



REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Menteri Hukum dan Hak Asasi Manusia Republik Indonesia, berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta yaitu Undang-Undang tentang perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra (tidak melindungi hak kekayaan intelektual lainnya), dengan ini menerangkan bahwa hal-hal tersebut di bawah ini telah tercatat dalam Daftar Umum Ciptaan:

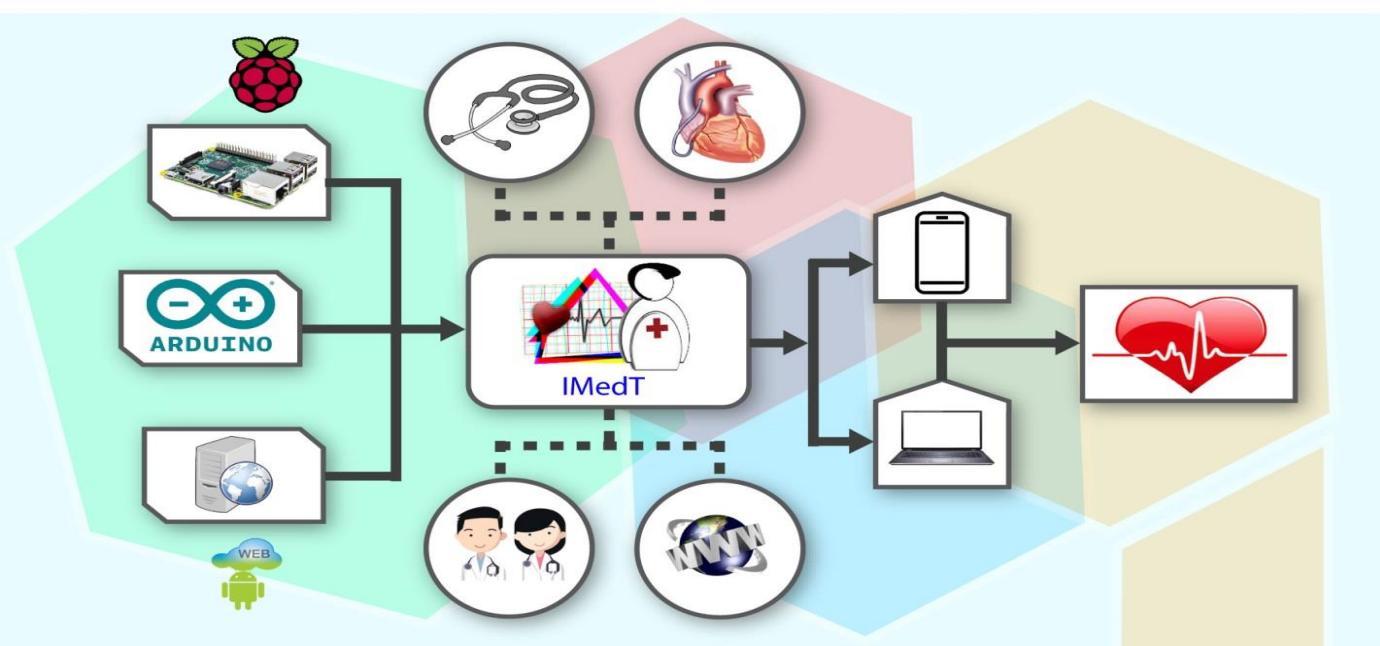
- I. Nomor dan tanggal permohonan : EC00201705410, 10 November 2017
- II. Pencipta
Nama : **Jusak ; Heri Pratikno ; Vergie Hadiana Putra**
Alamat : Taman Athena I-4 No.5 RT2 RW12 Gedangan Sidoarjo, Sidoarjo, JAWA TIMUR, 61254
- Kewarganegaraan : Indonesia
- III. Pemegang Hak Cipta
Nama : **Institut Bisnis dan Informatika Stikom Surabaya**
Alamat : Jalan Raya Kedung Baruk 98, Surabaya, Jawa Timur, Indonesia, Surabaya, JAWA TIMUR, 60298
- Kewarganegaraan : Indonesia
- IV. Jenis Ciptaan : Buku
- V. Judul Ciptaan : **Buku Laporan Hasil Penelitian Membangun Sistem Monitoring Suara Jantung Jarak Jauh Menggunakan Raspberry-PI 2 sebagai Web Server**
- VI. Tanggal dan tempat diumumkan : 12 Agustus 2016, di Surabaya
untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia
- VII. Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
- VIII. Nomor pencatatan : 04897

Pencatatan Ciptaan atau produk Hak Terkait dalam Daftar Umum Ciptaan bukan merupakan pengesahan atas isi, arti, maksud, atau bentuk dari Ciptaan atau produk Hak Terkait yang dicatat. Menteri tidak bertanggung jawab atas isi, arti, maksud, atau bentuk dari Ciptaan atau produk Hak Terkait yang terdaftar. (Pasal 72 dan Penjelasan Pasal 72 Undang-undang Nomor 28 Tahun 2014 Tentang Hak Cipta)

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
REPUBLIK INDONESIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
u.b.
DIREKTUR HAK CIPTA DAN DESAIN INDUSTRI

Dr. Dra. Erni Widhyastari, Apt., M.Si.
NIP. 196003181991032001

MEMBANGUN SISTEM *MONITORING SUARA JANTUNG JARAK JAUH MENGGUNAKAN RASPBERRY-PI 2 SEBAGAI WEB SERVER*



Pengarang :
Dr. Jusak
Heri Pratikno, M.T.
Vergie Hadiana Putra, S.Kom.

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

Heart & Mind Towards Excellence

ABSTRAK

Kesehatan dan perawatan medis merupakan isu serius yang mendesak dihadapi oleh penduduk dunia saat ini, masalah kesehatan semakin hari semakin berkembang dan kompleks seiring dengan perkembangan jaman yang menyebabkan perubahan gaya hidup, sikap dan perilaku yang terjadi pada semua bidang kehidupan manusia. Berdasarkan estimasi dari WHO, pada tahun 2020 penyakit jantung koroner (*coronary heart disease*) diprediksi akan menjadi salah satu pembunuhan utama bagi penduduk yang berada di kawasan Asia-Pasifik. Berkaitan dengan permasalahan tersebut diatas maka melalui buku ini, peneliti menuangkan sebuah karya tulis hasil dari implementasi dari sebuah ide atau gagasan yang telah dirancang serta dibangun dalam bentuk purwarupa (*prototype*) *Internet of Medical Things* (IMedT). Sistem berguna untuk memudahkan diagnosis kesehatan jantung melalui rekam jejak suara jantung yang bisa ditransmisikan secara jarak jauh melalui media internet, sehingga dokter dapat memantau, mendeteksi dan mengetahui kondisi pola-pola sinyal jantung pasiennya selama 24 jam *non-stop* dari manapun dan kapanpun.

Purwarupa alat IMedT yang dirancang dan dibangun berfungsi untuk memantau aktivitas jantung dalam bentuk sinyal *phonocardiograph* (PCG) dari sebuah sensor yang terhubung ke Arduino Uno, selanjutnya sinyal auskultasi suara jantung tersebut ditransmisikan melalui koneksi kabel serial USB ke mini komputer Raspberry Pi 2 yang difungsikan sebagai *web server* yang terkoneksi secara langsung ke internet. Karena itu sinyal PCG setiap orang dapat dipantau dengan cara melakukan ke akses ke *web server* tersebut. Kelebihan utama dari sistem IMedT ini antara lain: kemudahan untuk mobilitas, hemat energi dan portabilitas sistem. Sistem didesain dengan menggunakan modul piranti elektronik Arduino Uno yang kecil serta sistem komputer *web server* menggunakan Raspberry Pi yang ukurannya sebesar kartu ATM atau kartu kredit dengan catu daya 5 VDC. Disamping itu proses *monitoring* sinyal suara detak jantung bisa dilakukan secara daring yang bisa diakses dari perangkat *laptop*, *desktop* maupun *smartphone*.

Keyword: Raspberry Pi 2, Arduino Uno, Sinyal suara jantung, *Internet of Medical Things* (IMedT).

KATA PENGANTAR

Pertama-tama penulis panjatkan puji syukur kepada Tuhan Yang Maha Esa karena berkat, rahmat dan karunia-Nya penulis dapat menyelesaikan penulisan hasil penelitian dengan judul ‘‘Membangun Sistem *Monitoring* Suara Jantung Jarak Jauh Menggunakan Raspberry-Pi 2 sebagai *Web Server*’’ ini sebagai salah satu buku hak cipta di Institut Bisnis dan Informatika Stikom Surabaya.

Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada:

1. Pimpinan Institut Bisnis dan Informatika Stikom Surabaya yang telah banyak memberikan motivasi, keteladan dan bantuan dalam proses pengurusan buku hak cipta ini.
2. Bapak Tutut Wurjanto, S. Kom., M. Kom. selaku kepala bagian PPM (Penelitian dan Pengabdian Masyarakat) beserta jajarannya yang telah membantu serta mendukung dalam pengurusan pengajuan buku hak cipta ini.
3. Seluruh pihak yang tidak dapat penulis sebutkan satu persatu karena telah membantu penulis secara langsung maupun tidak langsung.

Sebagaimana pepatah mengatakan “tidak ada gading yang tidak retak” masih banyak hal perlu ditingkatkan dalam penyusunan dan penulisan buku hak cipta ini, oleh karena itu penulis mengharapkan saran dan kritik yang membangun dari semua pihak agar dapat menyempurnakan materi dari penulisan buku ini ke depan. Akhir kata penulis ucapan banyak terima kasih yang sebesar-besarnya kepada semua pihak, semoga tulisan pada buku ini dapat bermanfaat bagi para pembaca.

Surabaya, Oktober 2017

Penulis

DAFTAR ISI

BAB 1

Konsep IoT	1
1.1 Sinyal Suara Jantung (PCG)	3
1.2 Wavelet	5
1.3 <i>Internet od Things</i> (IoT)	5
1.4 <i>Internet of Medical Things</i> (IMedT)	6
1.5 Raspberry Pi2	7
1.6 Node.js	8
1.7 Web Server	9
1.8 Arduino	9
1.8.1 Arduino Uno	9
1.8.2 <i>Software</i> Arduino IDE	11
1.8.3 Bahasa Pemrograman Arduino	12
1.9 Parameter QoS	13

BAB 2

Desain Rancangan IMedT Untuk <i>Monitoring</i> Suara Jantung	15
2.1 Perancangan Perangkat Keras	17
2.1.1 <i>Heart Sound Sensor</i>	18
2.1.2 Perancangan Sensor Jantung	19
2.1.3 Perancangan Rangkaian Arduino ke Raspberry Pi 2	20
2.1.4 Perancangan Rangkaian Arduino ke PC Pembanding	21
2.2 Perancangan Perangkat Lunak	21
2.2.1 Algoritma Pembacaan Sinyal Jantung dan Pengiriman Data ..	21
2.2.2 Algoritma Penerimaan Data pada Raspberry Pi 2	25
2.2.3 Algoritma Pengolahan Data secara <i>Realtime</i> pada Raspberry Pi 2	27
2.2.4 Algoritma Pengolahan Data secara Offline pada Raspberry Pi 2	27
2.3 Metode Analisa	29
2.3.1 Peletakan Sensor Suara Detak Jantung pada Tubuh	29
2.3.2 Pengambilan Data Sinyal Suara Detak Jantung	29
2.3.3 Analisa Transmisi Sinyal Suara Detak Jantung	30

BAB 3

Membangun Purwarupa IMedT Untuk <i>Monitoring</i> Suara Jantung	32
3.1 Instalasi Raspberry Pi 2	33
3.2 <i>Update</i> dan <i>Upgrade</i> Raspbian	34

3.3 Pengujian Komunikasi Arduino Raspberry Pi 2 dan PC Pembanding	35
3.4 Pengujian Arduino	37
BAB 4	
Evaluasi Tampilan Grafik IMedT Monitoring Suara Jantung Secara Offline Dan Online	39
4.1 Pengujian Tampilan Penerimaan Data pada Raspberry Pi 2 secara <i>Live</i>	39
4.1.1 Prosedur Pengujian	39
4.1.2 Hasil Pengujian	40
4.2 Pengujian Tampilan Rekap Data pada Raspberry Pi 2 secara <i>Offline</i>	42
4.2.1 Prosedur Pengujian	43
4.2.2 Hasil Pengujian	43
BAB 5	
Evaluasi Kinerja Sistem IMedT Monitoring Suara Jantung	45
5.1 Prosedur Pengujian	45
5.2 Pengujian Sistem IMedT Monitoring secara Keseluruhan	46
5.2.1 Pengiriman Data Menggunakan Variasi <i>Baudrate</i>	47
5.2.2 Pengiriman Data Menggunakan <i>Baudrate</i> 115200 dan Variasi <i>Delay</i>	50
BAB 6	
Analisa Hasil Kinerja Sistem IMedT Monitoring Suara Jantung.....	53
6.1 Transmisi Sinyal dengan Baudrate 19200	53
6.2 Transmisi Sinyal dengan Baudrate 38400	54
6.3 Transmisi Sinyal dengan Baudrate 57600	55
6.4 Transmisi Sinyal dengan Baudrate 115200	55
BAB 7	
Kesimpulan Dan Saran Sistem IMedT Monitoring Suara Jantung	57
7.1 Kesimpulan	57
7.2 Saran	58
DAFTAR PUSTAKA	59
INDEKS	62

DAFTAR GAMBAR

Gambar 1.1 Bunyi jantung normal	4
Gambar 1.2 Anatomi jantung	4
Gambar 1.3 Bentuk sebuah <i>Wave</i> dan <i>Wavelet</i>	5
Gambar 1.4 Raspberry Pi 2	8
Gambar 1.5 Board Arduino Uno	10
Gambar 1.6 Tampilan <i>software</i> Arduino IDE	12
Gambar 2.1 Diagram blok sistem <i>monitoring</i> IMedT	16
Gambar 2.2 <i>Flowchart</i> sistem <i>monitoring</i> IMedT	16
Gambar 2.3 Perancangan perangkat keras sistem <i>monitoring</i> IMedT	17
Gambar 2.4 <i>Heart sound</i> sensor dari <i>Seeed Studio</i>	18
Gambar 2.5 Hubungan <i>heart sound</i> sensor dan Arduino	19
Gambar 2.6 Hubungan Arduino dengan Raspberry Pi 2	20
Gambar 2.7 Program penerimaan data dari Arduino ke Raspberry Pi 2	20
Gambar 2.8 Hubungan Arduino Uno dengan PC pembanding	21
Gambar 2.9 <i>Flowchart</i> pembacaan pada <i>heart sound</i> sensor	22
Gambar 2.10 <i>Flowchart</i> perbandingan bentuk sinyal auskultasi jantung	22
Gambar 2.11 Tampilan program pada <i>software</i> Arduino IDE	24
Gambar 2.12 <i>Flowchart</i> penerimaan data pada Raspberry Pi 2	25
Gambar 2.13 Program penerimaan data serial di NODE-RED	25
Gambar 2.14 Konfigurasi penerimaan data ke Raspberry Pi 2	26
Gambar 2.15 Lanjutan konfigurasi serial <i>port</i> di NODE-RED	26
Gambar 2.16 Pilihan menu <i>file</i> program NODE-RED	26
Gambar 2.17 Directory penyimpanan <i>file</i> serial.txt	27
Gambar 2.18 <i>Flowchart</i> pengolahan data secara <i>realtime</i> di Raspberry Pi 2	28
Gambar 2.19 <i>Flowchart</i> pengolahan data secara <i>offline</i> di Raspberry Pi 2 ..	28
Gambar 2.20 Letak posisi penempatan sensor suara detak jantung	29
Gambar 2.21 Hasil sinyal suara detak jantung	30

Gambar 2.22 Hasil komparasi kesesuaian data yang dikirim dan diterima ..	30
Gambar 2.23 Hasil komparasi ketidaksesuaian data yang diterima	31
Gambar 3.1 Tampilan koneksi peralatan IMedT	32
Gambar 3.2 Tampilan <i>software</i> untuk mem-format SD Card	33
Gambar 3.3 Tampilan <i>software</i> untuk ekstraksi sistem operasi Raspbian ...	34
Gambar 3.4 Tampilan antarmuka menu Terminal	35
Gambar 3.5 Tampilan Arduino terkoneksi pada terminal di Raspberry Pi 2	35
Gambar 3.6 Tampilan Arduino terkoneksi pada PC pembanding	36
Gambar 3.7 Komunikasi <i>serial</i> pada Arduino dan Raspberry Pi 2	36
Gambar 3.8 <i>Upload</i> program berhasil	38
Gambar 3.9 Program berhasil berjalan	38
Gambar 4.1 Tampilan menu utama <i>website</i> IMedT	40
Gambar 4.2 Tampilan pengaksesan data secara <i>live</i>	41
Gambar 4.3 Penyimpanan data <i>live</i>	41
Gambar 4.4 Tampilan konfirmasi <i>entry</i> rekap data pasien	44
Gambar 4.5 Hasil tampilan grafik sinyal jantung pasien terpilih	44
Gambar 5.1 Percobaan 1 pada Raspberry Pi 2 dengan <i>baudrate</i> 19200	47
Gambar 5.2 Percobaan 1 pada PC pembanding dengan <i>baudrate</i> 19200	47
Gambar 5.3 Percobaan 2 pada Raspberry Pi 2 dengan <i>baudrate</i> 38400	48
Gambar 5.4 Percobaan 2 pada PC pembanding dengan <i>baudrate</i> 38400	48
Gambar 5.5 Percobaan 3 pada Raspberry Pi 2 dengan <i>baudrate</i> 57600	49
Gambar 5.6 Percobaan 3 pada PC pembanding dengan <i>baudrate</i> 57600	49
Gambar 5.7 Percobaan 4 pada Raspberry Pi 2 dengan <i>baudrate</i> 115200	49
Gambar 5.8 Percobaan 4 pada PC pembanding dengan <i>baudrate</i> 115200 ...	50
Gambar 5.9 Raspberry Pi 2 dengan <i>baudrate</i> 115200 dan <i>delay</i> 1 milidetik	50
Gambar 5.10 PC pembanding dengan <i>baudrate</i> 115200 dan <i>delay</i> 1 milidetik	51
Gambar 5.11 Raspberry Pi 2 dengan <i>baudrate</i> 115200 dan <i>delay</i> 2 milidetik	51

Gambar 5.12 PC pembanding dengan <i>baudrate</i> 115200 dan <i>delay</i> 2 milidetik	51
Gambar 5.13 Raspberry Pi 2 dengan <i>baudrate</i> 115200 dan <i>delay</i> 3 milidetik	52
Gambar 5.14 PC pembanding dengan <i>baudrate</i> 115200 dan <i>delay</i> 3 milidetik	52

DAFTAR TABEL

Tabel 1.1 Spesifikasi Arduino Uno	11
Tabel 1.2 <i>Packet Loss</i>	14
Tabel 3.1 Spesifikasi <i>Heart Sound Sensor</i>	18
Tabel 5.1 Hasil percobaan 1 dengan <i>baudrate</i> 19200	48
Tabel 5.2 Tabel hasil percobaan 2 dengan <i>baudrate</i> 38400	48
Tabel 5.3 Tabel hasil percobaan 3 dengan <i>baudrate</i> 57600	49
Tabel 5.4 Tabel hasil percobaan 4 dengan <i>baudrate</i> 115200	50
Tabel 5.5 Tabel hasil percobaan 5 dengan <i>baudrate</i> 115200 dan <i>delay</i> 1 ms	51
Tabel 5.6 Tabel hasil percobaan 6 dengan <i>baudrate</i> 115200 dan <i>delay</i> 2 ms	52
Tabel 5.7 Tabel hasil percobaan 7 dengan <i>baudrate</i> 115200 dan <i>delay</i> 3 ms	52
Tabel 6.1 Perhitungan <i>throughput</i> dan utilisasi <i>bandwidth</i> dengan <i>baudrate</i> 115200 pada Raspberry Pi 2	54
Tabel 6.2 Tabel hasil perhitungan rata-rata <i>throughput</i> dan utilisasi <i>bandwidth</i> pada pengujian variasi <i>baudrate</i>	54
Tabel 6.3 Tabel hasil perhitungan rata-rata <i>throughput</i> dan utilisasi <i>bandwidth</i> pada pengujian variasi <i>delay</i>	55

BAB 1

Konsep IoT

Isi bab 1 membahas konsep dasar dari kerja jantung, sinyal suara jantung, *Internet of Things* (IoT) serta *Internet of Medical Things* (IMedT).

Kesehatan merupakan isu serius yang tengah di hadapi oleh masyarakat dunia saat ini sebagai akibat dari perubahan gaya hidup dan pola hidup manusia. Berdasarkan data terbaru yang dikeluarkan oleh *World Health Organization* (WHO) tahun 2014, angka kematian yang disebabkan oleh penyakit kardiovaskular pada tahun 2012 telah mencapai 17,5 juta orang atau 46% dari jumlah total kematian yang disebabkan oleh penyakit menular (WHO, 2014).

Selanjutnya WHO memprediksi bahwa pada tahun 2020 penyakit jantung coroner akan menjadi penyakit pembunuh utama di Indonesia serta kawasan negara-negara di Asia Pasifik (WHO, 2008). Jantung adalah organ tubuh manusia yang sangat penting dan berguna untuk memompa darah yang beredar ditubuh manusia. Tanpa jantung, darah di tubuh manusia tidak akan dapat mengalir, pola hidup yang tidak sehat seperti makanan yang berkolesterol tinggi atau berlemak tinggi serta kurangnya olahraga dapat memicu penyakit jantung (Daso, 2015).

Serangan penyakit jantung dapat dialami oleh siapa saja, kapanpun, dimanapun secara mendadak karena sulit terdeteksi secara dini, disarankan ntuk mencegah terjadinya serangan penyakit jantung sebaiknya melakukan pemeriksaan sejak dini dan secara berkala oleh dokter ahli atau dokter spesialis penyakit jantung. Proses pemeriksaan yang dilakukan oleh dokter ahli disebut dengan istilah auskultasi (Puspasari, 2013), auskultasi adalah pemeriksaan kinerja organ

tubuh seperti jantung dengan cara mendengarkan suara yang diakibatkan oleh vibrasi yang berasal dari proses kerja jantung.

Berdasarkan fakta tersebut maka diperlukan sebuah solusi teknologi yang tepat guna membantu melakukan deteksi dini terhadap penyakit kardiovaskular untuk mengurangi angka kematian yang disebabkan oleh serangan penyakit tersebut. Pada penelitian sebelumnya proses transmisi secara nirkabel sinyal auskultasi suara jantung dengan menggunakan teknologi *wireless* berbasis jaringan *Zigbee* (Jusak, 2015).

Maksud dan tujuan dari perancangan pada penelitian sebelumnya dibuat agar dokter dapat mengetahui kondisi jantung dari pasien tanpa melakukan pemeriksaan fisik secara langsung dari 2 *node* secara bersamaan dengan tepat dan data tidak tertukar, tetapi pada penelitian ini mempunyai kelemahan yang mendasar, yaitu: masih menggunakan komputer *desktop* (PC) yang berukuran besar sebagai pemrosesan komputasi datanya sehingga mengakibatkan tidak fleksibel, tidak *portable* serta tidak bisa digunakan secara *mobile* karena hanya bisa mengecek kondisi pasien yang berada sekitar daerah yang terjangkau saja serta data yang dikirimkan masih terdapat kemungkinan untuk hilang (Oktarina, 2015).

Masalah utama yang terjadi saat ini adalah pemeriksaan pasien jantung harus dilakukan kontak fisik secara langsung dengan dokter, terlebih bagi pasien yang berada pada daerah terpencil atau jauh dari jangkauan dokternya akan kesulitan untuk dilakukan pemeriksaan kondisi jantungnya sewaktu-waktu.

Sebagai solusi atas permasalahan diatas maka dibuatlah sebuah konsep dalam bentuk purwarupa atau *prototype* rancang bangun sistem monitoring suara jantung berbasis *web* dengan menggunakan Raspberry Pi 2 sebagai *server*. Hasil dari perancangan sistem *monitoring* secara jarak jauh ini dibuat agar dokter dapat mengetahui kondisi jantung dari pasien kapanpun, dimanapun selama 24 jam *non-stop* tanpa perlu bertemu atau kontak fisik secara langsung melalui *web browser* yang bisa diakses dari komputer maupun *smartphone*.

Kelebihan utama dari penelitian ini adalah: (i) Pengolah dan pemroses sinyal suara jantung menggunakan Arduino Uno sedangkan *Database web server*nya dibangun diatas mini-komputer Raspberry Pi 2 dengan ukurannya hanya sebesar kartu kredit atau kartu ATM yang bisa diletakkan diatas telapak tangan sehingga memiliki poratabilitas dan mobilitas yang tinggi. (ii) Modul Arduino Uno maupun mini-komputer *server* tersebut sangat hemat energi karena hanya perlu catu daya sebesar 5 VDC dengan kapasitas arus minimal 500 mA sehingga bisa menggunakan adaptornya *smartphone*. (iii) Mempunyai jarak jangkauan yang jauh karena menggunakan media transmisi via internet atau daring, (iv) sistem monitoring suara jantung ini sangat fleksibel, *portable* dan mempunyai mobilitas yang tinggi karena bisa diakses secara langsung melalui komputer maupun *handphone*.

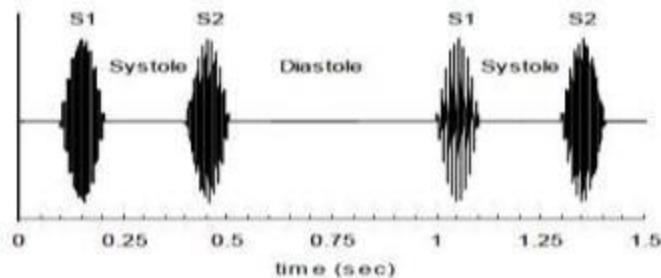
1.1 Sinyal Suara Jantung (PCG)

Jantung adalah organ tubuh yang berfungsi untuk memompa darah yang terdiri dari bagian atas disebut serambi (*atrium*) dan bagian bawah yang disebut dengan bilik (*ventricle*). Otot-otot jantung memompa darah dari satu ruangan ke ruangan lainnya, setiap kali terjadi proses pemompaan, katup jantung membuka sehingga darah dapat mengalir ke ruangan yang dituju. (Setiaji, 2011).

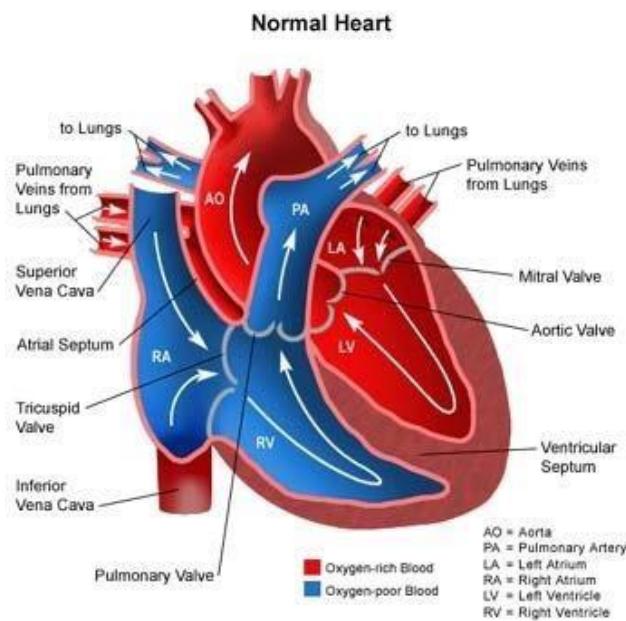
Pada detak jantung dihasilkan dua suara berbeda yang dapat didengarkan dengan stetoskop, seringkali dinyatakan dengan *lub-dub*. Suara *lub* disebabkan oleh penutupan katup *tricuspid* dan *mitral* (*atrioventrikular*) yang memungkinkan aliran darah dari *atrium* (serambi jantung) ke *ventricle* (bilik jantung) serta mencegah aliran balik biasanya disebut dengan suara jantung pertama (S1) yang terjadi pada awal *systole* (periode jantung berkontraksi).

Suara *dub* disebut suara jantung kedua (S2) yang terjadi pada akhir *systole* atau awal *diastole* dan disebabkan oleh penutupan katup *semilunar* (*aortic* dan

pulmonary) yang membebaskan darah ke sistem sirkulasi paru-paru dan seluruh tubuh. Sinyal suara jantung merupakan sinyal gelombang suara yang lemah, dimana sinyal ini berada di-range antara 10 Hz hingga 250 Hertz (Sangasoongsong, 2012).



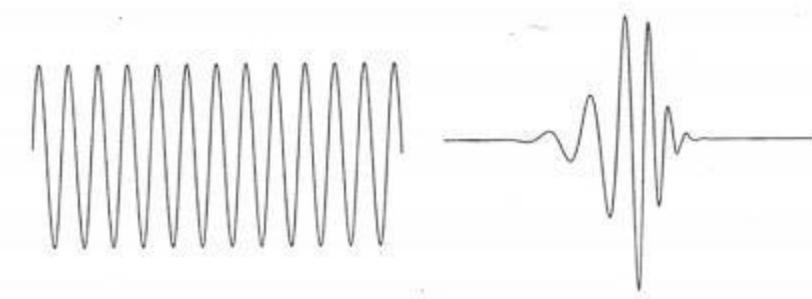
Gambar 1.1 Bunyi jantung normal (Setiaji, 2011).



Gambar 1.2 Anatomi jantung (Anonim, 2015).

1.2 Wavelet

Wavelet adalah sebuah gelombang kecil, yang energinya terkonsentrasi dalam waktu untuk menyediakan alat bantu analisis *non-stationer* atau perubahan waktu. Karakteristik *wave* bergerak masih tetap dimiliki, namun juga dapat mensimulasikan analisis waktu-frekuensi dengan dasar matematika yang fleksibel. Hal ini diilustrasikan dalam Gambar 1.3 dimana *wave* (kurva sinus) bergerak dengan amplitudo sama pada $-\infty \leq t \leq \infty$ sehingga memiliki energi yang tak berhingga dengan *Wavelet* yang memiliki energi berhingga terkonsentrasi pada suatu titik (Burrus, Gopinath, Guo, 1998).



Gambar 1.3 Bentuk sebuah *Wave* dan *Wavelet*.
(Burrus, Gopinath, Guo, 1998)

1.3 Internet of Things (IoT)

Cisco merilis pada salah satu *white papers*-nya dalam sebuah laporan tahunan dari Virtual Network Index (VNI) memberikan sebuah imajinasi bagaimana perkembangan internet ke depannya. Menurut *white-papers* tersebut ada bukti kuantitatif bahwa proliferasi lalu lintas IP global dalam pertukaran data bisa mencapai kapasitas sampai dalam satuan *Zettabyte* (ZB) pada tahun 2020. Jumlah data yang besar ini terutama didorong oleh sejumlah perangkat yang terkoneksi ke jaringan yang berbasis *IP Address*, diantaranya: *smartphone*, *tablet*, sensor dan aplikasi dari piranti yang berbasis Mesin ke Mesin (M2M) yang diperkirakan tiga kali lebih banyak dari populasi global.

Situasi dan kondisi saat ini sampai dengan tahun 2020, setiap objek fisik yang dilihat saat ini sebagai contoh: mobil, mesin, alat komunikasi, jam tangan, peralatan kantor maupun peralatan rumah tangga sudah terkoneksi ke internet, era ini biasa disebut dengan istilah *Internet of Things* (IoT). Implementasi dunia IoT menawarkan banyak solusi tepat guna untuk menghubungkan perangkat-perangkat yang berbasis koneksi secara nirkabel (*wireless*) yang dapat saling bertukar data dengan berbagai macam aplikasi dan layanan yang dapat dimanfaatkan beberapa contoh diantaranya: memantau *traffic* lalu lintas, *smart cities*, penerapan dalam bidang pertanian, perkebunan serta kebakaran hutan, manajemen limbah, logistik, transpotasi, medan perang serta perawatan kesehatan.

Pada perkembangan IoT komunikasi dari akses perangkat satu ke perangkat yang lainnya dapat dilakukan dengan menggunakan teknologi yang berbasis radio, seperti: komunikasi seluler, *Wireless Local Area Network* (WLAN), *Bluetooth*, *Radio Frequency Identification* (RFID), *Zigbee* dan *Near Field Communication* (NFC). Untuk lebih mengintensifkan ide koneksi *ubiquitous* dimana-mana maka diperlukan menyamakan berbagai standar yang berbeda termasuk di dalamnya masalah *data rate* dan *latency*, pada saat ini sedang berlangsung penelitian secara intensif yang dilakukan oleh para akademisi, pihak swasta dan konsorsium di Eropa yang menjalin kemitraan dalam pengembangan jaringan *mobile 5G* (5G-PPP) untuk mengevaluasi berbagai macam standarisasi 5G dengan harapan dapat mengakomodasi pertumbuhan *data rate* yang berkembang secara eksponensial dari milyaran koneksi perangkat di masa depan ke internet.

1.4 *Internet of Medical Things (IMedT)*

Implementasi pemanfaatan aplikasi dunia IoT yang paling menarik dan menjadi perhatian utama adalah penerapan dalam bidang kesehatan dan perawatan medis (Jusak, 2016). Adapun penerapan IoT dalam dunia medis dibangun melalui pengembangan berbagai macam jenis sensor yang dihubungkan dengan berbagai peralatan medis yang canggih atau cerdas maka

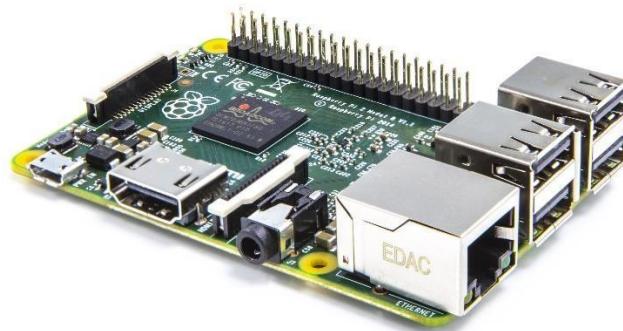
dapat dimanfaatkan salah satu diantaranya adalah untuk menyediakan layanan pemantauan tele-askultasi, diagnostik jarak jauh serta pemantauan perawatan medis bagi para lansia.

Penerapan teknologi IoT khusus dalam layanan bidang perawatan kesehatan (*health care services*) dapat disebut dengan istilah IMedT (*Internet of Medical Things*). Dengan adanya penerapan IMedT dapat mengurangi biaya konsultasi, biaya transportasi, mengatasi permasalahan jarak terlebih pada daerah yang terpencil, tidak perlu kontak fisik atau tatap muka secara langsung antara dokter dan pasien, aktivitas pemantauan dan monitoring bisa dilakukan kapan saja dan dimana saja baik melalui komputer, *laptop* maupun *handphone*.

Dokter dapat memantau kondisi kesehatan pasien pada masa proses berobat jalan sehingga segera mengetahui perkembangan efeknya secara langsung hasil dari pengobatan pada pasien. Dilihat dari sudut pandang informasi manajemen kesehatan, IMedT juga dapat membantu pengelolaan mata rantai stok logistik obat-obatan bagi rumah sakit maupun si pasien.

1.5 Raspberry Pi 2

Raspberry Pi yang biasa disingkat dengan Raspi adalah sebuah mini komputer yang ukurannya sebesar kartu kredit atau kartu ATM sehingga bisa diletakkan diatas telapak tangan, mini komputer Raspberry Pi merupakan hasil dari sebuah ide seorang mahasiswa dari Universitas Cambridge Inggris yang bernama Eben Upton pada tahun 2009. Adapun ide dibalik pembuatan Raspberry Pi diawali adanya keinginan untuk membuat komputer yang berukuran kecil, murah serta mencetak pemrogram generasi baru, saat ini Raspberry Pi dikembangkan oleh yayasan nirlaba Raspberry Pi Foundation



Gambar 1.4 Raspberry Pi 2

Raspberry Pi 2 berbasis mikrokontroller ATmega 644 dengan berdesain SoC (*System on Chip*), memiliki sistem *Broadcom BCM2836*, lebih cepat hingga 6x lipat dengan menggunakan *Quad-Core* ARM Cortex-A7 900MHz dari generasi sebelumnya yang menggunakan ARM1176JZF-S 700 MHz *single processor*, VideoCore IV GPU, kapasitas RAM sampai dengan 1024 MB, menggunakan SD Card (*microSDHC*) sebagai media untuk *booting* dan menyimpan data.

Yayasan Raspberry Pi Foundation membagikan *Debian* dan *Arch Linux ARM* yang bisa di-*download* secara gratis dengan dukungan *Python* sebagai bahasa pemrograman utama, *C* dan *Perl*. Pada tanggal 17 Desember 2012, yayasan Raspberry Pi Foundation bekerjasama dengan *IndieCity* dan *Velocix* membuka "*Store Pi*" sebagai "*one-stop shop*" untuk semua kebutuhan perangkat lunak Raspberry Pi.

1.6 Node.js

Javascript merupakan bahasa pemrograman yang lengkap akan tetapi selama ini hanya di pakai sebagai bahasa untuk pengembangan aplikasi *web* yang berjalan pada sisi *client* atau *browser* saja. Setelah ditemukannya Node.js oleh Ryan Dhal pada tahun 2009, *Javascript* bisa digunakan sebagai bahasa pemrograman di sisi *server* sekelas dengan PHP, ASP, C# dan Ruby. Sehingga Node.js menyediakan *platform* untuk membuat aplikasi *Javascript* dapat

dijalankan di sisi *server*, untuk mengeksekusi *Javascript* sebagai bahasa *server* diperlukan *engine* yang cepat dan mempunyai performa yang handal. *Engine Javascript* dari Google bernama V8 yang dipakai oleh Node.js merupakan *engine* yang sama digunakan pada *browser* Google Chrome. (idjs.github.io, 2016).

1.7 Web Server

Fungsi utama *Web server* adalah untuk mentransfer berkas permintaan dari pengguna melalui protokol komunikasi yang telah ditentukan sedemikian rupa. Halaman *web* yang diminta terdiri dari berkas teks, video, gambar, *file* dan sebagainya, pemanfaatan *web server* berfungsi untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman *web* termasuk di dalamnya dapat berupa teks, video maupun gambar.

Salah satu contoh dari *web server* adalah Apache, Apache merupakan *web server* yang paling banyak dipergunakan di Internet, program ini pertama kali didesain untuk sistem operasi di lingkungan UNIX. Apache mempunyai program pendukung yang cukup banyak sehingga hal ini memberikan layanan yang cukup lengkap bagi penggunanya (idcloudhost.com, 2015).

1.8 Arduino

Arduino adalah *prototipe platform* elektronik *open source* yang terdiri dari mikrokontroler, bahasa pemrograman dan IDE. Arduino adalah alat yang biasa digunakan untuk membuat aplikasi interaktif yang dirancang untuk mempermudah pembuatan proyek bagi pemula, tapi cukup fleksibel juga bagi para pengguna tingkat lanjut untuk mengembangkan proyek-proyek yang kompleks (Banzi, 2009).

1.8.1 Arduino Uno

Arduino Uno merupakan *board* mikrokontroller *open source* yang menggunakan ATmega328 sebagai mikrokontrolernya. Arduino Uno memiliki 14 *pin input / output*, 6 *pin* dapat digunakan sebagai *output* PWM, 6 analog

input, crystal oscillator 6 Mhz, koneksi USB, jack power, ICSP header, dan tombol reset.



Gambar 1.5 Board Arduino Uno

Secara umum Arduino Uno terdiri dari dua bagian, yaitu:

1. *Hardware*: papan *input/output* (I/O) mempunyai 2 sifat *pin* yaitu *pin* digital dan *pin* analog. Digital berarti sinyal yang dikirimkan atau diterima bernilai 1 atau 0, *on* atau *off*, *HIGH* atau *LOW*, ada atau tidak ada sinyal. Berbeda dengan sinyal analog yang datanya dalam bentuk gelombang yang nilainya bersifat kontinyu.
2. *Software*: perangkat lunak Arduino meliputi IDE (*Integrated Development Environment*) merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan dalam penulisan sintak pemrograman, driver untuk koneksi dengan komputer, contoh program dan *library*. Arduino IDE dibuat dari bahasa pemrograman Java yang juga dilengkapi dengan *library* C/C++ yang disebut juga dengan *Wiring* untuk membuat operasi *input* dan *output* menjadi lebih mudah.

Tabel 1.1 Spesifikasi Arduino Uno

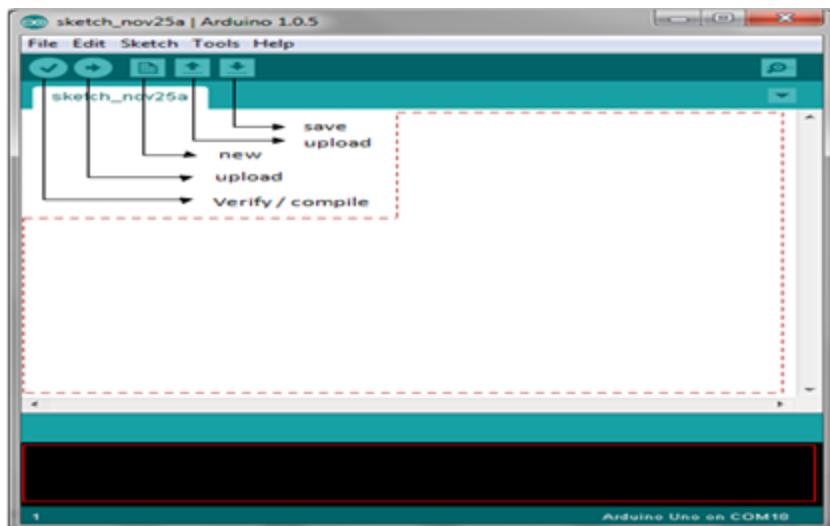
Mikrokontroler	ATMega 328P
Tegangan Operasi	5V
<i>Input</i> tegangan(rekomendasi)	7 – 12V
<i>Input</i> tegangan (Maksimal)	6 – 20V
Digital I/O Pin	14 (6 pin PWM)
Pin <i>input</i> Analog	6
DC current per I/O Pin	20mA
Pin DC Current untuk 3.3V	50mA
Memori <i>flash</i>	32Kb, 0.5Kb digunakan untuk <i>bootloader</i>
SRAM	2Kb
EEPROM	1Kb
<i>Clock</i> speed	16 Mhz

1.8.2 Software Arduino IDE

Sketch adalah program yang ditulis dengan menggunakan Arduino *software* (IDE), *sketch* ditulis dalam suatu editor teks jika disimpan menjadi sebuah *file* dengan ekstensi *.ino*. Software Arduino IDE memiliki 6 tombol utama dengan penjelasan sebagai berikut:

1. *Verify (Compile)*, berfungsi untuk melakukan pengecekan kode yang dibuat apakah sudah sesuai dengan sintak pemrograman yang benar.
2. *Upload*, berfungsi untuk melakukan kompilasi program atau kode yang dibuat kemudian dimuat ke *board* Arduino.
3. *New*, berfungsi untuk membuat *sketch* baru.
4. *Open*, berfungsi untuk membuka *sketch* yang pernah dibuat untuk diedit atau di-*upload* kembali *software* Arduino IDE.
5. *Save*, berfungsi untuk menyimpan *sketch* yang telah dibuat menjadi sebuah *file* dengan ekstensi *.ino*.
6. *Serial Monitor*, berfungsi untuk membuka tampilan jendela (*window*) sehingga data apa saja yang dikirim atau dipertukarkan antara Arduino

dengan *sketch* pada *port* serialnya, sangat berguna untuk proses *debugging* atau pada saat tidak ada LCD yang terpasang di Arduino.



Gambar 1.6 Tampilan *software* Arduino IDE

1.8.3 Bahasa Pemrograman Arduino

Bahasa pemrograman yang digunakan pada Arduino ini berdasarkan pada bahasa C/C++ terbagi menjadi tiga bagian utama yaitu: *Structure*, *Values* (berisi *variable* dan konstanta) dan *Function*. Penjelasannya sebagai berikut :

1. *Structure*

Struktur kode pada Arduino berisi fungsi **setup()** dan **loop()**:

- **Setup()**

Fungsi ini dipanggil pertama kali ketika menjalankan *sketch* pada saat *board* dinyalakan atau di *reset* sebagai tempat untuk inisialisasi *variable*, *pin mode*, penggunaan *library* dan lainnya.

- **Loop()**

Sesuai dengan namanya fungsi ini digunakan untuk melakukan perulangan sampai pada suatu kondisi bersyarat yang sudah ditentukan terpenuhi sehingga memungkinkan program untuk mengubah atau menanggapi dalam pengontrolan *board* Arduino.

2. Values

Berisi variable atau konstanta sesuai dengan type data yang didukung oleh Arduino.

3. Function

Segmentasi kode ke fungsi yang memungkinkan programer membuat potongan-potongan modular kode dalam melakukan tugas yang terdefinisi untuk kembali ke kode asal kode darimana fungsi tersebut dipanggil.

1.9 Parameter QoS

Quality of Service (QoS) merupakan terminologi yang didefinisikan sebagai kemampuan suatu jaringan untuk menyediakan tingkat jaminan layanan yang berbeda-beda. QoS didesain untuk membantu *end user* menjadi lebih produktif dengan memastikan bahwa *end user* mendapatkan performansi yang handal dari aplikasi-aplikasi yang berbasis jaringan. Parameter-parameter dalam QoS, diantaranya:

- *Throughput*, yaitu kecepatan (*rate*) transfer data efektif, yang diukur dalam bps. *Throughput* merupakan jumlah total kedatangan paket yang sukses diamati pada *destination* selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut.

$$\text{Throughput} = \frac{\text{Total frame yang diterima selama pengambilan}}{\text{Lama waktu pengambilan}}$$

- *Packet Loss*, merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang karena *collision* dan *congestion* pada jaringan sehingga berpengaruh pada semua aplikasi karena *retransmisi* akan mengurangi efisiensi jaringan secara keseluruhan meskipun jumlah *bandwidth* cukup tersedia untuk aplikasi-aplikasi tersebut. Umumnya perangkat jaringan memiliki *buffer* untuk menampung data yang diterima.

Tabel 1.2 *Packet Loss*

KATEGORI DEGREDASI	PACKET LOSS
Sangat bagus	0
Bagus	3 %
Sedang	15 %
Jelek	25 %

- *Delay/latency*, adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama.

$$\text{Delay rata - rata} = \frac{\text{Total delay transmisi}}{\text{Total paket yang diterima}}$$

- *Jitter*, adalah variasi kedatangan paket yang disebabkan oleh perubahan atau perbedaan panjang antrian, waktu pengolahan data dan waktu penghimpunan ulang paket-paket di akhir perjalanan.

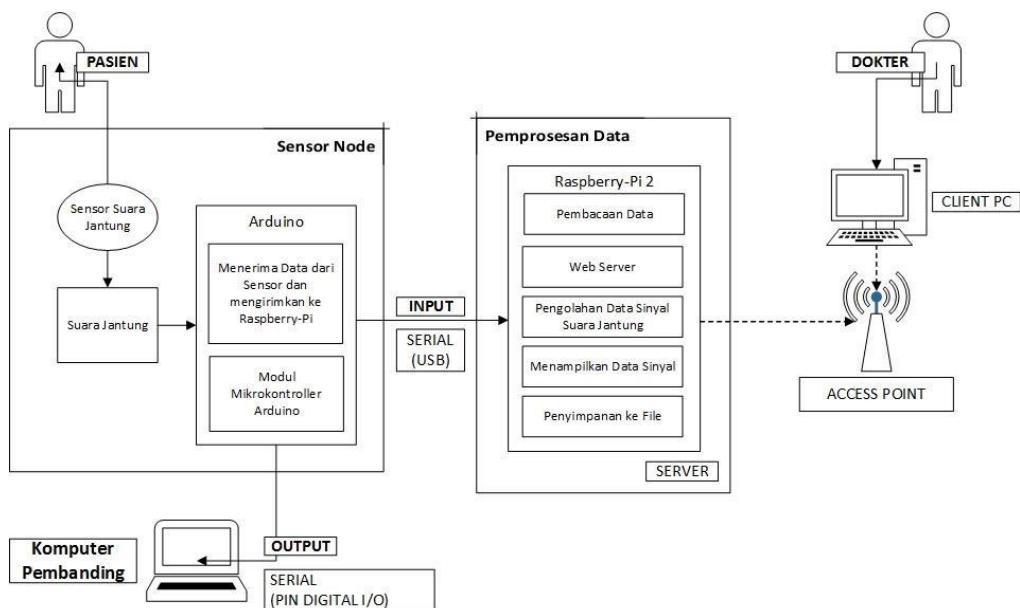
BAB 2

Desain Rancangan IMedT Untuk Monitoring Suara Jantung

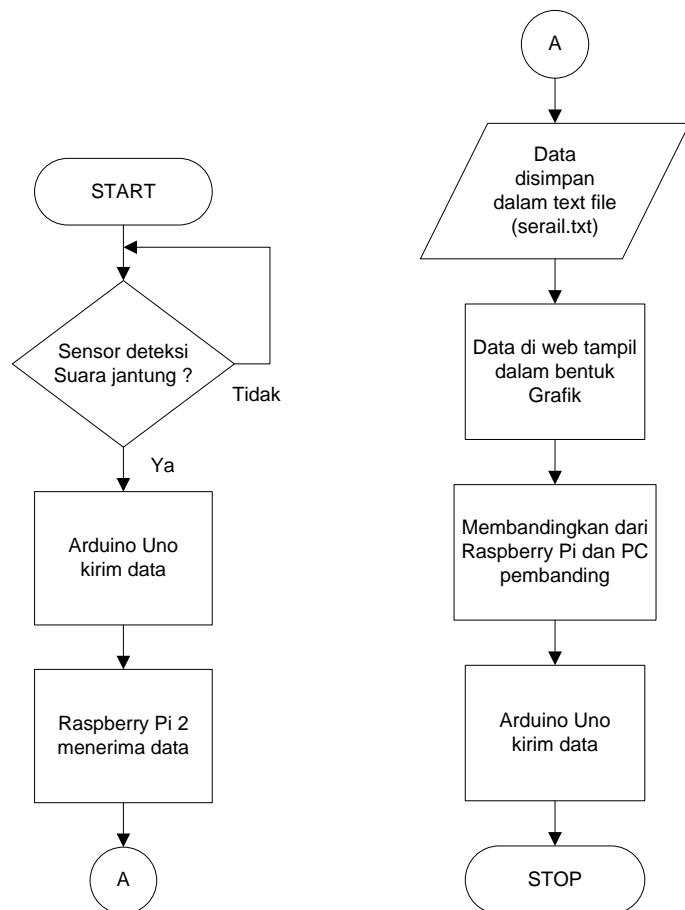
Pada bab 2 membahas diagram blok desain rancangan beserta penjelasan prinsip kerja dari sistem IMedT yang digunakan untuk monitoring aktivitas jantung dalam bentuk sinyal *phonocardiograph* (PCG).

Dalam karya tulis ini akan dijelaskan prinsip kerja dari desain rancangan IMedT yang digunakan untuk memantau aktivitas jantung dalam bentuk sinyal *phonocardiograph* (PCG). Untuk menyediakan layanan dan aplikasi yang handal dalam pemantauan jantung maka diperlukan persyaratan khusus atau minimal yang harus dipenuhi, diantaranya: (i) transmisi secara *real time* dan periodik untuk menunjukkan tanda vital seperti: suhu tubuh, denyut nadi atau detak jantung yang terwakili dari bentuk sinyal ECG atau PCG, tingkat respirasi dan tekanan darah untuk membantu perawat atau dokter sebagai data untuk analisis.

Selanjutnya (ii) akses berkecepatan tinggi dan aman ke jaringan berbasis nirkabel yang memungkinkan transmisi secara periodik dalam waktu nyata (*real time*), (iii) perangkat harus dalam bentuk yang *portable*, *wearable* dan *mobile* sehingga memberikan kenyamanan bagi penggunanya. (iv) piranti harus mempunyai kecerdasan buatan (*Artificial Intelligent*) sehingga dapat mengolah sinyal secara akurat dan presisi sehingga diharapkan bisa memprediksi adanya indikasi awal gagal jantung.



Gambar 2.1 Diagram blok sistem *monitoring* IMedT

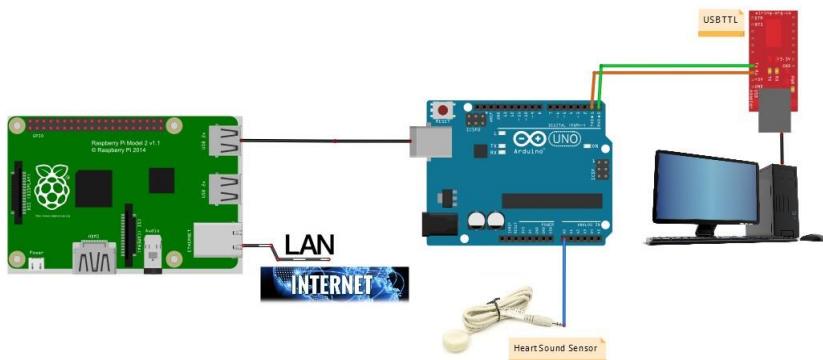


Gambar 2.2 Flowchart sistem *monitoring* IMedT

Sistem pemantauan jantung sebagaimana tampak pada Gambar 2.1 dan Gambar 2.2, terdiri dari beberapa komponen utama yang saling terkait antara satu dengan yang lainnya, diantaranya: bagian sensor *node*, bagian pemrosesan data, *router* serta komputer pembanding. Penjelasan lebih rinci dari *flowchart* sistem monitoring IMedT untuk sinyal suara jantung tersebut sebagai berikut:

1. Bagian sensor *node* terdapat dua alat yaitu *heart sound* sensor dan Arduino Uno, *heart sound* sensor berfungsi untuk mengambil sinyal suara detak jantung pada pasien, kemudian Arduino Uno mengkonversi sinyal analog dari suara jantung menjadi sinyal digital selanjutnya data tersebut dikirim ke Raspberry Pi 2 melalui kabel USB.
2. Bagian pemrosesan data menggunakan Raspberry Pi 2 yang difungsikan sebagai *web server* sebagai tempat untuk menerima, memproses dan menampilkan data sinyal suara jantung dalam bentuk grafik.
4. Untuk transmisi penampilan grafik sinyal suara jantung ke internet agar bisa diakses dari luar menggunakan modem *router*.
5. Komputer pembanding yang terhubung ke Arduino Uno digunakan sebagai pembanding bentuk tampilan grafik sinyal suara jantung terhadap bentuk tampilan yang ada di *web*, hal ini dilakukan untuk mengetahui apakah ada perbedaan atau kehilangan data pada saat proses transmisinya ke internet.

2.1 Perancangan Perangkat Keras



Gambar 2.3 Perancangan perangkat keras sistem *monitoring* IMedT

Perancangan perangkat keras yang dibangun pada penelitian ini secara lengkap tampak seperti pada Gambar 2.3 dengan penjelasan per modul dijelaskan sebagai berikut:

2.1.1 *Heart Sound Sensor*

Untuk mengambil suara jantung diperlukan sensor suara jantung yang sesuai, pada penelitian ini peneliti menggunakan *heart sound sensor* dari *Seeed Studio* untuk pengambilan data sensor suara jantung pada pasien.



Gambar 2.4 *Heart sound sensor* dari *Seeed Studio*.

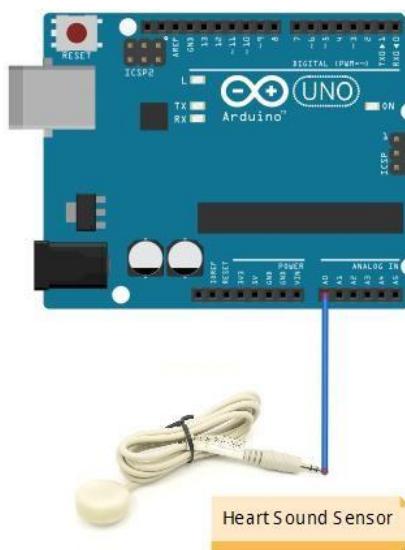
Pada perangkat *heart sound sensor* sudah terintegrasi komponen suara mikro yang dibuat dari material *polymer* yang dapat menghasilkan keluaran sinyal audio impedansi rendah. Dengan menggunakan *design* modular, serta PCB yang bersifat *plug-way closed* maka perangkat ini dapat digunakan secara luas dalam berbagai pengamatan, contohnya seperti pengamatan sinyal suara jantung, pengamatan sinyal suara janin serta dapat juga dilakukan pengamatan lainnya yang berbasis pengamatan suara.

Tabel 3.1 Spesifikasi *Heart Sound Sensor*

<i>Input</i> tegangan	3 – 5V DC
Konsumsi tegangan	5mA
Jarak frekuensi	1 – 600 Hz
Kesensitif-an	> 4mV/Pa
<i>Output</i> tegangan sinyal	0.5 – 1.5 V

2.1.2 Perancangan Sensor Jantung

Untuk mendektsi detak jantung pasien secara elektrik maka diperlukan sensor, sensor yang digunakan pada rancang bangun *monitoring* suara jantung berbasis *web* dengan menggunakan Raspberry Pi 2 sebagai *web server* ini adalah *Heart Sound Sensor*. Sensor ini telah dilengkapi dengan pengondisi sinyal dan *filter* yang bertugas meredam serta mengolah sinyal jantung dan mengkonversinya dalam bentuk tegangan. Dengan demikian keluaran sensor *Heart Sound Sensor* dapat langsung dibaca melalui ADC internal pada Arduino Uno. Adapun perancangan rangkaian *heart sound sensor* ditunjukkan pada Gambar 2.5.

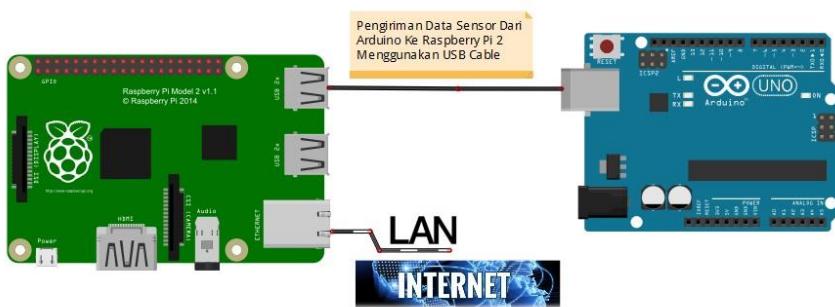


Gambar 2.5 Hubungan *heart sound* sensor dan Arduino.

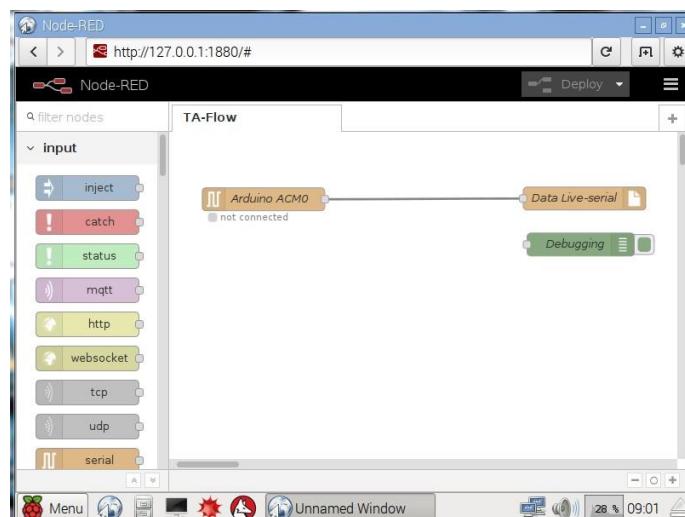
Dalam modul Arduino Uno melalui fungsi *ReadAnalog* memiliki fungsi untuk membaca nilai sinyal analog yang akan dikirimkan datanya ke Raspberry Pi 2 melalui koneksi serial menggunakan kabel USB dalam satuan waktu milidetik dan nilai data yang didapatkan dari sensor suara jantung dengan format pengiriman data : (waktu),(data).

2.1.3 Perancangan Rangkaian Arduino ke Raspberry Pi 2

Data yang di dapat dari sensor supaya dapat ditampilkan pada *Web* maka Arduino harus terkoneksi dengan Raspberry Pi 2 melalui komunikasi secara serial menggunakan kabel USB sebagaimana tampak pada Gambar 2.6. Data yang sudah dikirim oleh Arduino selanjutnya oleh Raspberry Pi 2 diolah menggunakan aplikasi NODE-RED terlihat pada Gambar 2.7 dan disimpan dalam *file* serial.txt, proses ini dilakukan secara berulang-ulang.



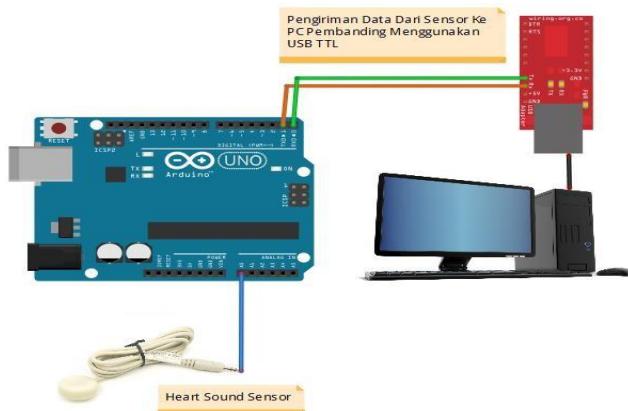
Gambar 2.6 Hubungan Arduino dengan Raspberry Pi 2.



Gambar 2.7 Program penerimaan data dari Arduino ke Raspberry Pi 2

2.1.4 Perancangan Rangkaian Arduino ke PC Pembanding

Untuk mendeteksi adanya perbedaan data yang hilang atau tidak terkirim antara di Arduino Uno dan yang diterima oleh Raspberry Pi 2 maka diperlukan PC pembanding seperti ditunjukkan pada Gambar 2.8.



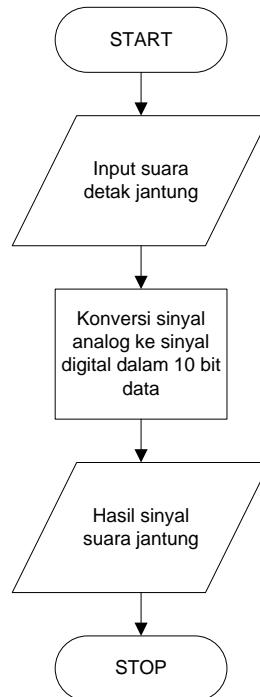
Gambar 2.8 Hubungan Arduino Uno dengan PC pembanding

2.2 Perancangan Perangkat Lunak

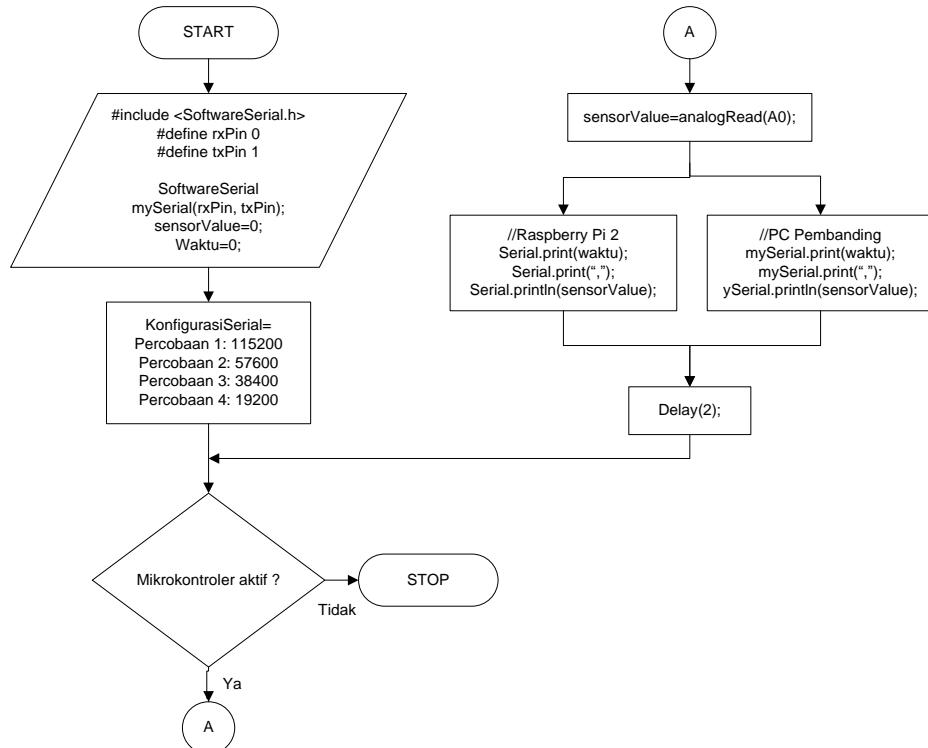
Setelah perangkat lunak sistem monitoring suara jantung dibangun, langkah berikutnya adalah diperlukan perancangan perangkat lunaknya. Perangkat lunak ini terdiri dari beberapa algoritma dalam sistem perancangannya, diantaranya: pembacaan sinyal jantung dan pengiriman datanya dari Arduino Uno, penerimaan data di Raspberry Pi 2, pengolahan data secara *real time* di Raspberry Pi 2 serta pengolahan data secara *offline* di Raspberry Pi 2, untuk lebih jelasnya masing-masing algoritma tersebut akan dibahas satu per satu.

2.2.1 Algoritma Pembacaan Sinyal Jantung dan Pengiriman Data

Sebagaimana *flowchart* pada Gambar 2.9, pembacaan sinyal analog dari sensor menggunakan fungsi *analogRead*, selanjutnya sinyal analog dikonversi menjadi data digital dengan resolusi 10 bit agar sinyal analog yang dibaca lebih presisi saat dikonversi ke digital.



Gambar 2.9 *Flowchart* pembacaan pada *heart sound* sensor



Gambar 2.10 *Flowchart* perbandingan bentuk sinyal auskultasi jantung

Pengiriman data dalam bentuk string atau karakter melalui kabel USB dan USB TTL yang terhubung ke Arduino pada pin *rx* dan *tx*. Pengiriman dilakukan dengan *delay* 2 ms karena sesuai dengan ketetapan 2 kali frekuensi *sampling* jantung normal (Lynn, 1994).

$$\text{frekuensi jantung } (fn) = 20 \text{ Hz} - 400 \text{ Hz}$$

$$\begin{aligned}\text{frekuensi sampling} &= 2 \times fn \\ &= 2 \times 200 \text{ Hz} \\ &= 400 \text{ Hz}\end{aligned}$$

$$\text{Periode sampling} = \frac{1}{400} = 2,5 \text{ ms} \approx 2 \text{ ms}$$

Berikut pemrograman modul Arduino Uno pada sensor *node* yang diprogram pada *software* Arduino IDE:

a. Pembuatan *variable*

Pada penelitian ini *variable* tipe *string* yang digunakan diberi nama “*sensorValue*” sebagai tempat untuk menampung data dari sensor suara detak jantung. Perintah “*#include <SoftwareSerial.h>*” berfungsi untuk memanggil *library SoftwareSerial* yang digunakan untuk mengirimkan data ke PC pembanding menggunakan RX pada *pin* 0 dan TX di *pin* 1 dengan perintah “*#define rxPin 0*”, “*#define txPin 1*”, dan “*SoftwareSerial mySerial(rxPin, txPin);*” untuk mendefinisikan letak *pin* rx, tx serta membuat nama *variable* *mySerial* dalam pengiriman pada fungsi rx tx. Kemudian membuat *variable* tipe *string* dengan nama “*waktu*” untuk menampung waktu pada saat mendapatkan data. Pembuatan *variable* ini diletakkan diluar fungsi *void* agar *variable* ini dapat digunakan secara global.

```
#include <SoftwareSerial.h>
#define rxPin 0
#define txPin 1
SoftwareSerial mySerial(rxPin, txPin);
```

```
int sensorValue = 0;
unsigned long waktu;
```

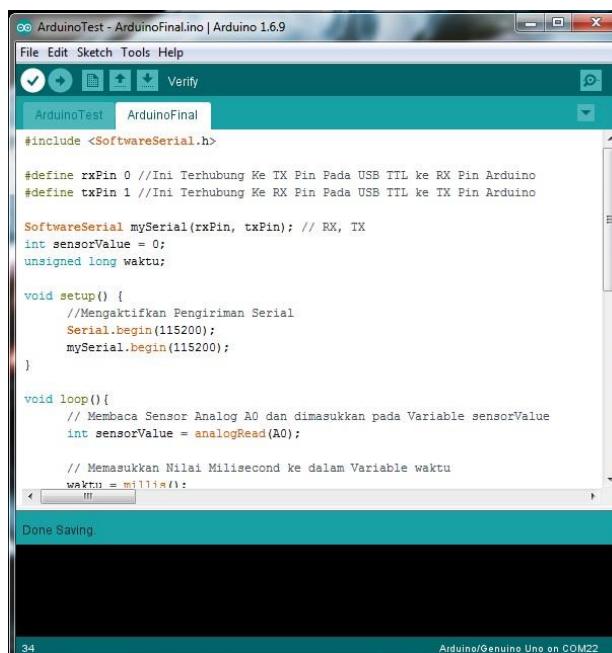
b. Fungsi void setup

Dalam fungsi *void setup* perintah hanya akan dibaca satu kali setelah program di-*running*. Adapun nilai *baudrate* yang dikonfigurasi adalah 115200 untuk pengiriman serial ke Raspberry Pi 2 dan PC pembanding.

```
void setup() {
//Mengaktifkan Pengiriman Serial
Serial.begin(115200);
mySerial.begin(115200);
}
```

c. Fungsi void loop

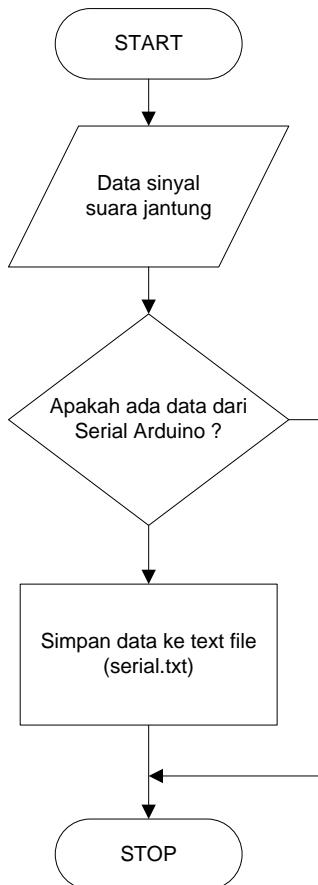
Pada fungsi *void loop* perintah akan dibaca berulang kali selama mikrokontroler tersambung diberi tegangan. Program ditulis sesuai dengan algoritma yang telah dibuat seperti pada Gambar 2.11.



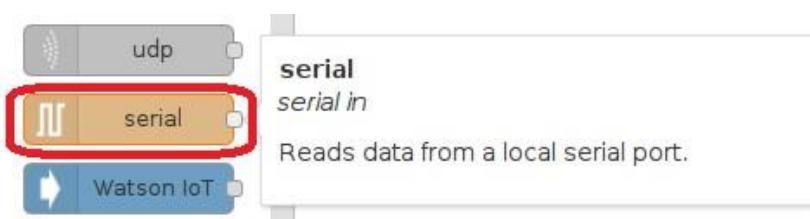
Gambar 2.11 Tampilan program pada *software* Arduino IDE

2.2.2 Algoritma Penerimaan Data pada Raspberry Pi 2

Data sinyal suara jantung yang diterima oleh Raspberry Pi dari proses transmisi yang dikirimkan dari Arduino disimpan dalam *file* serial.txt terlihat *flowchart* pada Gambar 2.12, konfigurasi serial pada program NODE-RED tampak pada Gambar 2.13, selanjutnya dilakukan konfigurasi penerimaan data pada Raspberry Pi sebagaimana pada Gambar 2.14.



Gambar 2.12 *Flowchart* penerimaan data pada Raspberry Pi 2

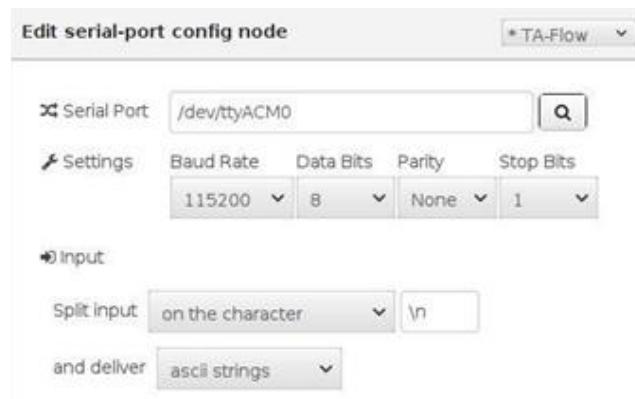


Gambar 2.13 Program penerimaan data serial di NODE-RED



Gambar 2.14 Konfigurasi penerimaan data ke Raspberry Pi 2

Isi konfigurasi serial *port* di NODE-RED dengan “/dev/ttyACM0” sebagai identitas Arduino Uno di Raspberry Pi 2, ditentukan juga setting *baudrate*, *data bits*, *parity*, *stopbits* untuk pengaturan penerimaan serial dari Arduino Uno terlihat pada Gambar 2.15.

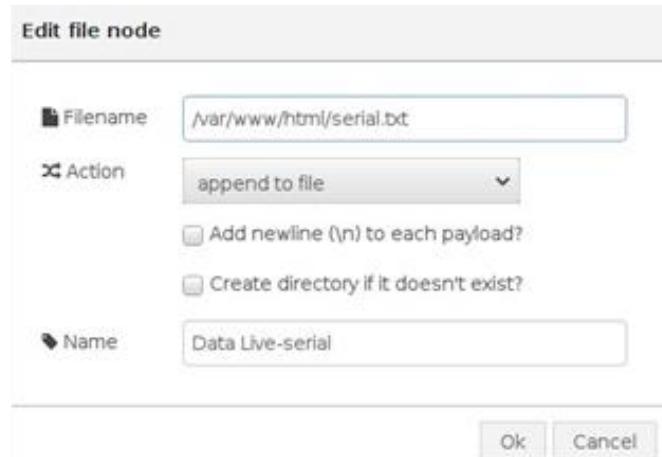


Gambar 2.15 Lanjutan konfigurasi serial *port* di NODE-RED

Untuk proses penyimpanan data dari NODE-RED pada pilihan tombol menu *file*, berikutnya *file* yang diberi nama *serial.txt* disimpan pada *directory* “/var/www/html/ serial.txt” tampak pada Gambar 2.16.



Gambar 2.16 Pilihan menu *file* program NODE-RED



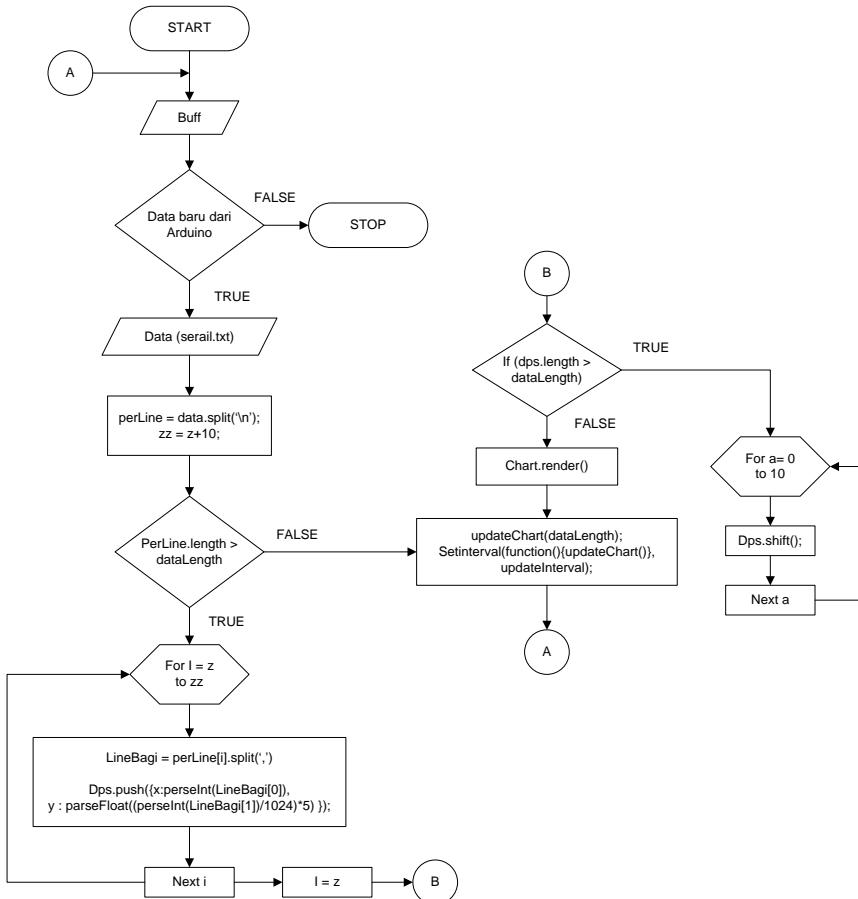
Gambar 2.17 Directory penyimpanan file serial.txt

2.2.3 Algoritma Pengolahan Data secara *Realtime* pada Raspberry Pi 2

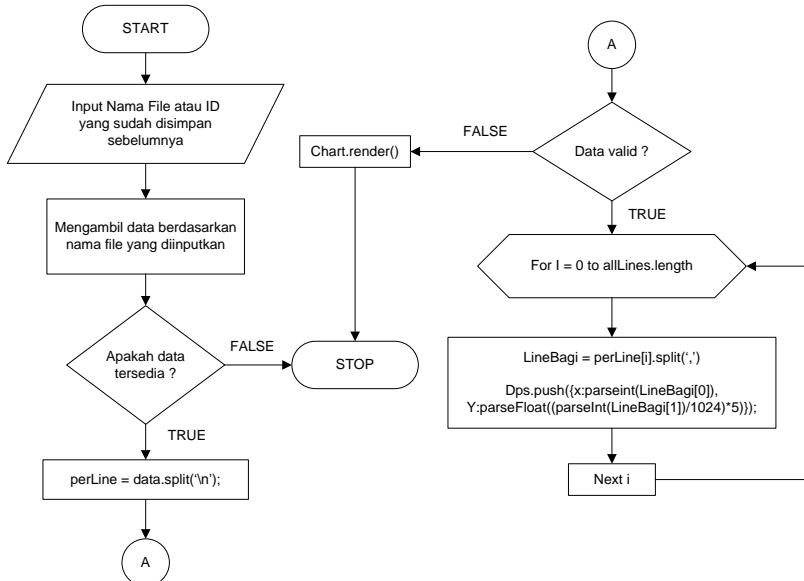
Raspberry Pi mengolah satu data yang memiliki dua data berbeda maka diperlukan pemisahan data untuk pengelompokkan data tersebut sesuai dengan kategori asal data. Pengelompokkan nilai data pertama merupakan waktu dan nilai kedua merupakan nilai dari sensor, pada saat data dipisahkan maka bisa dilakukan analisa data karena secara otomatis data tersimpan dalam satu *file*, *flowchart* pengolahan data secara *live* pada Raspberry Pi 2 tampak pada Gambar 2.18.

2.2.4 Algoritma Pengolahan Data secara *Offline* pada Raspberry Pi 2

Pada dasarnya pengolahan data secara *offline* sama saja dengan pengolahan data secara *realtime*, perbedaannya adalah pada pengolahan data secara *realtime* data terlebih dahulu disimpan kedalam sebuah *file* dengan nama yang sudah ditentukan oleh *user* sebelumnya. Data akan diambil dari *file* kemudian ditampilkan dalam bentuk grafik pada saat *user* memasukkan nama *file* tersebut, *flowchart* pengolahan data secara *offline* dapat dilihat pada Gambar 2.19.



Gambar 2.18 Flowchart pengolahan data secara *realtime* di Raspberry Pi 2



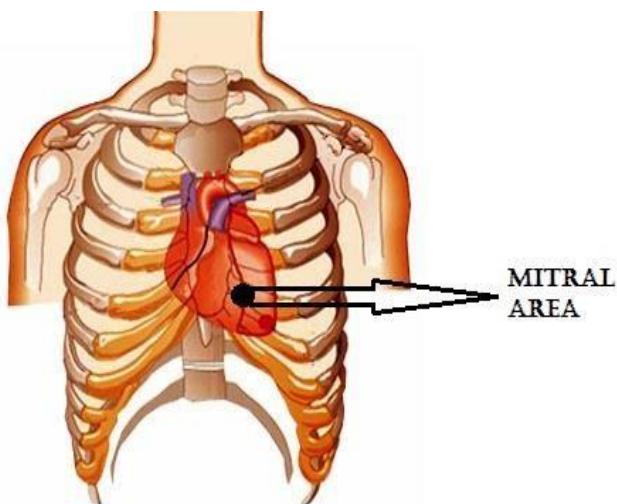
Gambar 2.19 Flowchart pengolahan data secara *offline* di Raspberry Pi 2

2.3 Metode Analisa

Setelah melakukan proses perancangan dan pembuatan algoritma pengiriman, penerimaan, pemrosesan data serta proses transmisi data sinyal suara detak jantung ini, langkah berikutnya adalah proses analisa pengirimannya untuk mengetahui performansi dari sistem yang telah dirancang.

2.3.1 Peletakan Sensor Suara Detak Jantung pada Tubuh

Kualitas transmisi auskultasi dari data sinyal suara detak jantung dipengaruhi oleh peletakan posisi sensor yang tepat pada area jantung *pasien*. Posisi jantung manusia berada pada tulang iga manusia ke-6 di bagian dada sebelah kiri atau 5 cm diatas ulu hati. Peletakkan posisi sensor yang tidak tepat akan mempengaruhi kualitas penerimaan data karena adanya peningkatan *noise* data. Posisi Mitral area sebagai pusat titik deteksi dapat dilihat pada Gambar 2.20.

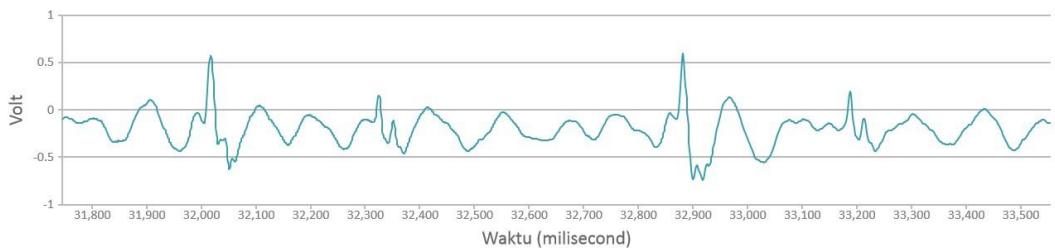


Gambar 2.20 Letak posisi penempatan sensor suara detak jantung

2.3.2 Pengambilan Data Sinyal Suara Detak Jantung

Proses pengambilan, penyimpanan serta pengiriman data dilakukan saat semua alat sudah terpasang, data tersimpan pada sebuah *file* serial.txt, *file* inilah yang menampilkan data dalam bentuk grafik secara *live* maupun *offline*. Tampilan

data dalam bentuk grafik inilah yang akan dianalisa untuk mengetahui performansi transmisi sinyal suara detak jantung dari Arduino ke Raspberry Pi 2 dan PC pembanding yang dilakukan secara *streaming*. Contoh bentuk tampilan grafik sinyal suara detak jantung terlihat pada Gambar 2.21.



Gambar 2.21 Hasil sinyal suara detak jantung

2.3.3 Analisa Transmisi Sinyal Suara Detak Jantung

Cara menganalisa hasil transmisi data sinyal suara detak jantung adalah dengan memindahkan data dari *file* penyimpanan serial.txt menjadi *file* excel yang berekstensi .xls. Selanjutnya dilakukan proses analisa melalui perbandingan bentuk tampilan grafik dari Raspberry Pi 2 dengan data yang berasal dari PC pembanding. Pada Gambar 2.22 dan Gambar 2.23, hasilnya menunjukkan bahwasannya proses pengiriman data mempunyai bentuk tampilan dan waktu yang sama, hanya terdapat empat data yang mengalami *delay*.

Perbandingan Data Sinyal Suara Jantung Menggunakan Baudrate 115.200					
	Waktu Terima Raspberry PI 2	Nilai Data	-Checking-SAMA / TIDAK	Waktu Terima PC Pembanding	Nilai Data
2					
3	2330 522	SAMA		2330 522	
4	2332 521	SAMA		2332 521	
5	2335 523	SAMA		2335 523	
6	2337 526	SAMA		2337 526	
7	2340 527	SAMA		2340 527	
8	2342 526	SAMA		2342 526	
9	2345 521	SAMA		2345 521	
10	2348 515	SAMA		2348 515	
11	2351 508	SAMA		2351 508	
12	2353 503	SAMA		2353 503	
13	2356 501	SAMA		2356 501	
14	2358 501	SAMA		2358 501	
15	2361 502	SAMA		2361 502	
16	2363 503	SAMA		2363 503	
17	2366 506	SAMA		2366 506	
18	2368 512	SAMA		2368 512	

HASIL		
JUMLAH / BANYAK DATA	47159	
TOTAL DATA TIDAK SAMA / LOSS	2	47159
TOTAL DATA SAMA	47157	

LEGEND	
= DATA SAMA	= DATA TIDAK SAMA

Gambar 2.22 Hasil komparasi kesesuaian data yang dikirim dan diterima

1078	2729 573	SAMA	2729 573	
1079	2735 572	TIDAK_SAMA	2733 571	
1080	2733 571	TIDAK_SAMA	2735 572	
1081	2738 576	SAMA	2738 576	
1082	2740 583	SAMA	2740 583	
1083	2743 592	SAMA	2743 592	
1084	2745 601	SAMA	2745 601	
1085	2748 605	SAMA	2748 605	
1086	2750 604	SAMA	2750 604	
1087	2753 600	SAMA	2753 600	
1088	2756 592	SAMA	2756 592	
1089	2758 578	SAMA	2758 578	
1090	2761 564	SAMA	2761 564	
1091	2763 556	SAMA	2763 556	
1092	2766 551	SAMA	2766 551	
1093	2768 547	SAMA	2768 547	
1094	2771 545	SAMA	2771 545	
1095	2774 545	SAMA	2774 545	
1096	2777 545	SAMA	2777 545	
1097	2779 542	SAMA	2779 542	
1098	2784 514	TIDAK_SAMA	2782 532	
1099	2782 532	TIDAK_SAMA	2784 514	
1100	2787 497	SAMA	2787 497	

Gambar 2.23 Hasil komparasi ketidaksesuaian data yang diterima

Selanjutnya dilakukan perhitungan besar *throughput* rata – rata dari setiap pengiriman data, *throughput* adalah *bandwidth* aktual yang terukur pada satuan waktu tertentu.

$$\text{Throughput} = \frac{\Sigma \text{ data masuk} \times \Sigma \text{ tiap paket data} \times \text{besar pengiriman data}}{\text{Lama pengamatan}}$$

Penjelasan rumus sebagai berikut:

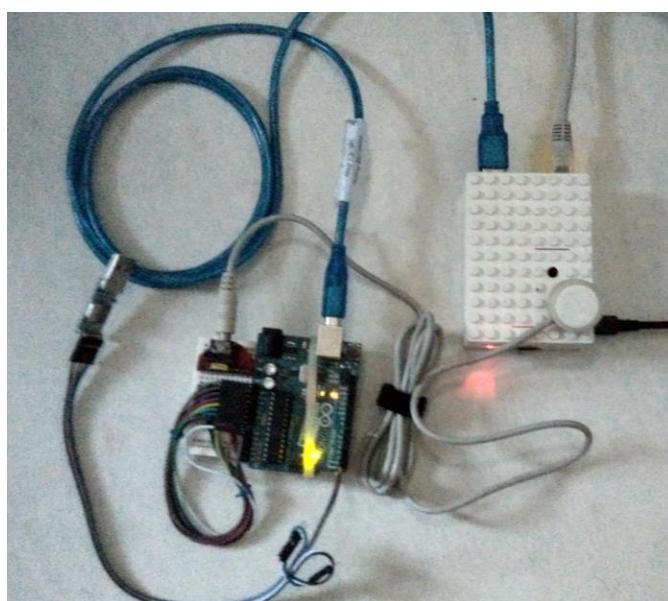
- Jumlah data masuk merupakan keseluruhan jumlah data yang dikirimkan setiap waktu pengiriman
- Jumlah tiap *packet data* adalah jumlah data yang dikirimkan, 1 buah paket data terdapat antara 6 karakter sampai dengan 10 karakter.
- Besar pengiriman data: satu karakter terbentuk dari 10 bit data, yaitu 8 *bit* untuk setiap karakter, dan 1 *bit* pembuka data, dan 1 *bit* penutup data.

BAB 3

Membangun Purwarupa IMedT Untuk Monitoring Suara Jantung

Materi bab 3 ini menjelaskan bagaimana cara membangun IMedT untuk monitoring Cardiovaskuler secara perangkat keras (*hardware*) maupun perangkat lunak (*software*).

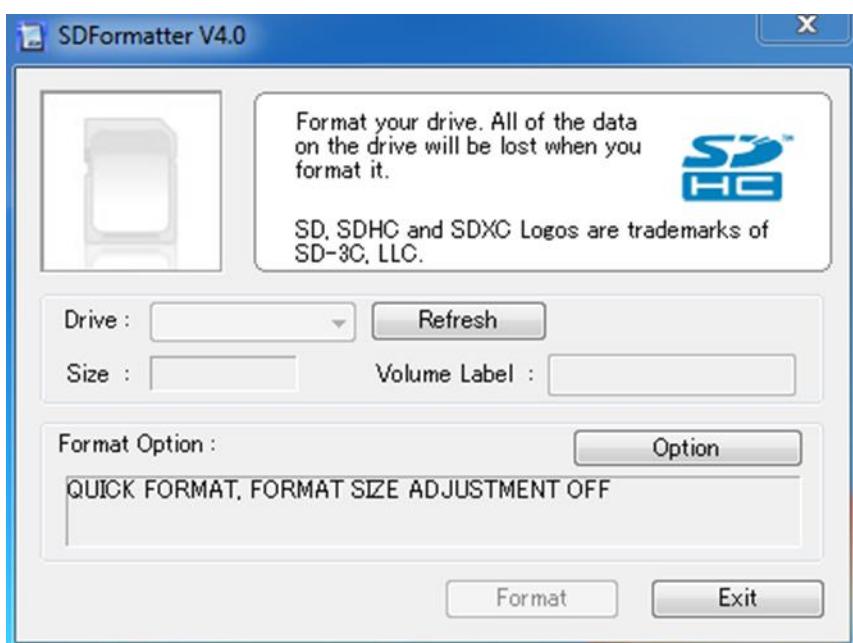
Bentuk fisik dan koneksi kabling pada IMedT untuk sistem monitoring suara jantung tampak pada Gambar 3.1, urutan koneksinya dimulai dari sensor PCG yang terkoneksi ke *pin* Arduino Uno, *output* pertama dari Arduino Uno dihubungkan ke Raspberry Pi melalui koneksi kabel USB sedangkan *output* kedua dari Arduino dihubungkan ke PC pembanding, selanjutnya konektor RJ-45 (LAN) dari Raspberry Pi dikoneksikan ke modem *router* Speedy yang terkoneksi ke jaringan internet.



Gambar 3.1 Tampilan koneksi peralatan IMedT

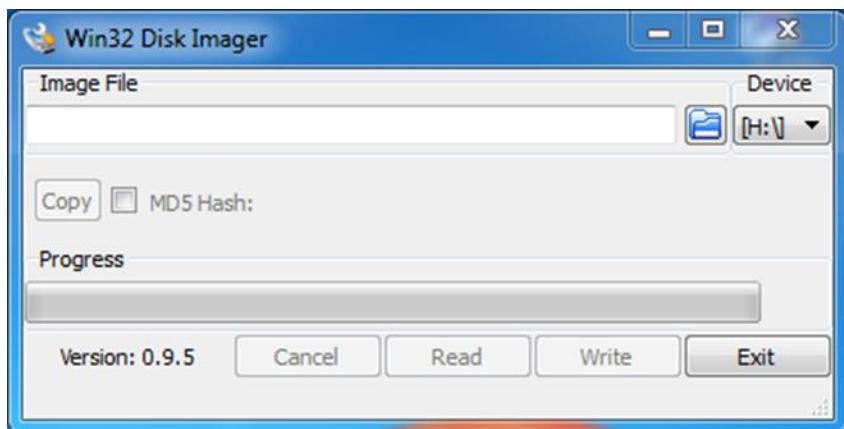
3.1 Instalasi Raspberry Pi 2

Raspberry Pi 2 pada penelitian ini berfungsi sebagai *web server* dari IMedT, sebelum dapat digunakan secara penuh maka ada beberapa hal atau langkah yang harus dilakukan, yaitu: diperlukan proses instalasi, setting dan konfigurasi di Raspberry Pi 2. Media penyimpan (*storage*) pada Raspberry Pi menggunakan Ultra SD *Card* dengan kapasitas yang disarankan adalah minimal 8 Gb dengan spesifikasi *class* 10 (C-10), sebelum di-*install* dengan sistem operasi Raspbian (Raspberry Debian) yang merupakan salah satu jenis sistem operasi yang khusus berjalan diatas Raspberry Pi maka SD *Card* tersebut terlebih dulu di *format* dengan menggunakan aplikasi SDFormatter V4.0 sebagaimana tampak pada Gambar 3.2.



Gambar 3.2 Tampilan *software* untuk mem-*format* SD *Card*

Proses instalasi sistem operasi di Raspberry Pi berbeda dengan proses instalasi sistem operasi Windows, Mac dan Linux, hal ini dikarenakan instalasi sistem operasi di Raspberry Pi dilakukan dengan cara mengekstraksinya menggunakan dengan salah *software* seperti Win32 Disk Imager sebagaimana tampak pada Gambar 3.3.

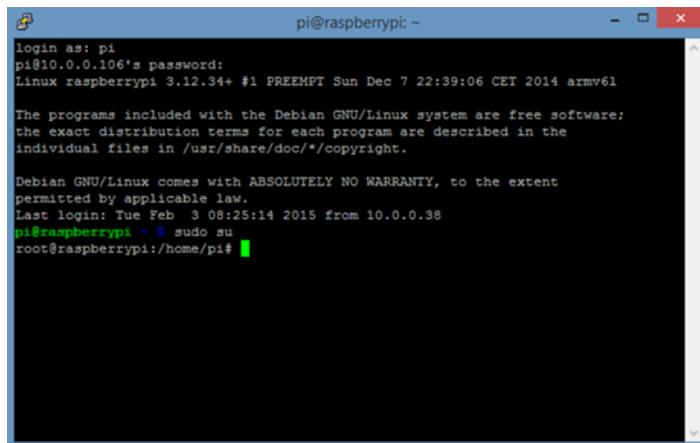


Gambar 3.3 Tampilan *software* untuk ekstraksi sistem operasi Raspbian

3.2 Update dan Upgrade Raspbian

File image sistem operasi Raspbian Jessie atau Raspbian Pixel dapat diunduh dari alamat *web*: <https://www.raspberrypi.org/downloads>, proses instalasi sistem operasi tersebut dengan cara diekstraksi ke *SD Card* menggunakan aplikasi Win32 Disk Imager yang dapat diunduh dengan alamat *web*: https://www.sdcard.org/downloads/formatter_4/eula_windows. Setelah proses instalasi sudah selesai, disarankan langkah berikutnya adalah melakukan proses *update* dan *upgrade*, proses *update* dan *upgrade* ini bertujuan untuk meningkatkan versi dan fitur yang terbaru secara daring.

Proses *update* dan *upgrade* dilakukan melalui terminal CLI dengan perintah `~$ sudo apt-get update` dan `~$ sudo apt-get upgrade`. Kedua proses tersebut memerlukan koneksi jaringan dengan estimasi waktu yang diperlukan antara 30 menit sampai dengan 40 menit tergantung dari versi Raspberry Pi dan kecepatan koneksi jaringannya. Untuk melihat sisa *space* atau kapasitas dari *SD Card*, setelah proses *update* dan *upgrade* bisa menggunakan perintah `~$ df -h`, Gambar 3.4 adalah bentuk tampilan dari antarmuka menu Terminal.



Gambar 3.4. Tampilan antarmuka menu Terminal

3.3 Pengujian Komunikasi Arduino Raspberry Pi 2 dan PC Pembanding

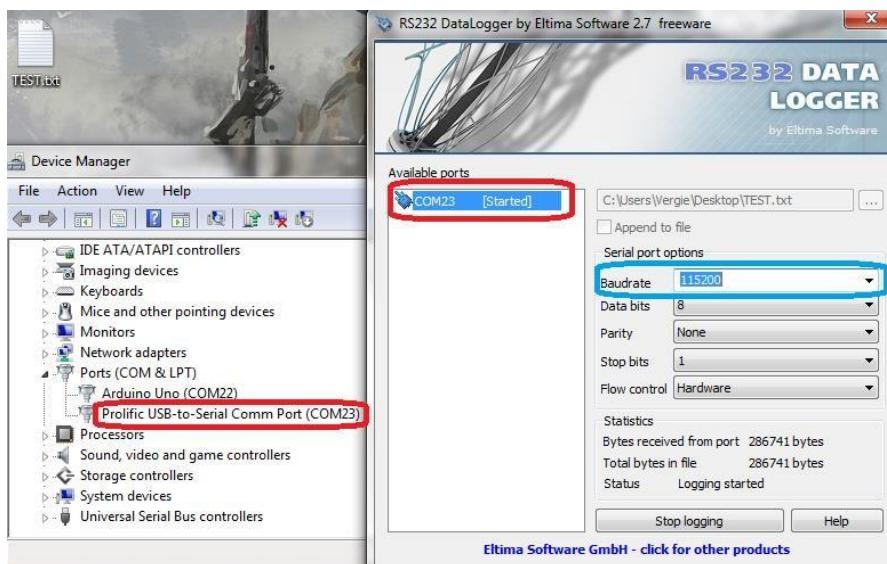
Pengujian komunikasi Arduino ke Raspberry Pi 2 dilakukan dengan mengatur *baudrate* pada kedua alat tersebut, hal ini diperlukan untuk mengetahui apakah Arduino, Raspberry Pi 2 serta ke PC pembanding dapat berkomunikasi dengan baik antara yang satu dengan yang lain setelah mendapatkan data dari sensor PCG.

Baudrate Pada Arduino, Raspberry Pi 2 dan PC pembanding di samakan nilainya, yaitu: 115.200. Untuk mengecek apakah Raspberry Pi 2 sudah terkoneksi dengan Arduino, maka pada antarmuka Terminal diketikkan perintah “*ls /dev/tty**” apakah terdapat tampilan dengan nama “*ttyACMO*” apabila ada hal itu menandakan bahwa Arduino sudah terkoneksi dengan Raspberry Pi, seperti pada Gambar 3.5.

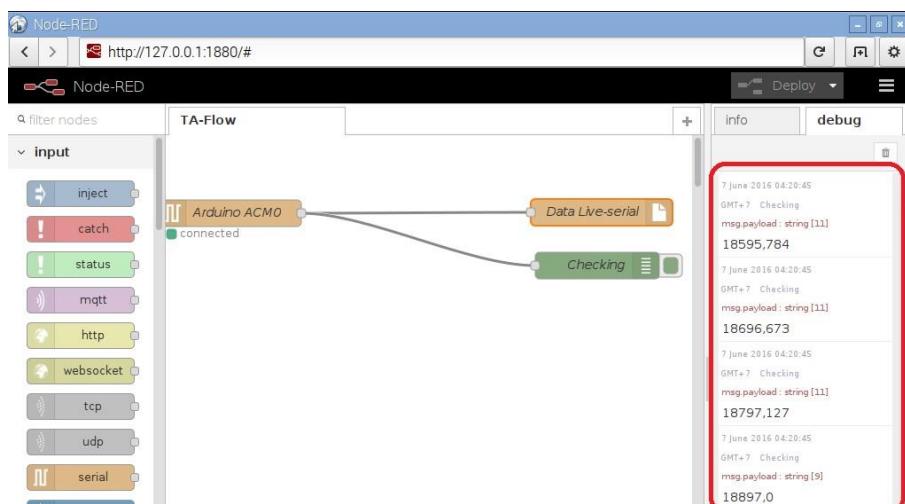
```
pi@raspberrypi:/dev $ ls tty*
tty   tty14  tty20  tty27  tty33  tty4   tty46  tty52  tty59  tty8
tty0  tty15  tty21  tty28  tty34  tty40  tty47  tty53  tty6   ttyv9
tty1  tty16  tty22  tty29  tty35  tty41  tty48  tty54  tty60  ttyACMO
tty10 tty17  tty23  tty3   tty36  tty42  tty49  tty55  tty61  ttyAMA0
tty11 tty18  tty24  tty30  tty37  tty43  tty5   tty56  tty62  ttyprintk
tty12 tty19  tty25  tty31  tty38  tty44  tty50  tty57  tty63
tty13 tty2   tty26  tty32  tty39  tty45  tty51  tty58  tty7
```

Gambar 3.5 Tampilan Arduino terkoneksi pada Terminal di Raspberry Pi 2

Dikarenakan pada PC pembanding menggunakan sistem operasi Windows maka diperlukan instalasi *driver* USB TTL terlebih dahulu, guna memastikan bahwa *port* USB sudah aktif (*enable*) bisa dilihat dari *Device Manager*. Berikutnya dari *software* RS232 DataLogger yang sudah ter-*install* akan terlihat aktivasi *port* USB pada COM 23 tampak pada Gambar 3.6 dengan konfigurasi *baudrate* yang digunakan sama di Arduino dan Raspberry Pi, yaitu: 115200.



Gambar 3.6 Tampilan Arduino terkoneksi pada PC pembanding



Gambar 3.7 Komunikasi *serial* pada Arduino dan Raspberry Pi 2

Pada Gambar 3.7, tulisan yang ada pada kotak berwarna merah merupakan Raspberry Pi 2 sedang menerima data dari Arduino dengan menggunakan Software NODE-RED yang sudah *pre-installed* pada sistem operasi RASPBIAN (Raspberry Debian).

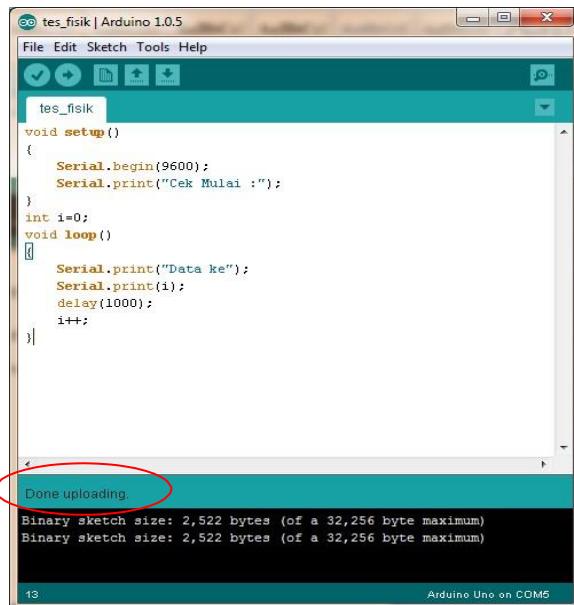
3.4 Pengujian Arduino

Diperlukan pengujian Arduino untuk memastikan bahwasanya Arduino tidak mengalami kerusakan atau terjadi kerusakan sehingga sistem bisa berjalan sesuai dengan harapan, cukup mengetikkan skrip program sederhana pada aplikasi Arduino IDE dari komputer melalui kabel penghubung USB.

```
void setup()
Serial.begin(9600);
Serial.println("Cek Mulai :");
}
int i=0;

void loop()
{
Serial.print("Data ke"); Serial.println(i); delay(1000);
i++;
}
```

Langkah berikutnya klik tombol “*Verify*” untuk mengecek apakah ada perintah atau sintak perintah yang salah dalam bahasa C, jika tidak ada pesan kesalahan yang tampil maka proses berikutnya bisa dilanjutkan dengan cara menekan tombol meu “*Upload*” yang berfungsi untuk meng-*compile* serta menyimpan hasil program tersebut ke *board* Arduino Mega 2560 terlihat seperti pada Gambar 3.8. Untuk me-*monitor* aktivitas pengiriman data yang terjadi pada *port* serial apakah sudah sesuai dengan harapan maka bisa di-klik tombol menu *icon Serial monitor* pada bagian pojok kanan atas sebagaimana terlihat pada Gambar 3.9.



Gambar 3.8 Upload program berhasil



Gambar 3.9 Program berhasil berjalan

BAB 4

Evaluasi Tampilan Grafik IMedT Monitoring Suara Jantung Secara *Offline* Dan *Online*

Komparasi bentuk tampilan grafik sinyal suara jantung yang berasal dari akses *web* secara *realtime (online)* dan tampilan grafik sinyal suara jantung dari *file* rekap data pasien (*offline*).

4.1 Pengujian Tampilan Penerimaan Data pada Raspberry Pi 2 secara *live*

Pengujian pertama yang dilakukan adalah pengujian menggunakan aplikasi berbasis *web* untuk mengetahui apakah sinyal suara jantung bisa diterima atau ditampilkan dengan baik setelah melalui proses transmisi via jaringan internet.

4.1.1 Prosedur Pengujian

- Menghubungkan Arduino Uno dan Raspberry Pi 2 dengan kabel USB.
- Menghubungkan Raspberry Pi 2 dan komputer ke *router* dengan menggunakan kabel UTP.
- Mengaktifkan *router* dan komputer.
- Mengaktifkan Raspberry Pi 2 dan buka program *Terminal*.
- Meletakkan sensor secara tepat pada jantung agar mendapatkan sinyal jantung yang baik dengan *noise* sekecil mungkin.
- Membuka aplikasi *browser* dari komputer atau *smartphone*.
- Mengetikkan alamat dari Raspberry Pi 2, setelah selesai me-*load* halaman utama *website* maka selanjutnya lihat Bagian Menu – lalu pilih Live Data Sinyal Jantung
- Melakukan pengambilan data selama 120 detik, untuk memperoleh sinyal jantung.

- Mengamati data, apakah data dapat diterima oleh aplikasi dan sinyal yang ditangkap merupakan sinyal jantung.
- Jika selesai mengamati dan ingin menyimpan data tersebut pilih *Save Data*, lalu masukkan ID atau Nama yang diinginkan. Setelah selesai pilih Simpan.

4.1.2 Hasil Pengujian

Web server bisa diakses melalui program *browser* Google Chrome, Mozilla Firefox dan sebagainya dari PC, *Laptop* maupun *Handphone* dengan nomer IP Address 192.168.255.250. Pada Gambar 4.1 menampilkan tampilan program utama *web* dari IMedT monitoring suara jantung yang mempunyai 9 tombol menu utama, diantaranya: Live Data Sinyal Jantung, Buka Rekap Data Sinyal Jantung, Lihat Rekap Data Semua Pengujian Sinyal Jantung, Data Master Website, Kecepatan Data Menggunakan Baudrate 19.200, Kecepatan Data Menggunakan Baudrate 38.400, Kecepatan Data Menggunakan Baudrate 57.800, Kecepatan Data Menggunakan Baudrate 115.200 dan Download Semua File Perbandingan Excel Percobaan 1.



Gambar 4.1 Tampilan menu utama *website* IMedT

Pada Gambar 4.2 manampilkan grafik sinyal jantung dalam satuan milidetik beserta pembacaan *log* data secara *live* yang diakses dari mini komputer

Raspberry Pi 2, berdasarkan grafik tersebut hal ini menunjukkan data telah diterima dan diakses dengan baik oleh Raspberry Pi 2. Pembacaan data dilakukan melalui kabel serial melalui aplikasi *NODE-RED* pada file serial.txt yang dibuat secara otomatis oleh program.



Gambar 4.2 Tampilan pengaksesan data secara *live*

Setelah dilakukan pengaksesan data secara *live*, pengguna dapat menyimpan sinyal suara jantung dengan nama *file* sesuai nama pasien yang sedang diamati seperti pada Gambar 4.3. *File* yang sudah disimpan tersebut dapat diakses kembali dapat dilihat *user* dapat menyimpan kedalam id atau nama *file* yang diinginkan untuk dapat dibuka kembali melalui tombol menu *LOG LIVE* pada Gambar 4.1.



Gambar 4.3 Penyimpanan data *live*

Hasil sinyal suara jantung yang dipresentasikan kedalam grafik merupakan hasil sinyal setelah dirubah kedalam tegangan, cara merubah data menjadi tegangan dilakukan dengan menggunakan rumus:

$$x = \text{data} / 1024 \times 5 - 2,5$$

Keterangan rumus sebagai berikut:

- Pembagian dengan angka 1024, dilakukan karena sinyal auskultasi jantung telah dikonversi menjadi data ADC dengan resolusi 10 bit.
- Perkalian dengan angka 5, dikarenakan data diambil dari tegangan antara 0V – 5V.
- Pengurangan dengan angka 2,5 agar data yang terambil berada pada posisi tengah atau 0 saat ditampilkan dalam bentuk grafik.

Pada saat pengambilan data jantung selain peletakkan posisi sensor PCG jantung yang tepat, kualitas hasil dari sinyal auskultasi jantung juga dipengaruhi oleh gerakan-gerakan yang dilakukan oleh pasien, seperti ketika pasien atau subjek berteriak, berjalan atau atau aktivitas yang lainnya maka nilai hasil sinyal akan menunjukkan angka yang sangat tinggi atau justru sangat rendah.

4.2 Pengujian Tampilan Rekap Data pada Raspberry Pi 2 secara *Offline*

Pengujian ini merupakan pengujian tampilan rekap data pada aplikasi berbasis *web* dari Raspberry Pi 2, pengujian ini bertujuan untuk mengetahui apakah aplikasi pada Raspberry Pi 2 dapat mempresentasikan sinyal jantung dengan baik dalam bentuk grafik yang ada pada *website* serta dapat menampilkan hasil grafik sinyal suara detak jantung dari sebuah *file* yang diperoleh dari hasil penerimaan data Arduino.

4.2.1 Prosedur Pengujian

Prosedur yang dilakukan hampir sama dengan prosedur yang dilakukan secara *online*, perbedaannya adalah pada pilihan menu yang dipilih dari tombol menu utama yang ada di *web*. Dimana pilihan menu secara *online* maka grafik yang tampil adalah bentuk grafik *realtime* dari pasien, sedangkan secara *offline* bentuk tampilan grafiknya pengguna yang dipilih dari *file* pasien yang telah tersimpan, hal ini diperlukan untuk mengantisipasi jika ada permasalahan serius maupun kondisi yang mendadak dimana pada suatu saat mungkin data tersebut diperlukan lagi atau pada hari, jam, tanggal tersebut dokter karena kesibukkannya sebelum sempat memonitor secara langsung sehingga informasinya tidak hilang apabila diperlukan proses analisis lebih lanjut.

Berikut langkah-langkah operasional yang akan dilakukan apabila prosedur pengaksesan secara *offline*:

- Menghubungkan Arduino Uno dan Raspberry Pi 2 menggunakan kabel USB.
- Menghubungkan Raspberry Pi 2 dan komputer ke *router* dengan menggunakan kabel UTP.
- Mengaktifkan *router* dan komputer.
- Mengaktifkan Raspberry Pi 2, kemudian membuka program *Terminal*.
- Meletakkan sensor PCG secara tepat pada jantung.
- Membuka aplikasi *browser* dari komputer
- Mengetikkan alamat IP *Address* dari Raspberry Pi 2 dari halaman utama *website*, memilih bagian Menu kemudian pilih Rekap Data Sinyal Jantung.
- Memasukkan ID atau nama *file* pasien yang pernah disimpan sebelumnya.
- Jika data *entry* yang dimasukkan benar maka akan menampilkan grafik sinyal jantung yang sesuai dengan ID atau nama *file* dari pasien.

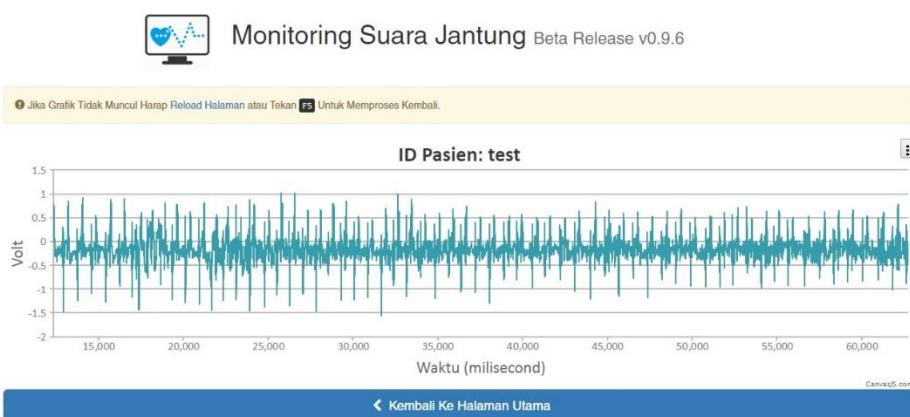
4.2.2 Hasil Pengujian

Untuk mengakses rekap data pasien yang akan ditampilkan grafik sinyal jantungnya dari pilihan menu pada tampilan *website* maka akan ada

konfirmasi *entry* nama *file* atau ID dari pasien yang telah disimpan sebagaimana tampak pada Gambar 4.4, sedangkan pada Gambar 4.5 menampilkan grafik sinyal jantung dari rekap pasien yang telah dipilih.



Gambar 4.4 Tampilan konfirmasi *entry* rekap data pasien



Gambar 4.5 Hasil tampilan grafik sinyal jantung pasien terpilih

BAB 5

Evaluasi Kinerja Sistem IMedT Monitoring Suara Jantung

Pengujian diperlukan untuk mengetahui performansi sistem IMedT monitoring Cardiovaskuler secara keseluruhan mulai dari proses data *acquisition*, transmisi auskultasi hingga komparasi bentuk tampilan grafik sinyal suara jantung melalui perbedaan perubahan konfigurasinya.

Pada bab 5 ini dilakukan analisa hasil pengujian sistem IMedT secara menyeluruh berdasarkan hasil perubahan data bentuk tampilan sinyal suara jantung mulai dari sensor PCG, komputer pembanding sampai dengan bentuk tampilan sinyal suara jantung hasil akses dari Raspberry Pi 2 melalui perubahan beberapa parameter jaringan komputer, diantaranya: perubahan *baudrate*, *data loss* dan juga *delay* yang terjadi ketika proses pentransmision data sinyal suara detak jantung dari Arduino ke Raspberry Pi 2.

5.1 Prosedur Pengujian

Langkah-langkah dibawah diperlukan untuk melakukan proses pengujian sistem IMedT monitoring suara jantung, diantaranya:

- Lokasi pengambilan data pasien idealnya dalam situasi dan kondisi yang tenang.
- Mengkoneksikan Raspberry Pi 2 dan PC pembanding ke *router* menggunakan kabel UTP.
- Meletakkan sensor suara detak jantung pada bagian jantung pasien secara tepat.
- Menghubungkan Arduino Uno dan PC pembanding dengan menggunakan koneksi kabel USB TTL.

- Menghubungkan Arduino Uno dan Raspberry Pi 2 dengan menggunakan kabel USB.
- Membuka aplikasi Terminal pada Raspberry Pi 2, kemudian mengetik perintah “*ifconfig*”, mencatat IP Address-nya dan berikutnya mengetik perintah “*node-red-start*”.
- Membuka aplikasi *RS232 Data Logger* dari komputer, meng-klik *start logging*, membuka aplikasi *browser* Google Chrome pada komputer, kemudian memasukkan IP Address Raspberry Pi 2 yang sudah dicatat.
- Lalu meng-klik pilihan tombol menu *Live Data Sinyal Jantung* pada tampilan *website* utama.
- Selanjutnya melakukan pengambilan data sinyal suara detak jantung selama 1 menit 20 detik (120 detik).
- Mengamati data dan bentuk grafik, apakah sinyal yang ditangkap sudah benar sinyal jantung.
- Setelah waktu pengambilan data sudah tercapai 120 detik, maka kabel USB Arduino, kemudian dilakukan penyimpanan data serta menghentikan proses pengambilan data pada *software RS232 Data Logger* melalui pilihan tombol menu “*Stop Logging*”.
- Mengumpulkan data sinyal suara detak jantung dari *Raspberry Pi 2* dan PC pembanding yang telah didapat kedalam satu *folder* di komputer agar dapat dianalisa.
- Meng-*Copy* data yang tersimpan pada *file* ke dalam *file* Excel untuk melihat jumlah data dan waktu yang dikirimkan oleh Arduino, diterima oleh *Raspberry Pi 2* dan PC Pembanding.
- Melakukan *sorting* data yang berasal dari aplikasi *RS232 Data Logger* dan data yang diperoleh *Raspberry Pi2* untuk di analisa.

5.2 Pengujian Sistem IMedT Monitoring Secara Keseluruhan

Pada penelitian transmisi sinyal suara detak jantung dilakukan beberapa kali variasi percobaan perubahan nilai *baudrate* serta nilai *delay* dalam rangka agar hasil dari proses analisa pengujian ini akan diketahui berapa besar nilai

baudrate yang diperlukan supaya proses transmisi mendapatkan hasil terbaik, berapa lama nilai *delay* yang terjadi pada saat proses transmisi serta berapa persenkah jumlah data yang hilang (*data loss*) saat pengiriman sinyal suara detak jantung berlangsung. Apabila ketiga parameter uji tersebut diatas diketahui maka bisa dicarikan kombinasi yang pas diantara ketiga parameter tersebut sehingga dapat menghasilkan kinerja sistem yang terbaik.

5.2.1 Pengiriman Data Menggunakan Variasi *Baudrate*

- A. Percobaan 1 menggunakan *baudrate* 19200, dimana percobaan dilakukan dalam waktu 1 menit 20 detik.



Gambar 5.1 Percobaan 1 pada Raspberry Pi 2 dengan *baudrate* 19200



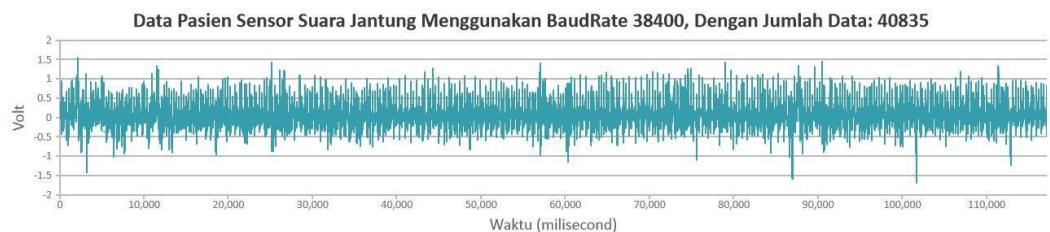
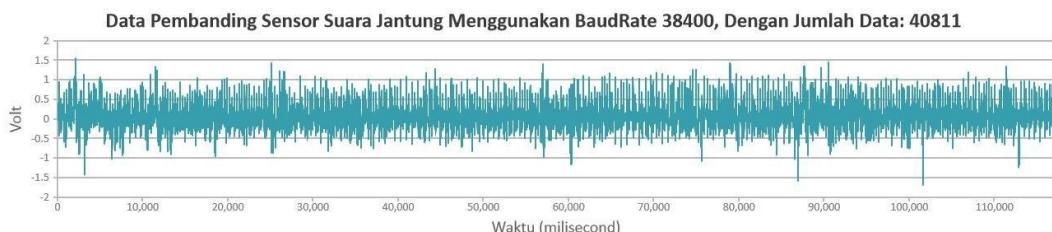
Gambar 5.2 Percobaan 1 pada PC pembanding dengan *baudrate* 19200

Berdasarkan keterangan dari Gambar 5.1 dan Gambar 5.2 maka didapatkan data sebagaimana terlihat pada Tabel 5.1 dibawah ini.

Tabel 5.1 Hasil percobaan 1 dengan *baudrate* 19200

Penerima	Start	Stop	Jumlah Data
Raspberry Pi 2	3,294 detik	122,015 detik	20553
PC Pembanding	3,294 detik	121,797 detik	20518

B. Percobaan 2 dengan *baudrate* 38400 yang mana percobaan dilakukan dalam kurun waktu 1 menit 20 detik.

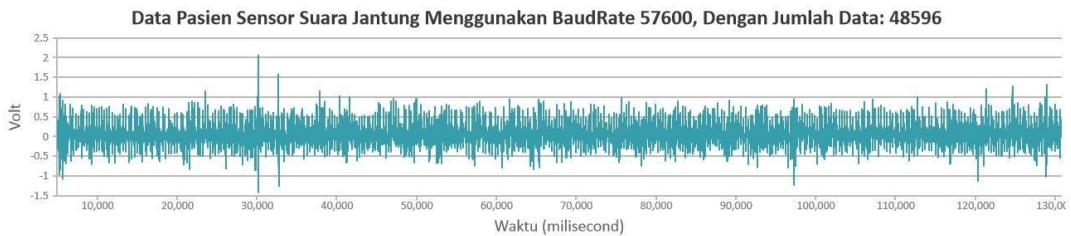
Gambar 5.3 Percobaan 2 pada Raspberry Pi 2 dengan *baudrate* 38400Gambar 5.4 Percobaan 2 pada PC pembanding dengan *baudrate* 38400

Berdasarkan Gambar 5.3 dan Gambar 5.4 dengan *baudrate* 38400 pada Raspberry Pi dan PC pembanding didapatkan data seperti pada Tabel 5.2.

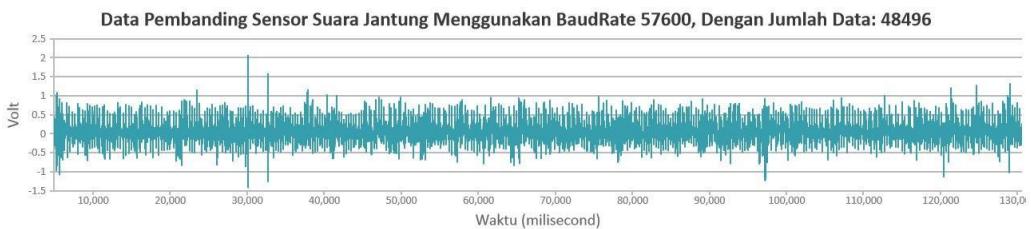
Tabel 5.2 Tabel hasil percobaan 2 dengan *baudrate* 38400

Penerima	Start	Stop	Jumlah Data
Raspberry Pi 2	0 detik	117,142 detik	40835
PC Pembanding	0 detik	117,066 detik	40811

- C. Percobaan 3 dengan *baudrate* 57600, dilakukan perekaman data sinyal suara jantung selama kurun waktu 1 menit 20 detik.



Gambar 5.5 Percobaan 3 pada Raspberry Pi 2 dengan *baudrate* 57600



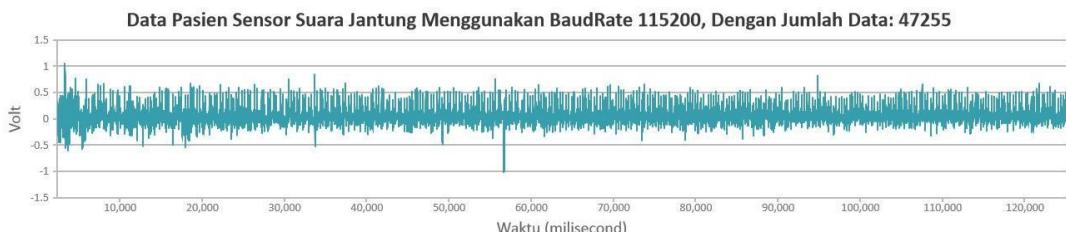
Gambar 5.6 Percobaan 3 pada PC pembanding dengan *baudrate* 57600

Melihat gambar grafik sinyal jantung dari Gambar 5.5 dan Gambar 5.6 didapatkan hasil jumlah data seperti Tabel 5.3.

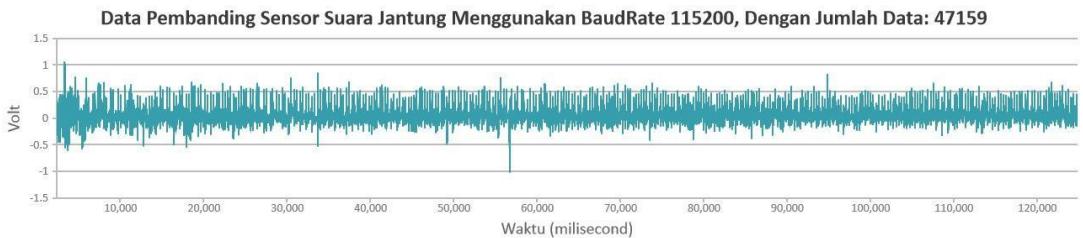
Tabel 5.3 Tabel hasil percobaan 3 dengan *baudrate* 57600

Penerima	Start	Stop	Jumlah Data
Raspberry Pi 2	4,895 detik	130,788 detik	48596
PC Pembanding	4,895 detik	130,252 detik	48496

- D. Percobaan 4 menggunakan *Baudrate* 115200, percobaan dilakukan selama waktu 1 menit 20 detik.



Gambar 5.7 Percobaan 4 pada Raspberry Pi 2 dengan *baudrate* 115200



Gambar 5.8 Percobaan 4 pada PC pembanding dengan *baudrate* 115200

Berdasarkan Gambar 5.7 dan Gambar 5.8 didapatkan hasil data yang terdapat pada Tabel 5.4.

Tabel 5.4 Tabel hasil percobaan 4 dengan *baudrate* 115200

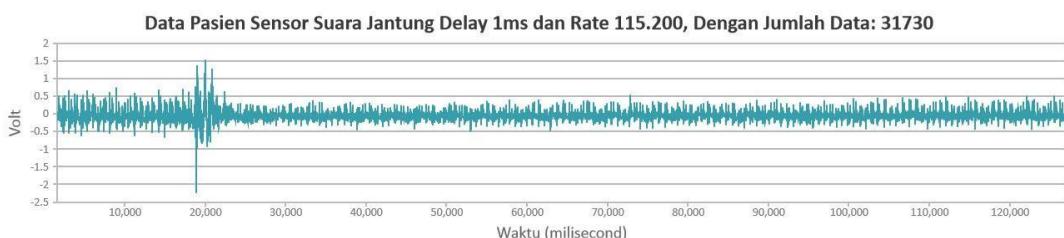
Penerima	Start	Stop	Jumlah Data
Raspberry Pi 2	2,330 detik	125,073 detik	47255
PC Pembanding	2,330 detik	124,819 detik	47159

Kesimpulan pada percobaan transmisi sinyal suara detak jantung dengan menggunakan perubahan *baudrate* 19200 sampai dengan 115200 tidak ditemukannya *packet loss* atau *delay*, karena menggunakan koneksi kabel serial USB jarak pendek antara Arduino Uno dan Raspberry Pi 2.

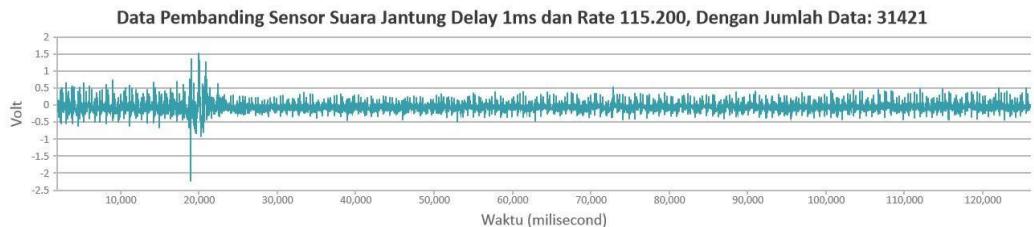
5.2.2 Pengiriman Data Menggunakan *Baudrate* 115200 dan Variasi *Delay*

A. Percobaan 5 dengan *delay* 1 milidetik

Percobaan dilakukan dengan selang waktu selama 1 menit 20 detik, dengan setting *delay* pengiriman data dilakukan setiap 1 milidetik.



Gambar 5.9 Raspberry Pi 2 dengan *baudrate* 115200 dan *delay* 1 milidetik



Gambar 5.10 PC pembanding dengan *baudrate* 115200 dan *delay* 1 milidetik

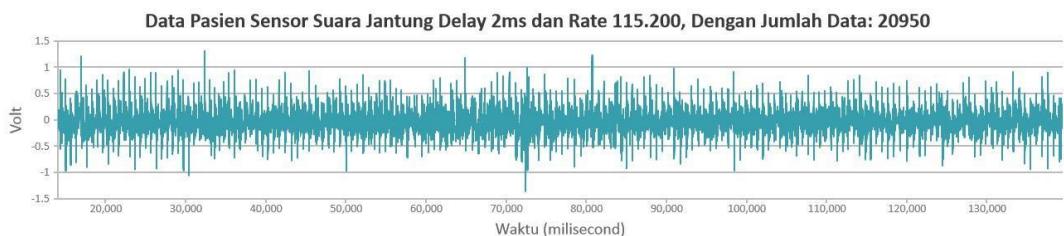
Berdasarkan Gambar 5.9 dan Gambar 5.10 didapatkan data sebagaimana yang ada pada Tabel 5.5.

Tabel 5.5 Tabel hasil Percobaan 5 dengan *baudrate* 115200 dan *delay* 1 ms

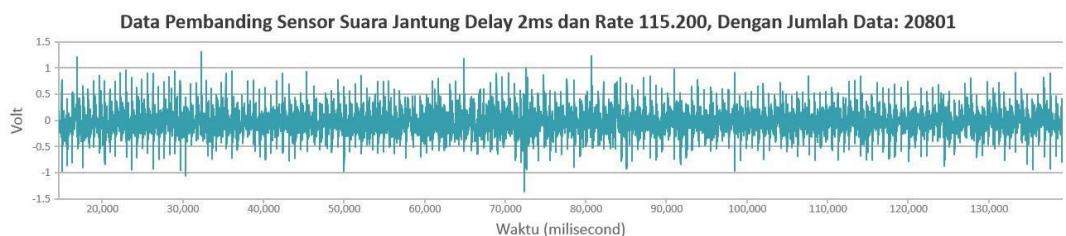
Penerima	Start	Stop	Jumlah Data
Raspberry Pi 2	1,551 detik	127,019 detik	31730
PC Pembanding	1,881 detik	126,105 detik	31421

B. Percobaan 6 dengan *delay* 2 milidetik

Percobaan keenam dilakukan dengan waktu 1 menit 20 detik sedangkan *delay* pengiriman datanya dilakukan setiap 2 milidetik.



Gambar 5.11 Raspberry Pi 2 dengan *baudrate* 115200 dan *delay* 2 milidetik



Gambar 5.12 PC pembanding dengan *baudrate* 115200 dan *delay* 2 milidetik

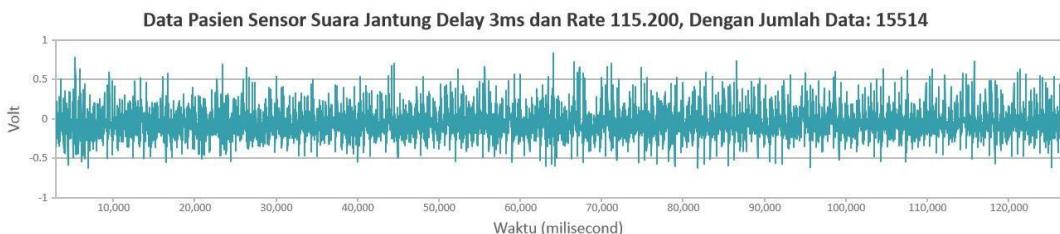
Melihat pada Gambar 5.11 dan Gambar 5.12 didapatkan data seperti Tabel 5.6 dibawah ini.

Tabel 5.6 Tabel hasil percobaan 6 dengan *baudrate* 115200 dan *delay* 2 ms

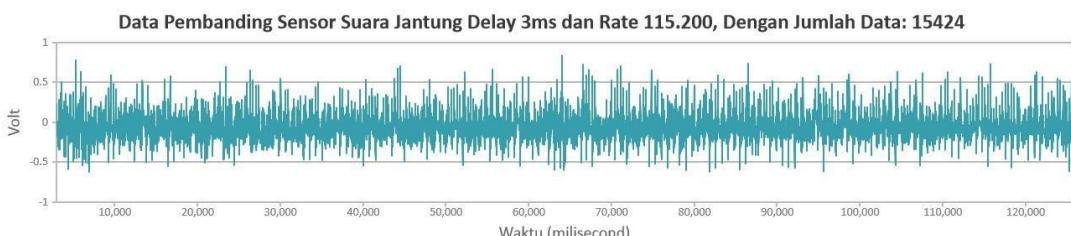
Penerima	Start	Stop	Jumlah Data
Raspberry Pi 2	14,001 detik	139,416 detik	20950
PC Pembanding	14,587 detik	139,111 detik	20801

C. Percobaan 7 dengan *delay* 3 milidetik

Percobaan dilakukan dalam waktu 1 menit 20 detik dengan *delay* pengiriman data dilakukan setiap 3 milidetik.



Gambar 5.13 Raspberry Pi 2 dengan *baudrate* 115200 dan *delay* 3 milidetik



Gambar 5.14 PC pembanding dengan *baudrate* 115200 dan *delay* 3 milidetik

Berdasarkan pada Gambar 5.13 dan Gambar 5.14 didapatkan sejumlah data sebagaimana seperti Tabel 5.7 berikut ini.

Tabel 5.7 Tabel hasil percobaan 7 dengan *baudrate* 115200 dan *delay* 3 ms

Penerima	Start	Stop	Jumlah Data
Raspberry Pi 2	2,904 detik	126,349 detik	15514
PC Pembanding	2,941 detik	125,660 detik	15424

BAB 6

Analisa Hasil Kinerja Sistem IMedT Monitoring Suara Jantung

Adanya perubahan konfigurasi nilai *baudrate* dan *delay* pada proses transmisi data sinyal detak suara jantung berpengaruh pada nilai *throughput*, utilisasi penggunaan *bandwidth* dan jumlah *packet loss*.

Setelah melakukan percobaan dengan beberapa variasi perubahan besaran *baudrate* dan variasi *delay* yang diamati dari aplikasi *website* maka dapat disimpulkan bahwa pada pengujian berdasarkan variasi *baudrate*, yaitu: semakin tinggi atau besar nilai *baudrate* pengiriman data maka *throughput* akan semakin besar serta persentase utilisasi *bandwidth* akan semakin kecil serta tidak ditemukannya *delay* pengiriman maupun *packet loss* yang terjadi saat pengiriman data. Pada pengujian perubahan variasi *delay* yang terjadi adalah semakin besar nilai *delay* pada program yang digunakan maka akan mempengaruhi nilai *throughput* dan persentase utilisasi *bandwidth* akan semakin kecil sebagaimana terlihat pada Tabel 6.1 dan Tabel 6.2.

6.1 Transmisi Sinyal dengan Baudrate 19200

Berdasarkan pada Tabel 6.2, pentransmisian sinyal suara detak jantung dengan *baudrate* 19200 menggunakan aplikasi *web* secara *live* dalam waktu 1 menit 20 detik menghasilkan rata – rata *throughput* sebesar 15777,50 bps dan persentase utilisasi *bandwidth* sebesar 82,174 % untuk Raspberry Pi 2 dan pada PC pembanding rata – rata *throughput* sebesar 15527,50 bps dan persentase utilisasi *bandwidth* persentase sebesar 80,872 %.

Tabel 6.1 Perhitungan *throughput* dan utilisasi *bandwidth* dengan *baudrate* 115200 pada Raspberry Pi 2

Baudrate 115200			
Perhitungan Frame Per Bagian		Throughput (bps)	Utilisasi Bandwidth (%)
■ Bagian 1 = 589120 bit per 17 detik		34654,1	30,082
■ Bagian 2 = 695160 bit per 20 detik		34758	30,172
■ Bagian 3 = 695070 bit per 20 detik		34753,5	30,168
■ Bagian 4 = 694980 bit per 20 detik		34749	30,164
■ Bagian 5 = 695080 bit per 20 detik		34754	30,168
■ Bagian 6 = 756300 bit per 20 detik		37815	32,826
■ Bagian 7 = 191900 bit per 5 detik		38380	33,316
Rata - rata		35694,80	30,985

Tabel 6.2 Tabel hasil perhitungan rata - rata *throughput* dan utilisasi *bandwidth* pada pengujian variasi *baudrate*

Perangkat <i>Baudrate</i>	Raspberry Pi 2		PC Pembanding	
	<i>Throughput</i>	Utilisasi <i>Bandwidth</i>	<i>Throughput</i>	Utilisasi <i>Bandwidth</i>
19200 bps	15777,50 bps	82,174 %	15527,50 bps	80,872 %
38400 bps	31531,86 bps	82,114 %	31508,33 bps	82,053 %
57600 bps	36077,90 bps	62,635 %	35935,05 bps	62,387 %
115200 bps	35694,80 bps	30,985 %	35420,52 bps	30,747 %

6.2 Transmisi Sinyal dengan Baudrate 38400

Pentransmisian sinyal suara detak jantung dengan *baudrate* 38400 menggunakan aplikasi *web* secara *live* dalam waktu 1 menit 20 detik menghasilkan rata – rata *throughput* sebesar 31531,86 bps dan persentase utilisasi *bandwidth* sebesar 82,114 % untuk Raspberry Pi 2 dan pada PC pembanding rata – rata *throughput* sebesar 31508,33 bps dan persentase utilisasi *bandwidth* sebesar 80,053 %.

6.3 Transmisi Sinyal dengan Baudrate 57600

Pada pentransmisian sinyal suara detak jantung dengan *baudrate* 57600 menggunakan aplikasi *web* secara *live* dalam waktu 1 menit 20 detik menghasilkan rata – rata *throughput* sebesar 36077,90 bps dan persentase utilisasi *bandwidth* sebesar 62,635 % untuk Raspberry Pi 2 dan pada PC pembanding rata – rata *throughput* sebesar 35935,05 bps dan persentase utilisasi *bandwidth* sebesar 62,387 %.

6.4 Transmisi Sinyal dengan Baudrate 115200

Sedangkan pentransmisian sinyal suara detak jantung dengan *baudrate* 115200 menggunakan aplikasi *web* secara *live* dalam waktu 1 menit 20 detik menghasilkan rata – rata *throughput* sebesar 35694,80 bps dan persentase utilisasi *bandwidth* sebesar 30,985 % untuk Raspberry Pi 2 dan pada PC pembanding rata – rata *throughput* sebesar 35420,52 bps dan persentase utilisasi *bandwidth* sebesar 30,747 %.

Tabel 6.3 Tabel hasil perhitungan rata - rata *throughput* dan utilisasi *bandwidth* pada pengujian variasi *delay*

Perangkat <i>Delay</i>	Raspberry Pi 2		PC Pembanding	
	Rata-rata <i>Throughput</i>	Utilisasi <i>Bandwidth</i>	Rata-rata <i>Throughput</i>	Utilisasi <i>Bandwidth</i>
1 milidetik	23295,44 bps	20,222 %	23292,25 bps	20,219 %
2 milidetik	15567,33 bps	13,513 %	15317,59 bps	13,297 %
3 milidetik	11680,28 bps	10,139 %	11474,54 bps	9,961 %

Pembacaan data pada Tabel 6.3, pentransmisian sinyal suara detak jantung dengan *baudrate* 115200 dan *delay* 1 milidetik menggunakan aplikasi *web* secara *live* dalam waktu 1 menit 20 detik menghasilkan rata – rata *throughput* sebesar 23295,44 bps dan persentase utilisasi *bandwidth* sebesar 20,222 %

untuk Raspberry Pi 2 dan pada PC pembanding rata – rata *throughput* sebesar 15567,33 bps dan persentase utilisasi *bandwidth* sebesar 20,219 %.

Pentransmisian sinyal suara detak jantung dengan *baudrate* 115200 dan *delay* 2 milidetik menggunakan aplikasi *web* secara *live* dalam waktu 1 menit 20 detik menghasilkan rata – rata *throughput* sebesar 15567,33 bps dan persentase utilisasi *bandwidth* sebesar 13,513 % untuk Raspberry Pi 2 dan pada PC pembanding rata – rata *throughput* sebesar 15317,59 bps dan persentase utilisasi *bandwidth* sebesar 13,297 %.

Pada pentransmisian sinyal suara detak jantung dengan *baudrate* 115200 dan *delay* 3 milidetik menggunakan aplikasi *web* secara *live* dalam waktu 1 menit 20 detik menghasilkan rata–rata *throughput* sebesar 11680,28 bps dan persentase utilisasi *bandwidth* sebesar 10,139 % untuk Raspberry Pi 2 dan pada PC pembanding rata – rata *throughput* sebesar 11474,54 bps dan persentase utilisasi *bandwidth* sebesar 9,961 %.

Pada semua pengujian IMedT sistem monitoring detak jantung dalam penelitian ini menggunakan sensor suara detak jantung atau *heart sound sensor* PCG yang memiliki spesifikasi *range* frekuensi antara 1 sampai dengan 600 Hz saja, sehingga mempunyai keterbatasan dalam melakukan *delay* program terkait dengan hal tersebut serta berdasarkan pengalaman percobaan dan pengujian maka disarankan pemilihan *delay* yang 2 milidetik untuk menghasilkan hasil yang terbaik.

BAB 7

Kesimpulan Dan Saran Sistem IMedT Monitoring Suara Jantung

7.1 Kesimpulan

Berdasarkan seluruh hasil analisa dari transmisi sinyal suara detak jantung dari Arduino Uno menuju Raspberry Pi 2 dan PC pembanding maka dapat diambil beberapa kesimpulan sebagai berikut.

1. Aplikasi *web* yang sudah dibuat dapat dimanfaatkan dalam memonitoring kondisi jantung secara *real time*.
2. Berdasarkan hasil penelitian, hasil transmisi sinyal suara detak jantung menunjukkan bahwa semakin tinggi *baudrate* dalam pengiriman data yang digunakan, maka akan semakin kecil utilisasi *bandwidth* pada sistem, dan juga semakin tinggi *delay* pengiriman pada program yang digunakan maka utilisasi *bandwidth* juga semakin kecil.
3. Semakin tinggi *baudrate* yang digunakan maka semakin cepat pengiriman data dari sensor ke pengolah data, dalam penelitian ini digunakan Raspberry Pi 2 sebagai pengolah data yang didapat dari sensor.
4. Untuk mendapatkan data yang baik dan benar tanpa kehilangan beberapa data yang diambil oleh sensor suara detak jantung digunakan dengan *baudrate* yang paling tinggi, dan menggunakan *delay* pengiriman sebesar 2 milidetik karena pada spesifikasi *hardware* sensor suara detak jantung yang digunakan hanya dapat menangkap dan memproses dengan jarak frekuensi 1 hingga 600 Hz.

7.2 Saran

Dari kesimpulan yang telah dibuat, maka agar transmisi sinyal suara detak jantung berjalan dengan lebih baik, maka hal yang perlu dipertimbangkan adalah:

1. Sistem ini dapat diterapkan untuk pengukuran dan analisis medis apabila terhubung ke internet.
2. Menggunakan *hardware* sensor suara detak jantung yang lebih baik dalam pengambilan data agar dapat mendapatkan data yang lebih detail untuk dilakukan bahan analisis kedepannya.
3. Memerlukan perlindungan data dengan cara melakukan enkripsi data maupun perlindungan secara sistem, karena data diletakkan pada *server* yang terhubung langsung dalam *internet* jadi privasi seseorang menjadi rentan untuk dilihat oleh orang yang tidak bertanggung-jawab.

DAFTAR PUSTAKA

- Anonim. 2015. *Bagaimana Cara Kerja Jantung Pada Tubuh Manusia?*, [online], (<http://sehatjantungku.com/bagaimana-cara-kerja-jantung-pada-tubuh-manusia/bagaimana-cara-kerja-jantung-normal/>). diakses pada tanggal 11 April 2016).
- Arduino.cc. 2013. *Arduino Program Language Reference*, [online], (<http://arduino.cc/en/Reference/HomePage> , diakses tanggal 17 Mei 2016).
- Arduino.cc. 2013. *Arduino Board Uno SMD*, [online], (<http://arduino.cc/en/Main/arduinoBoardUno>, diakses tanggal 17 Mei 2016).
- Banzi, M. 2009. *Getting Started with Arduino*. America: O'reilly.
- Burrus, S., Gopinath, A., Guo, H. 1998. *Introduction to Wavelets and Wavelets Transform A Primer*. New Jersey: Prentice-Hall.
- Daso, Fransiscus. 2015. *Analisis Sinyal Suara Jantung Dengan Menggunakan Representasi SpektrumI*. STIKOM Surabaya.
- Djuandi, F. 2011. *Pengenalan Arduino*. Banten: www.tokobuku.com.
- Eben, U. 2015. *Raspberry Pi 2 on sale now at \$35*, [online], (<https://www.raspberrypi.org/blog/raspberry-pi-2-on-sale/>, diakses tanggal 8 Maret 2016).
- Ferdoush, Li. 2014. *Wireless Sensor Network System Design using Raspberry Pi and Arduino for Environmental Monitoring Applications*. ScienceDirect.
- Hakim. 2013. *Pemanfaatan Mini Pc Raspberry PI Sebagai Pengontrol Lampu Jarak Jauh Berbasis Web Pada Rumah*. Bandung: Universitas Komputer Indonesia (UNIKOM).

- IDCloudHost.com. 2015. *Pengertian Web Server dan Fungsinya*, [online], (<https://idcloudhost.com/pengertian-web-server-dan-fungsinya/>, diakses tanggal 18 Mei 2016).
- Jusak, J., Pratikno, H., Putra, V.H. 2016. "Internet of Medical Things for Cardiac Monitoring: Paving the way to 5G Mobile Networks," *IEEE Conference on Communication, Networks and Satellite 2016*, Surabaya.
- Jusak dan Puspasari, I. 2015. "Wireless tele-ausculation for phonocardiograph signal recording through the zigbee networks," *IEEE Asia Pasific Conference on Wireless and Mobile (APWiMob)*, Bandung, Indonesia, August 27-19, 2015.
- Lynn, P. A. 1994. *Introductory Digital Signal Processing With Computer Application*. New York: John Wiley & Sons.
- Puspasari, Ira. 2013. *Study Analisis Metode Ekstraksi Ciri pada Sinyal Suara Jantung Diastolik*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Oktarina, E.S., Jusak, Puspasari, I. 2015. *Transmisi Nirkabel Sinyal Auskultasi Suara Jantung Dengan menggunakan Wireless Zigbee Network*. Surabaya: Stikom Surabaya.
- Sa-ngasoongsong,A.,Kunthong,J.,Sarangan,V.,Cai,X.,andBukkapatnam,S.T.22. "A low-cost, portable, high-throughput wireless sensor system for phonocardiography applications," *Journal of Sensors*, Volume 12, 2012, pp. 10851-10870
- Seeedstudio.com. 2011. *Heart Sound Sensor*, [online], (http://www.seeedstudio.com/wiki/Heart-Sound_sensor, diakses tanggal 18 Mei 2016).

- Setiaji, D., 2011, *Rekayasa Stetoskop Elektronik Dengan Kemampuan Analisis Bunyi Jantung*, Seminar Nasional Teknologi Informasi & Komunikasi Terapan. ISBN 979-26-0255-0, Salatiga: Prodi Teknik Elektro, UKSW.
- Vinay R., Hamid S. 2015. *Embedded Wireless Sensor Network for Environment Monitoring*, Journal of Advances in Computer Networks, Vol. 3 No. 1, March 2015, Bangalore, India.
- WHO, 2014. “Global Status Report on Noncommunicable disease (NCD) 2014,” Switzerland.
- WHO, 2008. “Health in Asia and Pacific, World Health Organization-Western Pacific & South-East Asia,” New Delhi, India.

INDEKS

A

Arduino Uno, i, iii, v, viii, 3, 9, 10, 11, 17, 19, 21, 23, 26, 32, 39, 43, 45, 46, 50, 57
atrium, 3
auskultasi, i, v, 1, 2, 22, 29, 42, 45

B

bilik, 3

D

data rate, 6
Delay, iv, 14, 50, 55
diastole, 3

H

heart sound sensor, v, 17, 18, 19, 22, 56

I

IMedT, i, iii, iv, v, vi, 1, 6, 7, 15, 16, 17, 32, 33, 39, 40, 45, 46, 53, 56, 57
Internet of Medical Things, i, iii, 1, 6, 7, 60
Internet of Things, 1, 5, 6
IoT, iii, 1, 5, 6, 7

J

Jantung, ii, iii, iv, 1, 3, 15, 19, 21, 29, 30, 32, 39, 40, 43, 45, 46, 53, 57, 59, 60, 61

Jitter, 14

L

latency, 6, 14

M

Mesin ke Mesin, 5
monitoring, i, v, 2, 7, 15, 16, 17, 19, 21, 32, 40, 45, 56

P

Packet Loss, viii, 13, 14
phonocardiograph, i, 15, 60

portable, 2, 3, 15, 60
prototype, i, 2
purwarupa, i, 2

Q

QoS. *See Quality of Service*
Quality of Service, 13

R

Raspberry Pi 2, i, iii, iv, v, vi, vii, viii, 2, 7, 8, 17, 19, 20, 21, 24, 25, 26, 27, 28, 30, 33, 35, 36, 37, 39, 41, 42, 43, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57
Raspbian, iii, vi, 33, 34

S

sampling, 23
semilunar, 3
serambi, 3
System on Chip, 8
systole, 3

T

Throughput, 13, 54, 55

V

ventricle, 3
Virtual Network Index, 5

W

Wavelet, iii, v, 5
WHO, i, 1, 61
Wireless, 6, 59, 60, 61
World Health Organization, 1, 61

Z

Zettabyte, 5
Zigbee, 2, 6, 60

Tentang Penulis



Dr. Jusak saat ini bekerja sebagai staff pengajar di Institut Bisnis dan Informatika STIKOM Surabaya, menyelesaikan pendidikan sarjana Teknik Elektro dari Universitas Brawijaya, Malang dan program Doktor dalam bidang Teknik Elektro dan Komunikasi dari Universitas RMIT, Melbourne, Australia. Penulis juga aktif sebagai konsultan dan peneliti dalam bidang rekayasa trafik, telekomunikasi dan jaringan komputer di berbagai tempat, salah satunya adalah sebagai peneliti dalam proyek pembangunan *Next Generation Network* bersama Telecom New Zealand pada tahun 2009-2011. Beberapa buku pernah ditulis antara lain: *Kreasi Situs Mobile Internet dengan xHTML MP* (Jakarta: Prestasi Pustaka, 2007), *Teknologi Komunikasi Data Modern* (Yogyakarta: Penerbit ANDI, 2013)



Heri Pratikno, M.T. , adalah dosen di prodi S1 Sistem Komputer, Institut Bisnis dan Informatika Stikom Surabaya, saat ini sedang studi lanjut S3 pada fakultas Electric & Electrical Engineering di Universiti Malaysia Pahang (UMP). Dalam bidang jaringan komputer berbasis MiKrotik mempunyai sertifikat MTCNA (*MikroTik Certified Network Associate*, 2015), MTCRE (*MikroTik Certified Routing Engineer*, 2016), Academy Trainer (2016) serta menjadi instruktur di Cisco Networking Academy (*Introduction-to-Network, Routing and Switching Essentials*). Buku populer yang pernah ditulis: *Proteksi Secara Hardware: Komputer Bebas Virus 100%* (Jakarta, Prestasi Pustaka Publisher, 2008) dan *Cara Mudah Wujudkan Komputer Bebas Virus* (Jakarta, Cerdas Pustaka, 2009).