada *library* tersebut.

### 2.7.1 Sensor ECG

Pengambilan ECG dilakukan dengan memasang tiga elektroda pada bagian tubuh sebagai berikut:

1. Kutub positif bagian dada sebelah kanan
2. Kutub netral bagian dada sebelah kiri
3. Kutub negatif bagian perut sebelah kiri

Data yang diakuisisi oleh elektroda tersebut merupakan te-

gangan yang muncul saat jantung berkontraksi dan relaksasi. Data yang muncul berupa data analog tegangan yang telah dikonversi antara nol sampai lima volt.

## 2.7.2 Sensor Hembusan Nafas

Sensor nafas menggunakan dua kutub yaitu positif dan negatif. Dua kutub tersebut dihubungkan dengan komponen sensor nafas yang peka terhadap hembusan udara. Data yang diakuisisi berupa data analog dengan nilai antara 0 sampai 1023. Nilai tersebut mempresentasikan kuat lemahnya hembusan nafas dari pasien. Nilai tersebut diteruskan ke mikrokontroler untuk diambil.

## 2.7.3 Sensor Tekanan Darah

Pada *library*, sensor tekanan darah diperoleh dengan *buffering* data. Sebelumnya dilakukan pembacaan data dari *Sphygmomanometer* yang berupa data *array* hasil setiap dilakukan pengukuran tekanan darah. Data yang diperoleh meliputi sistole, diastole, waktu pengukuran, dan *pulse*. Proses pengambilan data dari *Sphygmomanometer* dibutuhkan minimal 64 byte kapasitas RAM(Random Access Memory) pada mikrokontroler. Sehingga dibutuhkan perubahan konfigurasi pada *serial library*.

## 2.7.4 Sensor Suhu Tubuh

Pada sensor suhu terdapat pengukuran *wheatstone bridge* serta tegangan referensi yang mana bertujuan untuk kalibrasi sensor. Mulanya dilakukan pemeriksaan nilai dari tiga *resistor* yang berhubungan dengan sensor suhu. Nilai hambatan tersebut dimasukkan pada *library* e-Health Shield. Dari perhitungan *wheatstone bridge* dihasilkan nilai resistansi total. Rumus perhitungan sensor suhu dijelaskan sebagai berikut:

$$T = \frac{\log\left(\frac{R_t^a}{K_1}\right)}{\log(K_2)} \quad (2.1)$$

dimana :

$R_t^a$  = Resistansi total dari *wheatstone bridge* ( $\Omega$ ).

$K_1$  dan  $K_2$  = Konstanta.

Dari persamaan 2.1, nilai konstanta  $K_1$  dan  $K_2$  dapat berubah-ubah sesuai dengan resistansi total  $R_t^a$ . Tabel konstanta  $K_1$  dan  $K_2$  terhadap  $R_t^a$  dijelaskan pada tabel 2.5.

**Tabel 2.5:** Tabel Konstanta  $K_1$  dan  $K_2$  [14]

$R_t^a(\Omega)$	$K_1$	$K_2$
<988.1	5575.94572	0.96218
988.1–1204.7	5859.06368	0.96112
1204.8–1477.0	6118.01620	0.96008
1477.1–1822.7	6403.49306	0.95883
>1822.8	6638.20457	0.95768

## 2.8 Cloud Storage

Cloud Storage merupakan media penyimpanan secara daring. Saat ini penggunaan *cloud* telah menjadi paradigma baru dalam dunia teknologi informasi seperti *internet of things* [4]. Banyak data yang disimpan di *cloud* agar dapat diakses oleh pengguna di mana pun mereka berada. Saat ini telah banyak penyedia layanan *cloud storage* secara gratis maupun berbayar. Banyak perusahaan digital maupun bukan digital hingga pemerintahan telah menggunakan penyimpanan *cloud* sebagai tempat penyimpanan data-data yang dikelola maupun sebagai *back up* data. Dari segi privasi, *cloud storage* dibagi menjadi empat jenis yaitu [15]:

1. *Personal Cloud Storage*

*Storage* jenis ini biasanya digunakan oleh perseorangan untuk *back up* data-data yang dimiliki. Beberapa *smartphone* bahkan telah menambahkan aplikasi *cloud* untuk mempermudah penggunaanya ketika ingin melakukan *back up* terhadap data-data.

2. *Public Cloud Storage*

Jenis *storage* ini sering digunakan oleh perusahaan yang memiliki jenis data tidak terstruktur. Perusahaan menyewa *cloud storage* dari *provider cloud* dan meminta untuk mengelola seluruh data yang dimiliki perusahaan. Selain itu jenis *storage*

ini sering digunakan dalam hal berbagi secara publik.

### 3. *Private Cloud Storage*

Jenis ini umumnya digunakan oleh perusahaan yang menginginkan fasilitas keamanan serta kecepatan lebih dibanding jenis *cloud storage* yang lain. Pada *storage* ini *provider* akan membentuk sebuah infrastruktur dalam pusat data perusahaan, dan keduanya akan berintegrasi didalamnya.

### 4. *Hybrid Cloud Storage*

Merupakan gabungan antara *public* dan *private*. Data akan terbagi 2, yaitu data yang dianggap penting akan tersimpan dalam *private cloud storage* dan data lainnya di simpan dalam *public cloud storage*. Pada *storage* ini terdapat pilihan pengaturan data mana saja yang dijadikan *private* atau publik.

Sistem penyimpanan *cloud* dapat dianggap sebagai jaringan terdistribusi pusat data yang biasanya menggunakan teknologi *cloud computing* seperti virtualisasi dan menawarkan beberapa jenis interface untuk menyimpan data. [16]

Kelebihan *cloud storage* yaitu data dapat diakses dari berbagai lokasi (misalnya dari perangkat *mobile*) tanpa perlu menyamakan perangkat keras atau perangkat lunak yang sesuai, dan sinkronisasi fitur yang memungkinkan pelanggan untuk selalu memiliki akses ke versi terbaru dari perangkat (komputer personal, laptop, *smartphone*). Penggunaan *cloud storage* dapat dimanfaatkan sebagai layanan *Copy, Backup, Synchronization of Devices, dan Sharing Files*.

## 2.9 Protokol HTTP

Protokol HTTP (*Hypertext Transfer Protocol*) merupakan salah satu protokol yang sering digunakan dalam pertukaran informasi. HTTP Protokol populer diaplikasikan pada aplikasi web. Protokol HTTP mempunyai dua layanan / fitur terhadap pengiriman data yaitu *request* dan *response* [17]. HTTP Protokol menggunakan URL (*Uniform Resource Locator*) sebagai alamat terhubungnya antara *client* dan *server*. Ada empat bagian dari URL yaitu:

1. *Protocol*: Protokol yang digunakan (HTTP)
2. *Hostname*: Nama domain server (sebagai contoh *www.its.ac.id*)
3. *Port*: *Port* yang digunakan oleh server. Jika tidak dituliskan umumnya menggunakan *port 80*

4. *Path-and-file-name*: Nama dan lokasi file yang diminta pada direktori tertentu

Pada proses *request* dan *response* data antara *client* dan *server*, protokol HTTP menyediakan beberapa *method* pengiriman diantaranya:

1. GET: *Method* ini membolehkan *client* meminta *resource* atau pun file yang berada pada *server*.
2. HEAD: Untuk mendapatkan *header* yang diperbolehkan pada *method* GET.
3. POST: Digunakan untuk mengirim data ke *server*.
4. PUT: Meminta *server* menyimpan data.
5. DELETE: Meminta *server* menghapus data.
6. TRACE: Minta server untuk mengembalikan mendapatkan jejak dan tindakan.
7. OPSI: Digunakan untuk mengetahui daftar *method* yang didukung.
8. CONNECT: Digunakan untuk memberi tahu proxy untuk membuat koneksi ke *host* lain dan cukup membalas kontennya, tanpa mencoba mengurai atau menyimpannya. Biasanya digunakan pada koneksi SSL(*Secure Sockets Layer*) pada *proxy*.

Jika melakukan koneksi ke *server* dengan *method* GET pada sebagai contoh "*http://computer-its.com/index.php*", maka terdapat sebuah *header* yang dikirimkan sebagai contoh seperti berikut:

```
Connect to 203.177.40.1 on port 80 ... ok
GET /index.php HTTP/1.1
Host: computer-its.com
Connection: keep-alive
User-Agent: Mozilla/4.0
Accept-Encoding: gzip
Accept-Charset: ISO-8859-1,UTF-8;q=0.7,*;q=0.7
Cache-Control: no-cache
Accept-Language: en-us
```

Mulanya *port* 80 pada *server* dalam keadaan *listen*, artinya *server* menunggu *client* yang terhubung dengannya. Jika *client* telah terhubung dengan *server* maka, *client* melakukan *request*. *Request*

dari *client* diterima oleh *server* dengan *header* tertentu. Jika alamat atau sub direktori pada *server* valid maka, *server* akan merespon *client* dengan mengirimkan konten yang dituju. Saat proses berlangsung, *socket* koneksi akan tetap terbuka antara *client* dengan *server*. Respon yang dikirim misalnya seperti berikut:

```
Status: HTTP/1.1 200 OK
Date: Wed, 23 Mar 2017 12:07:00 GMT
Server: Apache/2.2.11 (Unix) mod_ssl/2.2.13
Content-Length: 35
X-Pingback: http://computer-its.com/index.php
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html
```

Untuk pemeriksaan *request client* sukses atau tidak dapat dilihat pada hasil respon dari server. Beberapa kode status HTTP yang umum dan sering ditemukan antara lain:

1. 200 - Server berhasil mengirim kembali halaman (sukses)
2. 301 - *Source* yang dituju telah dipindah ke lokasi lain
3. 400 - *Request* jelek, artinya *server* tidak mengerti dengan *request* dari *client*
4. 404 - Halaman yang diminta tidak ada
5. 500 - *Internal Server Error*. Terjadi kesalahan pada server
6. 502 - *Bad Gateway*. *Proxy* atau *Gateway* tidak dapat menerima respon dengan baik
7. 503 - Server sementara tidak tersedia

Pengiriman pada *method* GET dilakukan dengan cara membuat data dalam bentuk variabel dan nilai dari data tersebut lalu digabungkan dengan alamat URL yang dituju. Data akan diterima oleh *server* dengan cara menguraikan data yang ada pada URL tersebut berdasarkan variabel yang telah dibuat.

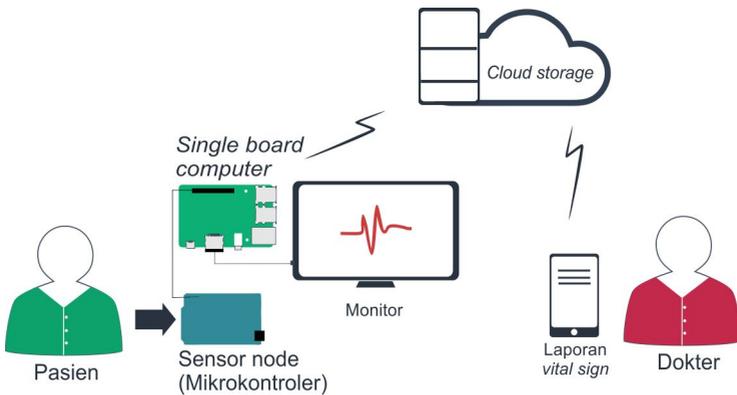
*Halaman ini sengaja dikosongkan*

# BAB 3

## DESAIN DAN IMPLEMENTASI SISTEM

### 3.1 Desain Sistem

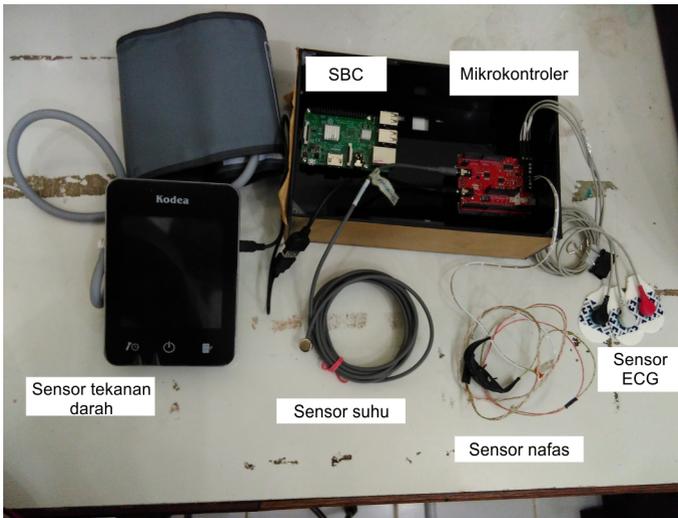
Guna merealisasikan penelitian tugas akhir ini, dibuat sebuah desain sistem yang menggambarkan konsep dan kerja sistem. Gambaran kerja sistem dijelaskan pada Gambar 3.1.



**Gambar 3.1:** Gambaran umum kerja sistem

Dari gambaran sistem kerja tersebut diuraikan secara singkat yaitu mulanya dilakukan pengambilan data oleh mikrokontroler yang terhubung dengan sensor kesehatan. Kemudian data dikirim melalui komunikasi serial UART ke SBC. Data yang telah dikirim divisualisasikan dengan tampilan yang interaktif dan dilakukan penyimpanan pada lokal. Jika terdapat koneksi ke *server* maka dilakukan penyimpanan data ke *cloud storage*. Data disimpan dalam bentuk *database* yang dapat diakses kembali guna laporan *vital sign* pasien. Terkait gambaran umum kerja sistem, sistem yang dibuat terbagi menjadi lima proses utama yaitu proses pengiriman data dari mikrokontroler ke SBC, proses penyimpanan lokal data *vital sign*, proses visualisasi data, proses penyimpanan data ke *cloud storage*, dan proses pembuatan laporan *vital sign*.

Dari desain sistem kerja tersebut diimplementasikan rancangan alat yang mana terintegrasi dengan empat buah sensor kesehatan yaitu sensor ECG, sensor suhu tubuh, sensor hembusan nafas, dan sensor tekanan darah. Masing-masing alat dan sensor ditunjukkan oleh gambar 3.2. Gambar 3.3 merupakan hasil *packing* dari keseluruhan rancangan yang telah dibuat.



**Gambar 3.2:** Rancangan alat dan integrasi sensor

### 3.2 Akuisisi dan Transmisi Data pada Sensor *Node*

Pada pembuatan program mikrokontroler, digunakan board Arduino yang berbasis mikrokontroler bertipe ATmega328. Selain itu, pada board tersebut ditambahkan sebuah *shield* yaitu *E-Health Shield V2.0*. *E-Health Shield V2.0* berfungsi untuk mengakuisisi data *vital sign* yang terintegrasi dengan sensor kesehatan.

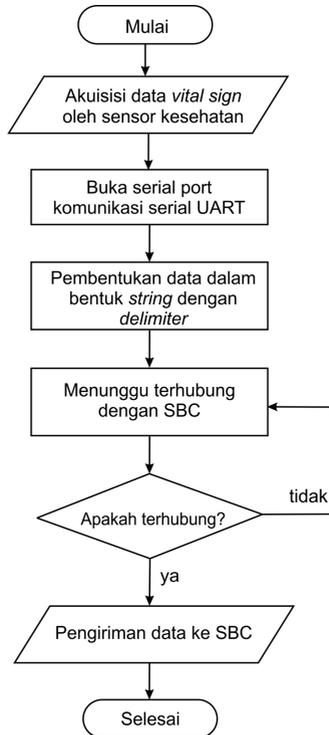
Program sistem mikrokontroler menggunakan *library E-Health Shield* untuk pengambilan data dari sensor. Kemudian dilakukan pengiriman data ke *single board computer* melalui komunikasi serial. Agar terhubung dengan *single board computer* diperlukan konfigurasi pada program untuk mendefinisikan pengaturan *port Rx* dan



**Gambar 3.3:** *Packing* alat dan sensor

*Tx* komunikasi serial. Selain itu dilakukan konfigurasi *baudratanya*. Program diunggah ke EEPROM Arduino untuk dijadikan instruksi biner yang akan memproses data *input/output*. Program tersebut dapat digunakan berkali-kali selama tidak diprogram dengan program baru meskipun alat di-*restart*.

Diagram alir akuisisi dan transmisi data dijelaskan oleh gambar 3.4. Dari diagram alir tersebut akuisisi data *vital sign* dilakukan oleh mikrokontroler (*sensor node*) dan melakukan pengiriman data ke penerima (SBC). Alur kerjanya yaitu mulanya data diakuisisi oleh sensor-sensor kesehatan. Data *vital sign* yang diakuisisi oleh sensor, mulanya berbentuk data *raw* yang kemudian diproses pada *library* e-Health menjadi data *vital sign* yang umum digunakan. Lalu program membuka *port* serial UART. Data yang terakuisisi dibuat dalam suatu deret data bertipe data *string*. Untuk membedakan data satu dengan yang lain, ditambahkan *delimiter* tertentu pada proses pembentukan deret data. Jika sensor *node* terhubung dengan SBC maka data dikirim secara berkala. Interval pengiriman



**Gambar 3.4:** Diagram alir akuisisi data dan transmisi data ke SBC

data yaitu 20 milidetik, sehingga ada 50 data yang terkirim dalam selang waktu satu detik.

### 3.3 Implementasi Komunikasi Serial UART

Guna mentransmisikan data yang telah diakuisisi oleh sensor node, sistem monitoring ini mengimplementasi komunikasi serial UART untuk proses pengiriman data dari mikrokontroler ke *single board computer*. Komunikasi serial menggunakan *port Rx* dan *Tx* pada masing-masing perangkat. Antara perangkat pengirim dan penerima, pengaturan *port* harus sama, seperti pengaturan.

Komunikasi serial yang digunakan komunikasi searah saja (*sim-*

*plex*). Data dari mikrokontroler dikirim secara terus menerus tanpa adanya timbal balik dari penerima (SBC), karena tidak ada tujuan untuk mengirim data dari SBC ke mikrokontroler.

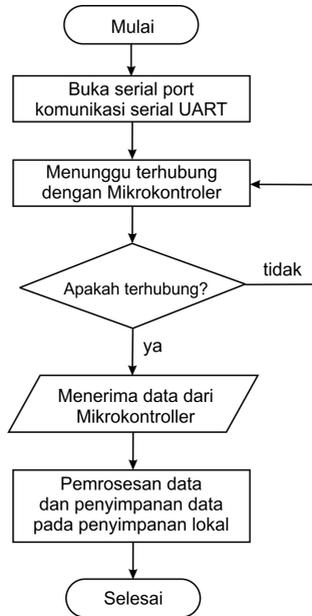
Pengaturan serial komunikasi pada sistem menggunakan *baudrate* sebesar 115200 *bit/s*, artinya dalam waktu satu detik dapat mengirimkan data sebesar 115200 bit. Pada sisi pengirim (mikrokontroler) digunakan *pinout* digital nomor 11 sebagai *Tx* dan *pinout* digital nomor 10 sebagai *Rx*. Pada sisi penerima (SBC) digunakan *pinout* GPIO (*General-purpose input/output*) nomor 15 sebagai *Rx* dan *pinout* GPIO 14 sebagai *Tx*.

### 3.4 Program *Receiver* dan Penyimpanan Lokal

Data yang dikirimkan oleh sensor *node* diterima SBC dengan program *receiver*. Program *receiver* pada sistem monitoring ini digunakan untuk menerima data dari mikrokontroler yang telah mengakuisisi data *vital sign*. Lalu memeriksa data dan menyimpan data pada penyimpanan lokal. Ada beberapa proses yang akan dilakukan pada sisi *receiver* ini yakni membuka port serial UART, periksa data yang terkirim, dan menyimpan data.

Diagram alir pada gambar 3.5) menjelaskan alir kerja sistem penerimaan data dan penyimpanan sementara oleh SBC. Program *receiver* mulanya membuka *port* UART sesuai dengan pengaturan yang terprogram pada sensor *node*. *Receiver* menunggu terhubung dengan sensor *node*. Lalu jika terhubung maka *receiver* siap menerima data dan data tersebut disimpan pada penyimpanan lokal.

Data yang masuk dilanjutkan dengan pemeriksaan terhadap data. Awalnya data berbentuk sekumpulan data lalu dilakukan pemeriksaan apakah data tersebut lengkap. Jika data hilang sebagian atau semua maka data tidak akan disimpan, namun jika valid maka data disimpan pada penyimpanan lokal. Selain itu sekaligus data diuraikan menjadi satu per satu data *vital sign*. Penguraian ini digunakan untuk menampilkan *vital sign* pada *user interface desktop*. Jika terdapat koneksi yang stabil, data yang tersimpan pada penyimpanan lokal, diunggah ke *cloud storage* melalui protokol HTTP.



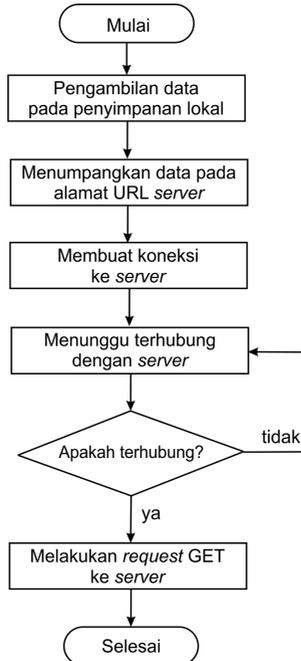
**Gambar 3.5:** Diagram alir *receive* data dan penyimpanan data oleh SBC

### 3.5 Penyimpanan ke *Cloud Server* Berbasis Protokol HTTP

Pada penyimpanan data ke *cloud storage* digunakan protokol HTTP yang mana mendukung metode *request* GET, POST, PUT, dan lain sebagainya. Selain *request* dari *client*, *server* juga merespon *request* tersebut dengan beberapa kode *response* seperti kode 200, 400, 404, dan lain sebagainya.

Gambar 3.6 menjelaskan alir kerja penyimpanan data ke *cloud storage*. Pada proses kerjanya mulanya data yang tersimpan pada penyimpanan lokal dibuka kembali. Data dibaca per baris hingga akhir data terekam. Lalu dibuat sebuah URL yang mana tertuju pada APIs yang telah dibuat di *server*. Proses selanjutnya program membuat koneksi ke *server*. Jika terhubung dengan *server* data dikirim per baris yang terbaca. Dalam proses tersebut dilakukan

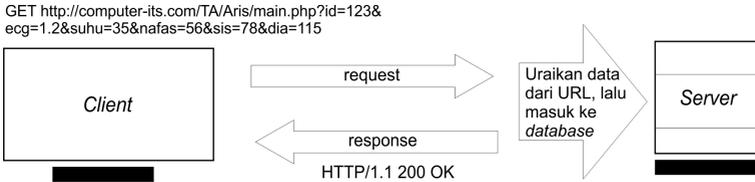
pengiriman dengan interval 100 milidetik per pengiriman. Jika pengunggahan berhasil maka, *server* membalas dengan kode 200 yang artinya *successful* (berhasil diterima dan diproses).



**Gambar 3.6:** Diagram alir penyimpanan data ke *cloud storage*

Pengunggahan data terlebih dahulu dilakukan dengan membuat *request* ke *server*. *Request* tersebut menggunakan *method* GET pada *port* 80. Data diunggah dengan cara menumpangkan data ke URL *server* yang dituju. Gambar 3.7 menjelaskan proses GET *request* pada protokol HTTP dengan data yang ditumpangkan. Pada bagian *response* dari *server* terdapat kode 200, ini menandakan bahwa *request* yang dilakukan sukses diterima server dan dilayani oleh server.

Pada saat melakukan *request* ke *server* yang dilakukan *client* yaitu mengirimkan *header* pada *method* GET sebagai berikut:



**Gambar 3.7:** HTTP GET *request*

```
GET /TA/Aris/main.php?tag=insert&id=340805132093008 & time_t =2017-04-11+15:00:00&ecg=2&suhu=37&nafas=100&dia =60&sys=100
HTTP/1.1
Host: computer-its.com
Cache-Control: no-cache
Postman-Token: b262f1ad-80fe-7e1f-9160-061d8c11b39e
```

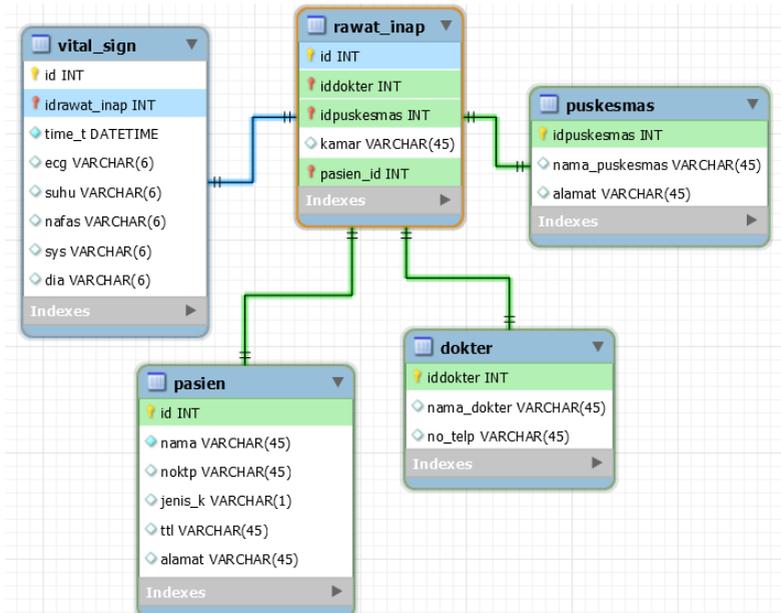
Lalu *server* merespon *request* tersebut dengan *header* sebagai berikut:

```
Accept-Ranges :bytes
Connection :keep-alive
Content-Length :18
Content-Type :text/html
Date :Mon, 08 May 2017 15:39:29 GMT
Server :LiteSpeed
Via :1.0 proxy-68.its.ac.id:8080
(squid/2.7.STABLE9), 1.0 b401 (squid/3.1.20)
X-Cache :MISS from proxy-68.its.ac.id, MISS from b401
X-Cache-Lookup :MISS from proxy-68.its.ac.id:8080, MISS from b401:8080
X-Powered-By :PHP/5.5.38
```

### 3.6 Desain Database Pada Server

Pada sistem ini, database digunakan untuk menyimpan data-data *vital sign* pasien yang telah dikirim dari *single board computer* ke *cloud storage*. Database yang digunakan jenis database relasi ERD (Entity Relation Database), yang mana antara tabel dapat berelasi dengan tabel lain dan tabel tersebut tidak bersifat fleksibel. Desain *database* ini berguna untuk mengurangi tingkat redundansi

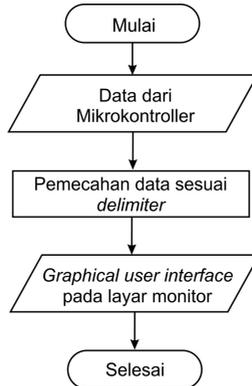
data, salah satunya dengan cara normalisasi *database*.



**Gambar 3.8:** Desain *Entity Relation Database*

Pada implementasinya, *database* yang dibuat dibagi menjadi lima tabel, yaitu tabel data pasien, tabel data *vital sign* pasien, tabel puskesmas, tabel dokter, dan tabel rawat inap. Desain *database* tersebut dijelaskan pada gambar 3.8. Penjelasan tabel-tabel tersebut yaitu tabel pasien berisi data identitas pasien yang terdaftar memakai alat di puskesmas. Di dalam tabel tersebut mendeskripsikan tentang identitas pasien, mulai dari id pasien, nama pasien, alamat pasien, di puskesmas mana pasien dirawat, dan lain sebagainya. Tabel yang kedua yaitu data *vital sign*. Di dalam tabel *vital sign* berisi rekaman hasil dari monitoring pasien berdasarkan id rawat inap pasien. Pada tabel puskesmas berisi tentang identitas puskesmas yang bertanggungjawab menangani pasien. Tabel selanjutnya yaitu tabel dokter yang berisi identitas dokter yang bekerja di puskesmas. Tabel terakhir yaitu tabel rawat inap, tabel ini berisi

gabungan id dari tabel pasien, dokter, dan puskesmas yang berguna untuk referensi data pada tabel *vital sign*.



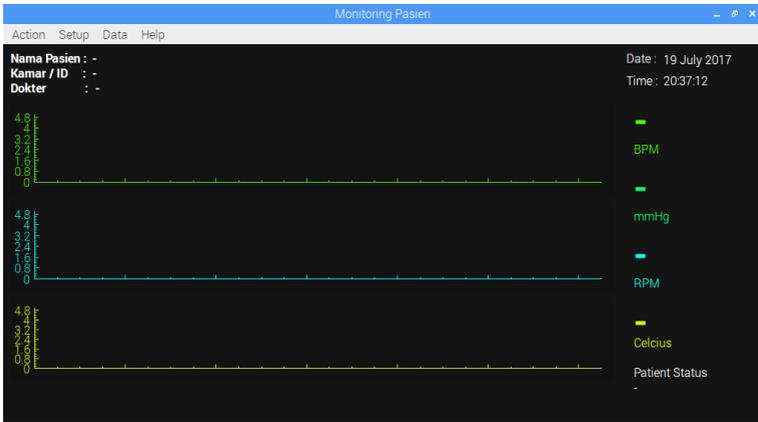
**Gambar 3.9:** Diagram alir visualisasi data

### 3.7 User Interface Berbasis Desktop

Pada dasarnya sisi user interface dibangun secara satu kesatuan dengan pemrograman *receiver*. Sehingga sekali data masuk akan ditampilkan pada sisi *user interface* sekaligus data disimpan pada penyimpanan lokal. Gambar 3.9 menjelaskan cara kerja visualisasi data. Proses visualisasi data pada program *receiver* diproses setelah data tersimpan pada lokal. Mulanya data dari sensor *node* diterima oleh *receiver*. Lalu data tersebut dilakukan pemecahan (*split data*) sesuai *delimiter* yang telah terprogram pada sensor *node*. Dari hasil pemecahan data didapatkan empat data penting yang nilainya divisualisasikan pada tampilan *desktop*.

*User interface* dibuat dari layout yang dibangun dengan bahasa *xml*. *Layout* dari *interface* berhubungan dengan pemrograman *receiver*. Pemrograman *receiver* pada dasarnya sudah termasuk mengontrol bagian *user interface*. Gambar 3.10 merupakan implementasi tampilan utama *user interface* yang dibuat.

*User interface* tersebut dibuat seinteraktif mungkin, guna mudah dipahami oleh paramedis. Pada *user interface* terdapat elemen-elemen penting guna pemantauan *vital sign* yaitu:



**Gambar 3.10:** Tampilan utama *user interface monitoring vital sign*

1. Identitas pasien - Identitas pasien mempresentasikan data pasien yang terpantau yakni meliputi nama pasien, nomor ID pasien, alamat pasien, dan nomor ruang pasien dirawat inap
2. *Real-time vital sign* - Keempat *vital sign* divisualisasikan dalam bentuk grafik yang sesuai nilainya dengan yang diakuisisi oleh sensor
3. Blok menu - blok menu berisi menu apa saja yang dapat digunakan guna menunjang pemantauan
4. Sistem peringatan - Peringatan akan muncul ketika nilai *vital sign* diluar batas yang telah diatur pada menu *setup*. Peringatan berupa visual dan audio. Jika terdapat peringatan maka, warna angka dan grafik *vital sign* berwarna merah dan sistem mengeluarkan bunyi tanda tidak normal.

Guna melakukan pengaturan terhadap *user interface* maka, menu yang dibuat meliputi menu *action*, menu *setup* pasien, menu *data*, dan menu *help*. Pada menu *action* terdapat perintah untuk menjalankan sistem dan juga menghentikannya. Pada menu *setup* terdapat pengaturan yang digunakan untuk mengatur identitas pasien dan mengatur normal *vital sign* pasien. Pada menu *setup* data identitas pasien, paramedis dapat melakukan pendataan identitas pasien yang akan dipantau seperti nama, alamat, jenis kelamin, kamar pasien dan lain sebagainya. Sedangkan pada menu *set normal*

*vital sign* (atur *vital sign* normal pasien), paramedis dapat mengatur batas bawah BPM pasien dan batas atasnya, batas bawah dan atas suhu pasien, batas bawah dan atas tekanan darah, dan batas bawah dan atas hembusan nafas per menit.

Data Laporan Vital Sign Pasien				
pilih puskesmas >> Medika Indah				
pilih pasien >> Adrian				
pilih tanggal >> 20 Mei 2017				
pilih jam >> 06:00 - 07:00				
BPM	SUHU	RPM	BP	ECG
70	36.5	16	120/80	<a href="#">link</a>

Gambar 3.11: Desain *user interface* laporan

**Sistem Informasi Monitoring Pasien**  
 Sabtu, 10 Juni 2017 13:00:13

Data Laporan Pasien

Puskesmas Medika Indah

31 - Adrian K. Rahardjo

08 May 2017

12:00 - 13:00

Ok

BPM	SUHU	RPM	BP	ECG
76	36.6	15	118/75	<a href="#">View</a>
76	36.5	13	118/75	<a href="#">View</a>
66	36.8	23	118/75	<a href="#">View</a>
63	36.6	15	118/75	<a href="#">View</a>
76	36.6	15	118/75	<a href="#">View</a>
70	35.7	18	118/75	<a href="#">View</a>

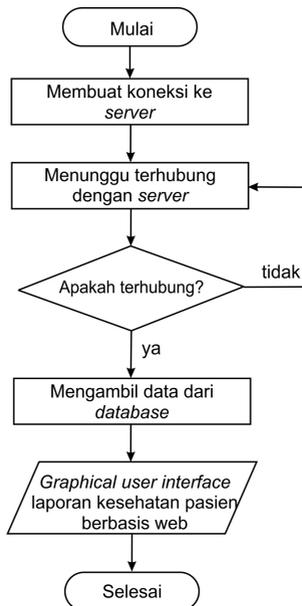
Gambar 3.12: Implementasi laporan berbasis web

### 3.8 Laporan *Vital Sign*

Pembuatan laporan *vital sign* digunakan untuk memvisualisasikan laporan kondisi pasien pada waktu yang diinginkan oleh dokter. Sistem yang dibuat dilakukan pengambilan data *vital sign* dari

*database server* yang telah menyimpan data *vital sign* dari alat pemantau. Data yang ditampilkan dikelompokkan berdasarkan waktu per jam. Hasil dari pengelompokan per jam dibuat *list* data per menit. Gambar 3.11 merupakan desain dari laporan *vital sign* yang mana terdiri dari elemen penting yaitu pilihan data yang akan ditampilkan dan hasil data yang ditampilkan sesuai pilihan.

Diagram alir pada gambar 3.13 menjelaskan cara kerja pembuatan laporan *vital sign* berbasis web. Mulanya membuat koneksi ke *server* lalu jika terhubung maka, sistem mengambil data dari *database*. Data yang diambil merupakan data yang diinginkan oleh pengguna (dokter). Program memilah data sesuai kueri yang diminta sebagai contoh menampilkan *vital sign* pasien "A" yang berada di puskesmas "B" pada jam enam.



**Gambar 3.13:** Diagram alir visualisasi laporan

Laporan *vital sign* dibuat seinteraktif mungkin guna mudah dipahami oleh dokter. Laporan *vital sign* tersebut berbasis web dengan mengimplementasikan pemrograman PHP (*HyperText Pre-*

*processor*) sebagai *back end*. Gambar 3.12 merupakan hasil implementasi dari pembuatan laporan *vital sign*. Laporan tersebut dapat diakses oleh dokter dimana pun dokter berada dan kapan saja. Untuk mengakses laporan dokter harus *log in* terlebih dahulu dengan autentikasi tertentu. Lalu dokter memilih nomor identitas pasien yang ingin divisualisasikan kondisinya.

## BAB 4

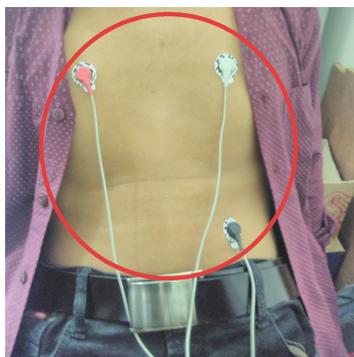
# PENGUJIAN DAN ANALISA

Pada bab ini, dilakukan pengujian terhadap sistem yang telah dibuat. Pengujian dilakukan guna mengetahui tingkat kesalahan dan menarik kesimpulan dari sistem yang telah dibuat. Pengujian yang dilakukan meliputi pengujian *receiver*(SBC), validasi data hasil akuisisi, perbandingan sensor, penyimpanan data ke *cloud storage*, pengujian terhadap *throughput* yang digunakan, dan pengujian *user interface*.

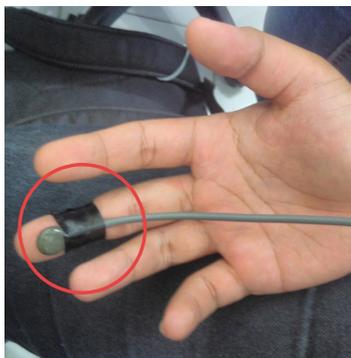
Sebelum dilakukan pengujian terhadap sistem, dilakukan pemasangan sensor-sensor pada tubuh pengguna guna mendapat data *vital sign* pengguna. Pemasangan dilakukan sesuai pada dokumentasi yang ada. Gambar 4.1 merupakan pemasangan sensor pada tubuh pengguna. Masing-masing sensor ditunjukkan oleh lingkaran berwarna merah. Pemasangan sensor ECG (gambar 4.1a) ditempatkan pada tiga titik di dada, pemasangan sensor suhu (gambar 4.1b) ditempatkan pada jari telunjuk, pemasangan sensor nafas (gambar 4.1c) ditempatkan tepat di depan lubang hidung, dan sensor tekanan darah (gambar 4.1d) pada lengan atas. Pemasangan sensor-sensor kesehatan tersebut berdasarkan dokumentasi dari *E-Health Shield*.

### 4.1 Pengujian *Receiver*

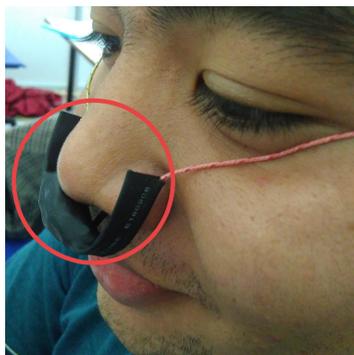
Pada pengujian ini dilakukan guna menguji seberapa persen data yang masuk dari akuisisi sensor kemudian dikirim ke SBC. Pengujian dilakukan dengan cara membuka *port UART* pada SBC lalu menerima data yang dikirim oleh mikrokontroler. Sebelumnya dibuat program *receiver* dan menjalankan program tersebut dengan menambahkan perintah *timeout* lalu waktu yang diperlukan (Gambar 4.2). Dari program tersebut dijalankan sebanyak tiga kali percobaan dengan waktu yang berbeda-beda. Sehingga ada 15 kali percobaan pada pengujian ini. Jika pada program *transmitter* data yang dikirim dalam satu detik ada 50 data maka, data yang dikirim dalam satu detik sebanyak 50 data. Data bertambah sesuai



(a) Pemasangan sensor ECG



(b) Pemasangan sensor suhu



(c) Pemasangan sensor nafas



(d) Pemasangan sensor tekanan darah

**Gambar 4.1:** Pemasangan sensor-sensor kesehatan

bertambahnya waktu.

Dari hasil pengujian pada gambar 4.2 dapat diketahui rata-rata data yang berhasil terkirim dari tiga kali percobaan dengan selang waktu yang berbeda. Hasil tiap pengujian dijelaskan pada tabel 4.1. Dari tabel tersebut pengujian yang dilakukan dalam selang waktu satu detik rata-rata terdapat 43 data yang berhasil terkirim dari 50 data yang seharusnya terkirim. Pada selang dua detik rata-rata data yang berhasil terkirim sebanyak 86 data da-

```

pi@raspberrypi:~/Akhir $ timeout 1s ./serialRead
2.85,29,172,118,75
1.84,29,129,118,75
1.20,30,164,118,75
1.98,29,290,118,75
3.15,30,338,118,75
3.92,30,257,118,75
3.26,30,192,118,75
2.15,29,145,118,75
1.33,30,135,118,75

```

**Gambar 4.2:** Data yang berhasil terkirim dalam selang waktu satu detik

**Tabel 4.1:** Tabel pengujian data terkirim terhadap selang waktu

Selang waktu	Data terkirim pada pengujian ke			Data seharusnya terkirim	Rata-rata keberhasilan
	1	2	3		
1 detik	43	44	43	50	86%
2 detik	86	86	86	100	86%
3 detik	130	130	130	150	86.6%
4 detik	172	173	173	200	86.5%
5 detik	215	216	215	250	86%
<b>Keseluruhan pengujian</b>					86.22%

ri 100 data yang harus terkirim dan seterusnya sebagaimana hasil pengujian yang dijelaskan pada tabel. Dari keseluruhan pengujian didapatkan rata-rata persentase keberhasilan data yang masuk yaitu sebesar 86.22% dari data yang seharusnya terkirim. Sehingga dapat dikatakan pengiriman data dari sensor *node* ke SBC cukup baik.

## 4.2 Validasi Data *Vital Sign*

Pengujian ini dilakukan guna mengetahui apakah data yang terpantau di SBC benar-benar data yang diakuisisi oleh sensor *node*. Pada rancangan alat digunakan komunikasi serial untuk mentransmisikan data dari mikrokontroler ke SBC. Untuk pengujiannya dilakukan dua macam cara pengujian, yaitu yang pertama dengan cara membuat *log* pada masing-masing *hardware* baik disisi mikrokontroler maupun SBC dan yang kedua dengan cara *labeling*.

Proses cara pertama yaitu data dari sensor dibuat sebuah *log file* ketika data tersebut sebelum dikirim ke SBC. Jika *log* atau data sudah tersimpan pada *log file* maka, data dikirim ke SBC. Pada SBC data yang diterima juga disimpan pada sebuah *log file*. Sehingga

ga didapat tingkat validasi data dengan cara membandingkan data sebelum terkirim dan data sesudah terkirim. Cara yang kedua yaitu dengan cara melabeli setiap data yang akan terkirim. Proses cara ini adalah data yang diakuisisi sensor ditambahkan sebuah label misalnya nomor urut. Jika data yang terpantau dengan *sampling* 50 data dalam satu detik maka, dilakukan penambahan nomor urut satu sampai 50 didepan data yang akan dikirim. Lalu nomor urut akan kembali ke satu lagi begitu pun seterusnya. Pada sisi penerima data disimpan pada *log*. Sehingga didapat data yang berurutan satu sampai 50. Jika ada penambahan data atau data tidak urut maka terjadi *error* dalam validasinya.

[1] 3.64,28.5,3,118,75	[26] 1.26,28.5,3,118,75	[1] 3.63,28.4,3,118,75	[26] 1.93,28.4,2,118,75
[2] 2.38,28.4,4,118,75	[27] 1.75,28.4,6,118,75	[2] 3.55,28.5,3,118,75	[27] 1.14,28.5,3,118,75
[3] 1.45,28.4,4,118,75	[28] 3.09,28.4,2,118,75	[3] 2.29,28.4,2,118,75	[28] 1.99,28.4,2,118,75
[4] 1.37,28.5,3,118,75	[29] 3.98,28.3,4,118,75	[4] 1.40,28.4,3,118,75	[29] 3.29,28.4,3,118,75
[5] 2.65,28.3,1,118,75	[30] 2.86,28.6,3,118,75	[5] 1.46,28.5,3,118,75	[30] 3.90,28.2,1,118,75
[6] 3.70,28.4,3,118,75	[31] 1.74,28.6,3,118,75	[6] 2.76,28.5,3,118,75	[31] 2.63,28.4,3,118,75
[7] 3.43,28.6,3,118,75	[32] 1.15,28.6,3,118,75	[7] 3.77,28.6,3,118,75	[32] 1.58,28.6,3,118,75
[8] 2.20,28.4,3,118,75	[33] 2.28,28.4,3,118,75	[8] 3.33,28.4,3,118,75	[33] 1.25,28.6,3,118,75
[9] 1.32,28.4,3,118,75	[34] 3.49,28.4,3,118,75	[9] 2.11,28.5,3,118,75	[34] 2.52,28.6,3,118,75
[10] 1.55,28.6,3,118,75	[35] 3.71,28.3,3,118,75	[10] 1.27,28.4,1,118,75	[35] 3.66,28.5,3,118,75
[11] 2.91,28.5,3,118,75	[36] 2.39,28.4,3,118,75	[11] 1.70,28.4,4,118,75	[36] 3.46,28.5,3,118,75
[12] 3.89,28.4,3,118,75	[37] 1.45,28.4,3,118,75	[12] 3.05,28.6,3,118,75	[37] 2.18,28.5,2,118,75
[13] 3.05,28.4,3,118,75	[38] 1.44,28.5,3,118,75	[13] 3.96,28.5,3,118,75	[38] 1.29,28.5,3,118,75
[14] 1.87,28.5,3,118,75	[39] 2.83,28.4,3,118,75	[14] 2.90,28.5,3,118,75	[39] 1.67,28.5,3,118,75
[15] 1.13,28.5,3,118,75	[40] 3.82,28.4,4,118,75	[15] 1.76,28.6,3,118,75	[40] 3.04,28.5,3,118,75
[16] 2.07,28.4,3,118,75	[41] 3.19,28.4,3,118,75	[16] 1.14,28.5,3,118,75	[41] 3.95,28.4,7,118,75
[17] 3.34,28.4,4,118,75	[42] 1.98,28.4,5,118,75	[17] 2.23,28.5,3,118,75	[42] 2.94,28.5,4,118,75
[18] 3.85,28.4,2,118,75	[43] 1.16,28.5,4,118,75	[18] 3.47,28.4,4,118,75	[43] 1.78,28.4,3,118,75
[19] 2.56,28.5,1,118,75	[44] 1.93,28.4,4,118,75	[19] 3.72,28.4,11,118,75	[44] 1.13,28.7,3,118,75
[20] 1.54,28.6,3,118,75	[45] 3.24,28.4,3,118,75	[20] 2.38,28.4,1,118,75	[45] 2.20,28.6,3,118,75
[21] 1.30,28.6,3,118,75	[46] 3.93,28.3,4,118,75	[21] 1.42,28.6,3,118,75	[46] 3.44,28.5,3,118,75
[22] 2.60,28.6,3,118,75	[47] 2.68,28.4,1,118,75	[22] 1.46,28.5,3,118,75	[47] 3.75,28.4,3,118,75
[23] 3.70,28.5,3,118,75	[48] 1.61,28.5,3,118,75	[23] 2.80,28.4,3,118,75	[48] 2.44,28.5,3,118,75
[24] 3.38,28.4,3,118,75	[49] 1.23,28.6,3,118,75	[24] 3.75,28.4,10,118,75	[49] 1.47,28.4,11,118,75
[25] 2.12,28.5,3,118,75	[50] 2.48,28.5,3,118,75	[25] 3.13,28.4,4,118,75	[50] 1.42,28.4,4,118,75

**Gambar 4.3:** Data terkirim pengujian *labeling* data

Dari cara pengujian pertama yaitu dengan cara *labeling* didapatkan hasil *log* data yang urut (Gambar 4.3). Pada gambar diketahui bahwa dari awal hingga akhir data yang terkirim per detik tidak terdapat data yang terselip maupun data yang hilang pada

<i>Transmitter</i>	<i>Receiver</i>	Validasi
4.26,31,874,118,75	4.26,31,874,118,75	valid
4.64,31,867,118,75	4.64,31,867,118,75	valid
4.38,32,866,118,75	4.38,32,866,118,75	valid
4.28,32,873,118,75	4.28,32,873,118,75	valid
4.42,32,875,118,75	4.42,32,875,118,75	valid
4.84,32,876,118,75	4.84,32,876,118,75	valid
4.98,32,869,118,75	4.98,32,869,118,75	valid
4.52,32,869,118,75	4.52,32,869,118,75	valid
4.38,31,866,118,75	4.38,31,866,118,75	valid
4.26,32,875,118,75	4.26,32,875,118,75	valid
4.54,32,875,118,75	4.54,32,875,118,75	valid
4.88,31,875,118,75	4.88,31,875,118,75	valid
4.97,32,867,118,75	4.97,32,867,118,75	valid
4.44,32,867,118,75	4.44,32,867,118,75	valid
4.35,32,869,118,75	4.35,32,869,118,75	valid
4.27,32,874,118,75	4.27,32,874,118,75	valid
4.65,32,874,118,75	4.65,32,874,118,75	valid
4.91,32,872,118,75	4.91,32,872,118,75	valid
4.86,31,867,118,75	4.86,31,867,118,75	valid
4.37,31,867,118,75	4.37,31,867,118,75	valid
4.32,31,871,118,75	4.32,31,871,118,75	valid
4.30,32,874,118,75	4.30,32,874,118,75	valid
4.75,31,877,118,75	4.75,31,877,118,75	valid
4.96,32,869,118,75	4.96,32,869,118,75	valid
4.69,32,866,118,75	4.69,32,866,118,75	valid
4.36,32,867,118,75	4.36,32,867,118,75	valid

**Gambar 4.4:** Perbandingan *log* data sebelum dikirim dan sesudah terkirim

urutan tertentu. Lalu pada cara pengujian kedua yaitu dengan cara menyimpan *log* data di mikrokontroler (sebelum pengiriman data) dan di SBC (data terkirim). Gambar 4.4 merupakan *log* data yang tersimpan pada sisi pengirim dan *log* data yang terkirim pada sisi penerima. Pengujian ini dilakukan sebanyak lima kali dalam selang waktu 10 detik atau terdapat 500 data dalam pengirimannya. Dari lima kali pengujian semua data tersebut jika dibandingkan maka, data yang tersimpan pada pengirim diketahui sama dengan data yang terkirim pada penerima. Dari awal *log* data hingga akhir data tidak terdapat data yang terselip maupun hilang. Sehingga pada pengujian validasi dengan kedua cara yang telah dilakukan dapat dikatakan data yang terkirim ke sisi SBC merupakan data valid dari sensor *node* (mikrokontroler).

**Tabel 4.2:** Tabel pengujian ECG

Pengujian ke	Sensor E-Health	Hitung Manual	Galat
1	59 bpm	63 bpm	6.35%
2	35 bpm	69 bpm	49.28%
3	56 bpm	71 bpm	21.13%
4	65 bpm	68 bpm	4.41%
5	66 bpm	67 bpm	1.49%
<b>Rata-rata</b>			<b>16.53%</b>

### 4.3 Perbandingan Sensor

Pengujian perbandingan sensor dilakukan untuk mengetahui galat pada data yang diakuisisi oleh sensor yang terpasang dengan data alat lain atau manual. Pada pengujianya dilakukan terhadap empat buah sensor.

#### 4.3.1 Sensor ECG

Pengujian pada sensor ECG digunakan untuk mengetahui ketepatan perhitungan detak jantung per menit. Pengujian dilakukan dengan membandingkan hasil perhitungan sistem dan secara manual. Pengujian dilakukan selama satu menit per sesi uji dan dilakukan sebanyak lima kali. Tabel 4.2 merupakan hasil uji terhadap sensor ECG.

Dari hasil pengujian sebanyak lima kali (Tabel 4.2) didapatkan hasil galat terendah pada pengujian kelima yaitu sebesar 1.49%. Sedangkan galat tertinggi terjadi pada pengujian kedua yaitu sebesar 49.28%. Dari keseluruhan pengujian didapat rata-rata galat yaitu sebesar 16.53%. Sehingga dapat dikatakan hasil perhitungan sistem terhadap nilai BPM masih kurang tepat untuk melakukan perhitungan.

#### 4.3.2 Sensor Suhu

Pengujian pada sensor suhu dilakukan guna mengetahui seberapa tepat pengukuran dari sistem. Pada pengujianya dilakukan dengan membandingkan hasil hitung suhu pengguna menggunakan termometer digital dengan hasil data yang diakuisisi oleh sensor suhu. Pada pengukuran menggunakan termometer digital cara pengukurannya secara *axillary* (pada ketiak pengguna). Tabel 4.3 me-

**Tabel 4.3:** Tabel pengujian sensor suhu

Pengujian ke	Sensor E-Health	Termometer digital	Galat
1	36.81 °C	35.20 °C	4.57%
2	36.43 °C	35 °C	4.09%
3	36.11 °C	35.60 °C	1.43%
4	36.40 °C	35.90 °C	1.39%
5	36.10 °C	35.50 °C	1.69%
<b>Rata-rata</b>			2.64%

**Tabel 4.4:** Tabel pengujian sensor nafas

Pengujian ke	Sensor <i>E-Health</i>	Hitung Manual	Galat
1	26 rpm	15 rpm	73.33%
2	16 rpm	15 rpm	6.67%
3	15 rpm	13 rpm	15.38%
4	15 rpm	13 rpm	15.38%
5	13 rpm	14 rpm	7.14%
<b>Rata-rata</b>			23.58%

rupakan hasil perbandingan dan galat dari perhitungan suhu pengguna.

Dari tabel tersebut didapatkan hasil galat terendah pada pengujian keempat yaitu sebesar 1.39% dan galat tertinggi sebesar 4.57%. Pada hasil pengukurannya nilai yang diakuisisi oleh sensor selalu lebih tinggi dibandingkan dengan termometer digital. Dari hasil lima kali uji didapatkan rata-rata galat sebesar 2.64%. Sehingga dapat dikatakan sensor bekerja dengan baik dan dapat menggantikan pengukuran termometer digital.

### 4.3.3 Sensor Nafas

Pengujian pada sensor nafas dilakukan untuk mengetahui berapa jumlah nafas yang terhitung oleh sensor dibandingkan perhitungan secara manual. Pada sistem, setiap hembusan nafas dengan *threshold* tertentu dikatakan satu kali nafas. Tabel 4.4 menjelaskan hasil dan galat dari perbandingan uji sensor nafas dengan hitung manual.

Dari hasil pengujian sebanyak lima kali didapat bahwa pada pengujian galat tertinggi sebesar 73.33% dan galat terendah sebe-

sar 6.67%. Dari hasil keseluruhan didapat rata-rata galat sebesar 23.58%. Sehingga dapat dikatakan sistem perhitungan masih belum optimal. Hal tersebut dikarenakan nilai *threshold* setiap hembusan tidak dapat diprediksi untuk dilakukan perhitungan.

#### 4.3.4 Sensor Tekanan Darah

Pada pengujian tekanan darah hanya dilakukan uji terhadap data yang terekam pada *sphygmomanometer* digital dengan data yang tervisualisasi pada *user interface*. Pada dasarnya data yang diakuisisi oleh sensor *node* tidak mengalami pemrosesan data lanjut seperti sensor suhu, ECG, dan nafas.

### 4.4 Pengujian Penyimpanan ke *Cloud Storage*

Pada bagian ini, dilakukan pengujian untuk memastikan apakah data berhasil tersimpan pada penyimpanan ke *cloud*. Pengujian dimulai dari masuknya data ke SBC. Data lalu tersimpan pada penyimpanan lokal. Data disimpan dalam bentuk *file*. Jika dilakukan proses pengiriman data ke *cloud storage* data yang sedang terekam akan dihentikan. Lalu sistem membuat data *log* baru.

Pada *log file* yang pertama, *file* dibuka guna mengambil data yang telah tersimpan. Data yang diunggah dibaca per baris. Lalu untuk melakukan pengunggahan data digunakan *method GET request* pada protokol HTTP. Data yang terbaca dari file dilakukan proses *looping* hingga akhir baris. Tiap pembacaan satu bari data dilakukan penguraian terhadap data menjadi bentuk URL. Cara pengiriman data *method GET*, data ditumpangkan pada URL misalnya "[http://computer-its.com/TA/Aris/main.php?tag=insert&id=31&time\\_t=12:00:00 20:00&ecg=0.23&suhu=34&nafas=25&dia=100&sys=70](http://computer-its.com/TA/Aris/main.php?tag=insert&id=31&time_t=12:00:00 20:00&ecg=0.23&suhu=34&nafas=25&dia=100&sys=70)".

Setiap sukses melakukan pengiriman data ke *database server*, pada terminal akan menampilkan kode 200 dan tulisan "*success*". Kode tersebut merupakan respon yang diberikan oleh server dari *request*. Jika gagal maka akan menampilkan status *error*. Sukses dan gagal dalam pengiriman ditunjukkan pada Gambar 4.5. Pada pengujian ini dilakukan percobaan pengunggahan sebanyak lima kali. Data yang dikirim berturut-turut sebanyak 3000, 6000, 9000, 12000, dan 15000 data. Data tersebut merupakan data yang telah terekam oleh sensor *node*.

```

GET http://computer-its.com/TA/Aris/main.php?tag=insert&id=31&time_t=2017-05-28
12:00:00&ecg=3.2&suhu=35.6&nafas=31&sys=118&dia=75
200 OK "success"
GET http://computer-its.com/TA/Aris/main.php?tag=insert&id=31&time_t=2017-05-28
12:00:00&ecg=1.1&suhu=35.6&nafas=31&sys=118&dia=75
200 OK "success"
GET http://computer-its.com/TA/Aris/main.php?tag=insert&id=31&time_t=2017-05-28
12:00:00&ecg=0.8&suhu=35.6&nafas=31&sys=118&dia=75
200 OK "success"
GET http://computer-its.com/TA/Aris/main.php?tag=insert&id=31&time_t=2017-05-28
12:00:00&ecg=2.2&suhu=35.6&nafas=32&sys=118&dia=75
200 OK "success"
GET http://computer-its.com/TA/Aris/main.php?tag=insert&id=31&time_t=2017-05-28
12:00:00&ecg=2.1&suhu=35.6&nafas=30&sys=118&dia=75
200 OK "success"
GET http://computer-its.com/TA/Aris/main.php?tag=insert&id=31&time_t=2017-05-28
12:00:00&ecg=2.1&suhu=35.6&nafas=30&sys=118&dia=75
200 OK "success"

```

**Gambar 4.5:** Proses penyimpanan data ke *cloud storage*

Proses per pengiriman data ke *server* dilakukan dengan selang waktu 100 milidetik. Dari hasil lima kali pengujian pengunggahan data ke *database*, didapatkan bahwa data keseluruhan pada lima kali pengujian tersebut berhasil tersimpan pada *database*. Namun semakin banyak data yang disimpan ke *cloud storage* maka, semakin lama waktu yang dibutuhkan untuk proses penyimpanan. Untuk mengetahui durasi yang dibutuhkan dapat dicari dengan cara mengalikan data dengan waktu per pengiriman. Jika data sebanyak 3000 maka, waktu yang dibutuhkan  $3000 \times 0.1 \text{detik} = 300 \text{detik}$ . Sehingga dapat disimpulkan bahwa persentase pengiriman yakni 100% namun waktu yang dibutuhkan untuk proses penyimpanan tergantung jumlah data yang disimpan.

## 4.5 Pengujian Throughput

Pengujian ini dilakukan guna mengetahui seberapa cepat-an *upload* data yang dibutuhkan saat mengunggah data ke *database server*. Sebelumnya dilakukan proses sinkronisasi data ke *cloud storage* dan instalasi program "nload" pada SBC. Lalu dijalankan program "nload" dan terlihat beberapa informasi mengenai *bandwidth* yang dibutuhkan saat proses sinkronisasi data. Pengujian throughput dilakukan pada enam jenis yaitu:

1. *Throughput downlink* rata-rata
2. *Throughput downlink* maksimal
3. *Throughput downlink* total
4. *Throughput downlink* rata-rata
5. *Throughput uplink* maksimal

**Tabel 4.5:** Tabel pengujian *throughput* rata-rata

Pengujian ke	<i>Throughput</i> rata-rata	
	Uplink	Downlink
1	168 Bit/s	3.32 kBit/s
2	280 Bit/s	3.35 kBit/s
3	216 Bit/s	3.27 kBit/s
4	192 Bit/s	3.22 kBit/s
5	2005 Bit/s	4.57 kBit/s
<b>Rata-rata</b>	572.2 Bit/s	3.54 kBit/s

#### 6. *Throughput* downlink total

##### 4.5.1 *Throughput* rata-rata

Pada pengujian ini dilakukan untuk mengetahui seberapa besar rata-rata *throughput* yang masuk pada saat proses sinkronisasi data ke *cloud*. Tabel 4.5 menjelaskan pangujian *throughput* rata-rata sebanyak lima kali pengujian.

Dari tabel tersebut didapatkan *throughput* rata-rata *uplink* tertinggi terjadi pada saat pengujian kelima yaitu sebesar 2005 *Bit/s* begitu juga *throughput* rata-rata *downlink* tertinggi terjadi pada pengujian kelima yaitu sebesar 4.57 *kBit/s*. Sedangkan *throughput* rata-rata *uplink* terendah sebesar 168 *Bit/s* dan *throughput* rata-rata *downlink* terendah sebesar 3.22 *kBit/s*. Dari keseluruhan pengujian didapatkan rata-rata *throughput* rata-rata *uplink* sebesar 572.2 *Bit/s* sedangkan *throughput* rata-rata *downlink* sebesar 3.54 *kBit/s*.

##### 4.5.2 *Throughput* maksimal

*Throughput* maksimal merupakan besar *bandwidth* maksimal yang digunakan dalam proses sinkronisasi ke *cloud*. Tabel 4.6 menjelaskan pengujian terhadap *throughput* maksimal sebanyak lima kali pengujian.

Dari tabel pengujian tersebut didapatkan nilai *throughput* maksimal dengan nilai rata-rata *uplink* sebesar 8.89 *kBit/s* dan *downlink* sebesar 17.88 *kBit/s*. *Throughput* maksimal *uplink* terendah terjadi pada pengujian pertama yaitu sebesar 1.91 *kBit/s* dan tertinggi pada pengujian kelima yaitu sebesar 21.63 *kBit/s*. Sedangkan *throughput* maksimal *downlink* terendah sebesar 9.84 *kBit/s* dan tertinggi

**Tabel 4.6:** Tabel pengujian *throughput* maksimal

Pengujian ke	<i>Throughput</i> maksimal	
	Uplink	Downlink
1	1.91 kBit/s	9.84 kBit/s
2	6.98 kBit/s	12.71 kBit/s
3	6.98 kBit/s	12.71 kBit/s
4	6.98 kBit/s	15.20 kBit/s
5	21.63 kBit/s	38.96 kBit/s
<b>Rata-rata</b>	8.89 kBit/s	17.88 kBit/s

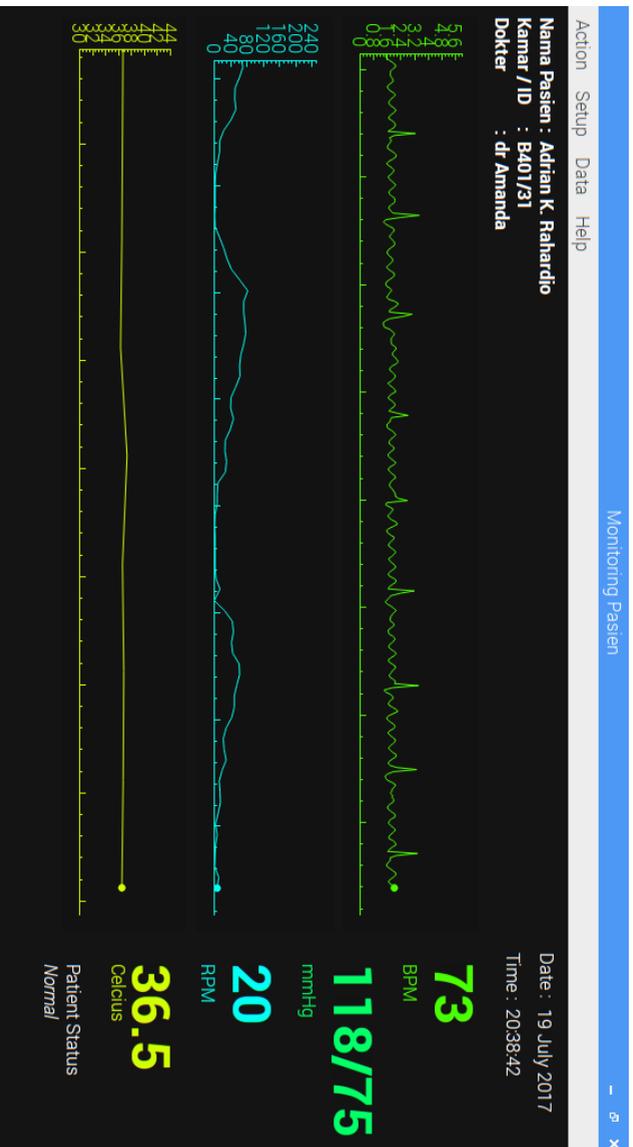
sebesar 38.96 *kBit/s*.

## 4.6 Pengujian *User Interface* Pemantauan

Pada pengujian ini dilakukan guna mengetahui seberapa baik data divisualisasikan. *User interface* yang dibuat berjalan dengan lancar sesuai dengan elemen yang dipaparkan pada bab 3.7. Visualisasi dilakukan pada kondisi pasien yang berbeda yaitu kondisi normal yang ditunjukkan oleh gambar 4.6 dan kondisi abnormal *vital sign* yang ditunjukkan oleh gambar 4.9. Pada kondisi normal tampilan *user interface* parameter *vital sign* berwarna kebiruan dan terdapat informasi status pasien normal. Sedangkan pada kondisi abnormal ditunjukkan dengan adanya parameter *vital sign* yang berwarna merah dan terdapat informasi bahwa status pasien dalam keadaan abnormal.

Pada tampilan utama terdapat menu yang diuji fungsionalitasnya. Menu yang utama yaitu menu *setup* identitas pasien yang ditunjukkan pada gambar 4.7 dan menu atur normal *vital sign* pasien yang ditunjukkan oleh gambar 4.8. Menu *setup* pasien berfungsi dengan baik sebagaimana yang telah didesain yaitu dapat mengatur atau mendaftarkan pasien yang akan dipantau. Sedangkan menu atur normal *vital sign* juga berjalan dengan baik yakni dapat digunakan untuk mengatur batas atas dan bawah normal *vital sign* pasien yang dipantau.

Selain menguji fungsionalitas *user interface* juga dilakukan pengujian terhadap seberapa baik *user interface* dipahami. Pengujian ini dilakukan dengan membuat survei yang ditujukan kepada umum.



Gambar 4.6: Visualisasi *vital sign* keadaan normal

Set up pasien

Identitas pasien

Nama pasien :

No. KTP :

Jenis kelamin:

Tempat, tanggal lahir :

Alamat :

Kamar :

ID puskesmas :

ID dokter :

No. ID : 31

Berhasil daftar pasien dan rawat

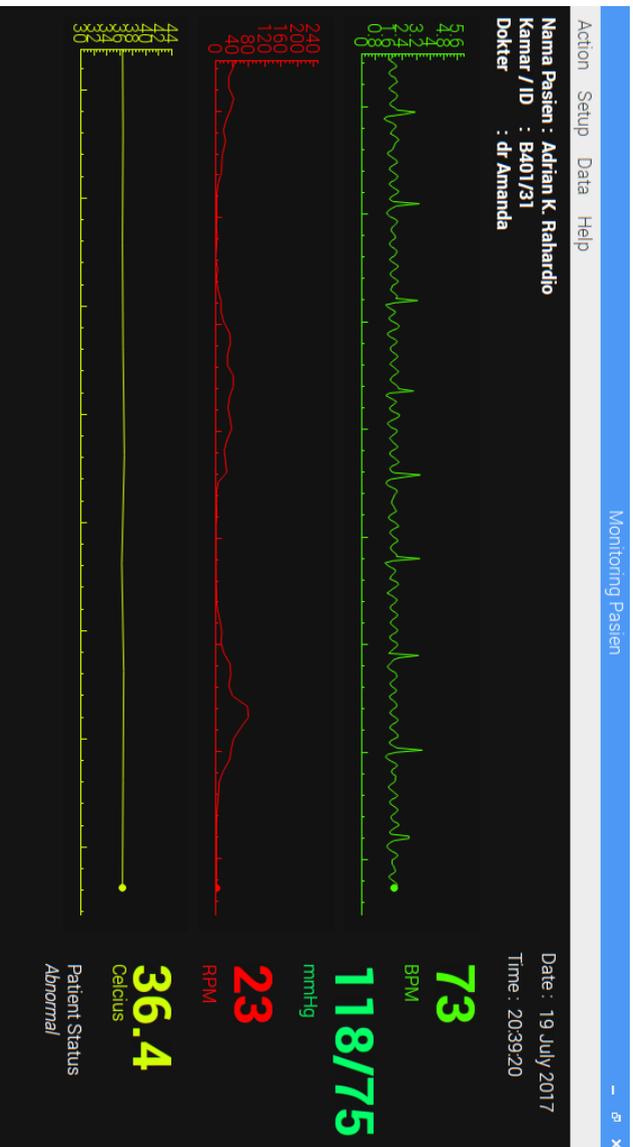
**Gambar 4.7:** Menu *setup* identitas pasien

Set up pasien

Set normal vital sign

	Min	Max
Heart rate :	<input type="text" value="60"/>	<input type="text" value="100"/>
Respiratory rate :	<input type="text" value="12"/>	<input type="text" value="20"/>
Temperature :	<input type="text" value="36"/>	<input type="text" value="38"/>
Sistol BP :	<input type="text" value="110"/>	<input type="text" value="140"/>
Diastol BP :	<input type="text" value="60"/>	<input type="text" value="90"/>

**Gambar 4.8:** Menu *set normal vital sign* pasien



Gambar 4.9: Visualisasi *vital sign* keadaan abnormal

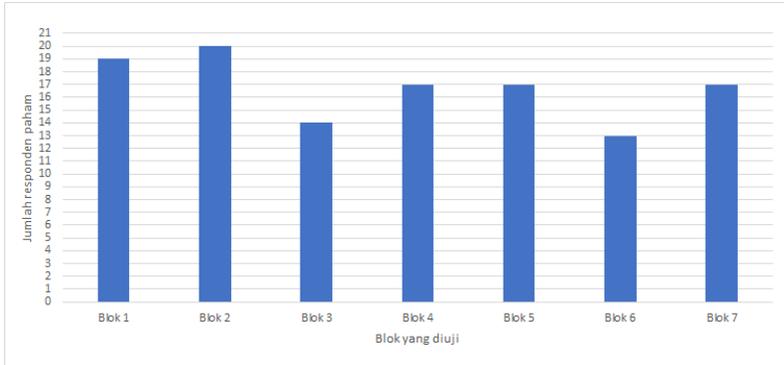
Pada survei dibuat tiga sub pertanyaan yaitu tentang kepehaman terhadap *user interface*, kejelasan *user interface*, dan pengetahuan terhadap kondisi normal dan abnormal *vital sign*. Pengujian dilakukan kepada 20 responden. Pada pertanyaan pertama yaitu tentang kepehaman terhadap *user interface* dibuat blok-blok pada elemen yang terdapat pada tampilan utama yang ditunjukkan oleh gambar 4.10. Elemen tersebut yaitu:

1. Blok 1 menunjukkan menu bar
2. Blok 2 menunjukkan identitas pasien
3. Blok 3 menunjukkan detak jantung
4. Blok 4 menunjukkan suhu tubuh
5. Blok 5 menunjukkan nafas per menit
6. Blok 6 menunjukkan tekanan darah
7. Blok 7 menunjukkan informasi status pasien



**Gambar 4.10:** Blok pada elemen *user interface* yang diujikan

Hasil dari survei pertanyaan tentang pengetahuan terhadap elemen *user interface* ditunjukkan oleh grafik pada gambar 4.11. Pada grafik tersebut menunjukkan bahwa dari 20 responden, terdapat 83% dari responden mengetahui dengan baik elemen-elemen yang terdapat pada tampilan utama. Pada pertanyaan kedua yakni apakah *user interface* mudah dipahami dan terbaca dengan baik. Hasil dari survei menunjukkan hal tersebut berbanding lurus dengan uji pada elemen yaitu sebesar 83% responden menyatakan *user in-*



**Gambar 4.11:** Hasil pengujian elemen tampilan utama

**Tabel 4.7:** Hasil pengujian kejelasan *user interface*

Pertanyaan	Jumlah responden yang menyatakan		
	Ya	Ragu-ragu	Tidak
Apakah <i>user interface</i> mudah dipahami?	18	2	0
Apakah <i>user interface</i> dapat terbaca dengan baik?	15	5	0

*terface* mudah dipahami dan terbaca dengan baik. Hasil tersebut ditunjukkan oleh tabel 4.7. Pertanyaan terakhir yaitu mengenai ke-pahaman terhadap perbedaan kondisi normal dan abnormal. Dari 20 responden, sebanyak 75% mengetahui dengan benar dan 25% tidak mengetahui perbedaannya.

# BAB 5

## PENUTUP

### 5.1 Kesimpulan

Pada tugas akhir ini telah dijelaskan pembuatan sistem pemantauan *vital sign*. Sistem yang dibuat terdiri dari akuisisi data empat *vital sign* yang divisualisasikan secara *real-time* lalu melakukan penyimpanan ke *cloud storage* dan membuat laporan *vital sign*. Sistem telah berhasil dikembangkan dan berdasarkan pengembangan sistem hingga pengujian terhadap sistem diperoleh kesimpulan sebagai berikut:

1. Persentase data terkirim ke SBC melalui komunikasi serial dari hasil akuisisi oleh sensor *node* yaitu sebesar 86.22%.
2. Validasi data dari sensor *node* yang dikirim ke SBC merupakan data yang valid berdasarkan dua cara pengujian *log data* dan *labeling data*.
3. Pada proses penyimpanan data ke *cloud storage* data berhasil terkirim secara baik yaitu jika dipersentase sebesar 100%, namun semakin banyak data maka, semakin lama pula proses penyimpanan.
4. Tingkat akurasi masing-masing sensor jika dipersentase yaitu pada sensor ECG sebesar 83.47%, sensor suhu sebesar 97.36%, dan pada sensor nafas memiliki tingkat akurasi sebesar 76.42%.
5. Data pada sensor tekanan darah mempunyai kekurangan tidak dapat mengakuisisi secara terus-menerus karena pengukuran tekanan darah dibutuhkan sekali pengukuran. Data terbaru harus dilakukan proses pengukuran pada selang waktu tertentu dan memerlukan proses *restarting* pada sensor *node*.
6. Pemakaian sumber daya jaringan pada proses penyimpanan data ke *cloud storage* cukup kecil. Nilai rata-rata *uplink* sebesar 572.2 *Bit/s* dan *downlink* sebesar 3.54 *kBit/s*.
7. Berdasarkan hasil *survey*, *user interface* bersifat informatif. Sebanyak 83% responden menyatakan jelas dan paham terhadap visualisasi data yang ditampilkan.

*Halaman ini sengaja dikosongkan*

## DAFTAR PUSTAKA

- [1] Kemenkes, “Keputusan menteri kesehatan republik indonesia nomor 128/menkes/sk/ii/2004.” <https://pelayanan.jakarta.go.id/download/regulasi/keputusan-menteri-kesehatan-nomor-128-menkes-sk-ii-2004-tentang-kebijakan-dasar-pusat-kesehatan-masyarakat.pdf>. Diakses pada: 2017-03-22. (Dikutip pada halaman 1, 5).
- [2] S. D. Indonesia, “Jumlah puskesmas rawat inap dan non rawat inap.” <https://data.go.id/dataset/puskesmas-perawatan-dan-non-perawatan/resource/1d391dfe-4728-47ff-b7b4-98c836fe6792>. Diakses pada: 2017-03-22. (Dikutip pada halaman 1).
- [3] M. F. Ismail, Pemantauan Kesehatan Multi-Pasien Secara Terpusat Menggunakan Parameter Parameter Vital Sign Berbasis Mikrokontroler dan Protokol TCP/IP. 2015. (Dikutip pada halaman 1).
- [4] J. Mohammed, “Internet of things: Remote patient monitoring using web services and cloud computing,” (Dikutip pada halaman 1, 18).
- [5] M. Elliott and A. Coventry, “Critical care: The eight vital signs of patient monitoring,” British Journal of Nursing, vol. 21, pp. 621–625, may 2012. (Dikutip pada halaman 6).
- [6] Clinicalgate, “The physical examination and its basis in physiology.” <https://clinicalgate.com/the-physical-examination-and-its-basis-in-physiology>. Diakses pada: 2017-03-20. (Dikutip pada halaman 7, 8, 10).
- [7] N. H. Lung and B. I. (NHLBI), “Description of high blood pressure.” <https://www.nhlbi.nih.gov/health/health-topics/topics/hbp>, September 2015. Diakses pada: 2017-03-22. (Dikutip pada halaman 9).

- [8] S. Mukherjee, K. Dolui, and S. K. Datta, “Patient health management system using e-health monitoring architecture,” in 2014 IEEE International Advance Computing Conference (IACC), pp. 400–405, Feb 2014. (Dikutip pada halaman 11).
- [9] “Raspberry pi pinout diagram circuit notes.” <http://www.jameco.com/Jameco/workshop/circuitnotes/raspberry-pi-circuit-note.html>. Diakses pada: 2017-07-20. (Dikutip pada halaman 13).
- [10] “Raspberry pi 3 is out now! specs, benchmarks & more.” <https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/>. Diakses pada: 2017-07-19. (Dikutip pada halaman 14).
- [11] “Introduction to arduino uno.” <http://www.elecrom.com/introduction-arduino-uno-uses-avr-atmega328/>. Diakses pada: 2017-07-20. (Dikutip pada halaman 14).
- [12] “Arduino uno rev3.” <https://store.arduino.cc/usa/arduino-uno-rev3>. Diakses pada: 2017-07-19. (Dikutip pada halaman 15).
- [13] M. Vidojevic, “Uart – serial communication.” <https://learn.mikroe.com/uart-serial-communication/>, September 2016. Diakses pada: 2017-03-22. (Dikutip pada halaman 15).
- [14] “e-health sensor platform v2.0 for arduino and raspberry pi [biometric / medical applications].” <http://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical>. Diakses pada: 2016-10-18. (Dikutip pada halaman 16, 18).
- [15] V. Beal, “What is cloud storage?.” [www.webopedia.com/TERM/C/cloud\\_storage.html](http://www.webopedia.com/TERM/C/cloud_storage.html). Diakses pada: 2016-09-30. (Dikutip pada halaman 18).
- [16] M. Borgmann, T. Hahn, M. Herfert, T. Kunz, M. Richter, U. Viebeg, and S. Vowe, On the Security of Cloud Storage

Services. Fraunhofer Institute for Secure Information Technology SIT, Germany: Mediendiensteleistungen des, Fraunhofer-Informationszentrum Raum und Bau IRB, Stuttgart, 2012. (Dikutip pada halaman 19).

- [17] “In introduction to http basics.” [https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP\\_Basics.html](https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html). Diakses pada: 2017-02-10. (Dikutip pada halaman 19).

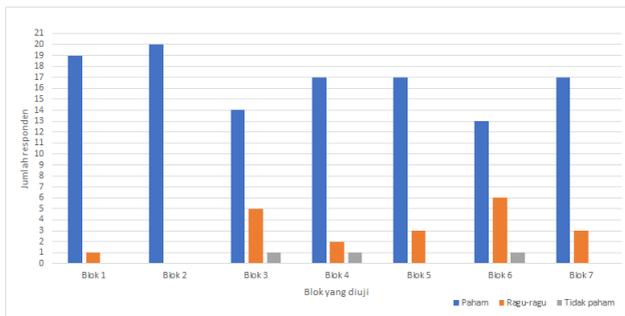
*Halaman ini sengaja dikosongkan*

## LAMPIRAN

Rekap kuisioner komposisi elemen pada *user interface*

Elemen user interface	Jumlah responden		
	Paham	Ragu-ragu	Tidak paham
Blok 1	19	1	0
Blok 2	20	0	0
Blok 3	14	5	1
Blok 4	17	2	1
Blok 5	17	3	0
Blok 6	13	6	1
Blok 7	17	3	0

Dalam bentuk grafik



*Halaman ini sengaja dikosongkan*

## BIOGRAFI PENULIS



Ahmad Aris Habibi, lahir pada 29 Desember 1994 di Blitar, Jawa Timur. Penulis mulai belajar formal di TK Kusuma Mulia, lulus pada tahun 2000 kemudian lulus dari SD Tales II tahun 2007 dan lanjut ke MTsN Kunir Blitar lulus pada tahun 2010 hingga akhirnya lulus pada jenjang SMA di SMAN 4 Kediri pada tahun 2013. Penulis melanjutkan pendidikan Strata satu di Departemen Teknik Komputer ITS Surabaya bidang studi Telematika. Saat di perkuliahan penulis aktif menjadi Asisten laboratorium komputasi multimedia (B401) hingga penulis menyelesaikan buku ini. Selama kuliah penulis juga aktif mengikuti ajang perlombaan teknologi seperti "Smart Digitizing Your City 2016". Penulis sangat tertarik dalam hal yang berhubungan dengan teknologi piranti cerdas, *embedded system*, dan *internet of things*, dan berencana mendalami cabang ilmu komputer lainnya.

*Halaman ini sengaja dikosongkan*