

# Proyek Instrumentasi Medis Berbasis Internet of Things


**Anton Yudhana, Hendril Satrian Purnama, dkk**


Buku ini disusun sebagai bahan ajar untuk mahasiswa strata 1 maupun diploma untuk jurusan teknik elektro, teknik informatika, teknik biomedis, dll. di sisi lain buku ini juga dapat digunakan sebagai referensi untuk para praktisi yang berkerja di bidang yang berkaitan dengan instrumentasi medis, perancangan aplikasi dan internet of things.

buku ini disusun menjadi 9 bab, yang mana bab 1 terdiri dari pendahuluan yang menjelaskan latar belakang penulisan buku, kemudian bab 2-5 menyajikan dasar-dasar yang memberikan pengetahuan secara singkat tentang software dan hardware yang dibutuhkan untuk perancangan proyek instrumentasi medis, bab 6-9 menyajikan uraian tentang beberapa proyek instrumentasi medis yang telah dikerjakan oleh penulis. Proyek instrumentasi medis yang dikerjakan pada buku ini antara lain:

- Implementasi Sensor Heart Rate untuk mendeteksi detak jantung secara real-time melalui interface komputer
- Aplikasi Mobile Phone Untuk Monitoring Tingkat Kebugaran Seseorang Dengan Algoritma Fast Fourier Transform
- Rancang Bangun Sistem Pemantauan Infus Berbasis Android
- Aplikasi Pemantau Terapi Jari Pasca Stroke Menggunakan Sarung Tangan Pintar Secara Real Time

 083867708263

 cv.mine7

 mine mine

  
Penerbit : cv. Mine  
Perum Sidorejo Bumi Indah F 153  
Rt 11 Ngestiharjo Kasihan Bantul  
Mobile : 083867708263  
email : cv.mine.7@gmail.com



Proyek Instrumentasi Medis Berbasis Internet of Things

Anton Yudhana, dkk

ISBN 978-623-7550-17-4



# Proyek INSTRUMENTASI MEDIS berbasis Internet of Things



**Anton Yudhana,  
Hendril Satrian Purnama, dkk**

# **Proyek Instrumentasi Medis Berbasis Internet of Things**

**Anton Yudhana, Hendril Satrian Purnama, dkk**



# **Proyek Instrumentasi Medis Berbasis Internet of Things**

## **Penulis**

Anton Yudhana

Hendril Satrian Purnama

Ismail Aprianto

Kaspul Anwar

Marta Dwi Darma Putra

Dawam Alfaqih

## **Desain Sampul & Layout**

Hendril Satrian Purnama

Hak Cipta © 2019, pada penulis

Hak publikasi pada Penerbit CV Mine

*Dilarang memperbanyak, memperbanyak sebagian atau seluruh isi dari buku ini dalam bentuk apapun, tanpa izin tertulis dari penerbit.*

**© HAK CIPTA DILINDUNGI OLEH UNDANG-UNDANG**

Cetakan ke-1 November Tahun 2019

CV Mine

Perum SBI F153 Rt 11 Ngestiharjo, Kasihan, Bantul,

Yogyakarta-55182 Telp: 083867708263

Email: [cv.mine.7@gmail.com](mailto:cv.mine.7@gmail.com)

ISBN : 978-623-7550-17-4

## Kata Pengantar

Puji syukur kita panjatkan kehadirat Allah SWT atas selesainya buku **“Proyek instrumentasi Medis berbasis Internet of Things”**. Buku ini dirancang sebagai buku teks dan referensi untuk mahasiswa Strata 1 ataupun Diploma dalam bidang Teknik Elektro, Teknik Informatika Fisika, Teknik Fisika, maupun Teknik Biomedis.

Buku ini disusun berdasarkan hasil penelitian dan referensi dari beberapa buku yang berkaitan dengan proyek instrumentasi medis dan penerapan *internet of things*. Penelitian dikti yang didanai dari Ristekdikti pada tahun 2019 dengan judul : *“Heart Beat Signal dan Parent-Cognitive Behavioral Therapy: Deteksi dan Intervensi Regulasi Emosi Orang Tua Pelaku Tindak Kekerasan pada Anak”* dan *“Sinyal EEG dan Cognitive-load: Analisis Persepsi Psikologis dalam User Experience Intelligence User Interface”* menjadi rujukan untuk terbitnya buku ini. Buku ini terbagi menjadi 9 Bab yang masing-masing menjelaskan bagian-bagian dasar dan contoh-contoh proyek di bidang elektromedis.

Kami menyadari bahwa masih terdapat banyak kekurangan dalam penyusunan buku ini. Kami sangat mengharapkan kritik dan saran dari para pembaca untuk penyempurnaan buku ini. Pada akhirnya penulis berharap semoga buku ini dapat memberi manfaat yang luas bagi para pembaca baik dari kalangan mahasiswa maupun para praktisi yang bekerja pada bidang yang sesuai.

Yogyakarta, November 2019

**Tim Penulis**

# Sistematika Penulisan Buku

## **Bab 1: Pendahuluan**

- Berisi latar belakang dan pengantar terkait pembuatan proyek yang akan dibahas dalam buku ini secara umum

## **Bab 2: Sensor-sensor untuk aplikasi instrumentasi medis**

- Berisi ulasan tentang jenis-jenis sensor yang digunakan dalam proyek aplikasi instrumentasi biomedis

## **Bab 3: Pengenalan Arduino**

- Berisi pengenalan dasar Arduino, baik terkait *Hardware* maupun *software*-nya

## **Bab 4: Pengenalan Software Processing**

- Berisi pengenalan dasar dan cara penggunaan *software* Processing untuk berbagai macam aplikasi

## **Bab 5: Mengkoneksikan Arduino dengan Processing**

- Berisi tutorial terkait cara mengkoneksikan Arduino IDE dengan Processing IDE untuk menampilkan data grafis dari hasil pembacaan sensor oleh Arduino

## **BAB 6: Proyek visualisasi detak jantung dengan menggunakan interface komputer berbasis Arduino dan Processing**

- Berisi uraian lengkap terkait proses pembuatan proyek dan hasil dari penelitian yang telah dilakukan

### **BAB 7: Aplikasi Mobile Phone untuk Monitoring Tingkat Kebugaran Seseorang dengan Algoritma Fast Fourier Transform**

- Berisi uraian tentang proses perancangan dan hasil percobaan terkait proyek Aplikasi *Mobile Phone* untuk Monitoring Tingkat Kebugaran

### **BAB 8: Rancang Bangun Sistem Pemantauan Infus berbasis Android**

- Berisi uraian tentang proses perancangan dan hasil percobaan terkait proyek sistem pemantauan infus berbasis android

### **BAB 9: Aplikasi Pemantau Terapi Jari Pasca Stroke Menggunakan Sarung Tangan Pintar Secara Real Time**

- Berisi uraian tentang proses perancangan dan hasil percobaan dari proyek Pemantau Terapi Jari Pasca *Stroke* Menggunakan Sarung Tangan Pintar Secara *Real Time*

# Daftar Isi

|                                                                                                   |           |
|---------------------------------------------------------------------------------------------------|-----------|
| Halaman Judul.....                                                                                | i         |
| Kata Pengantar.....                                                                               | iii       |
| Sistematika Penulisan Buku .....                                                                  | iv        |
| Daftar Isi .....                                                                                  | vi        |
| <b>Bab 1: Pendahuluan .....</b>                                                                   | <b>1</b>  |
| <b>1.1 Pengantar proyek deteksi dan visualisasi detak jantung .....</b>                           | <b>3</b>  |
| 1.1.1 Ulasan Singkat Tentang Jantung .....                                                        | 5         |
| 1.1.2 Cara Kerja Jantung .....                                                                    | 7         |
| 1.1.3 Sinyal Detak Jantung.....                                                                   | 7         |
| 1.1.4 Detak Jantung Normal Saat Berolahraga .....                                                 | 9         |
| <b>1.2 Pengantar proyek sistem pemantau infus berbasis Android .10</b>                            |           |
| 1.2.1 Latar Belakang .....                                                                        | 10        |
| 1.2.2 Pengertian Infus.....                                                                       | 12        |
| 1.2.3 Prinsip Kerja Infus .....                                                                   | 14        |
| 1.2.4 Cara Menghitung Tetesan Infus .....                                                         | 14        |
| <b>1.3 Pengantar proyek pemantau terapi jari pasca <i>stroke</i> secara <i>real-time</i>.....</b> | <b>15</b> |
| 1.3.1 Latar Belakang .....                                                                        | 15        |
| 1.3.2    Penyebab <i>Stroke</i> .....                                                             | 17        |
| 1.3.3    Upaya Rehabilitasi Pasien <i>Stroke</i> .....                                            | 21        |
| 1.3.4    Enam Terapi Dasar Pemulihan Pasca <i>Stroke</i> .....                                    | 23        |
| 1.3.5 Terapi Jari untuk Pemulihan Pasca <i>Stroke</i> .....                                       | 28        |

|                                                                                                                            |           |
|----------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>BAB 2: Sensor untuk Proyek Instrumentasi Medis .....</b>                                                                | <b>31</b> |
| <b>2.1 Sensor Detak Jantung .....</b>                                                                                      | <b>32</b> |
| <b>2.1.1 Cara Kerja Sensor Heart Rate .....</b>                                                                            | <b>33</b> |
| <b>2.1.2 Gelombang Elektrokardiogram .....</b>                                                                             | <b>34</b> |
| <b>2.2 Sensor Photodiode.....</b>                                                                                          | <b>36</b> |
| <b>2.3 Sensor Flex .....</b>                                                                                               | <b>38</b> |
| <b>Bab 3: Pengenalan Arduino .....</b>                                                                                     | <b>43</b> |
| <b>3.1 Apa itu Arduino? .....</b>                                                                                          | <b>44</b> |
| <b>3.2 Jenis-jenis board Arduino .....</b>                                                                                 | <b>44</b> |
| <b>3.3 Arduino IDE .....</b>                                                                                               | <b>46</b> |
| <b>3.4 Menginstall Software Arduino pada PC .....</b>                                                                      | <b>47</b> |
| <b>Bab 4: Pengenalan Processing .....</b>                                                                                  | <b>53</b> |
| <b>4.1 Sejarah.....</b>                                                                                                    | <b>55</b> |
| <b>4.2 Fitur .....</b>                                                                                                     | <b>55</b> |
| <b>4.3 Contoh Pemrograman .....</b>                                                                                        | <b>56</b> |
| <b>Bab 5: Mengkoneksikan Arduino dengan Processing.....</b>                                                                | <b>60</b> |
| <b>5.1 Cara mengirim data dari Arduino ke Processing melalui port serial .....</b>                                         | <b>61</b> |
| <b>5.2 Cara menerima data dari Arduino di Processing .....</b>                                                             | <b>65</b> |
| <b>5.3 Cara mengirim data dari Processing ke Arduino .....</b>                                                             | <b>71</b> |
| <b>5.4 Cara menerima data dari Processing di Arduino .....</b>                                                             | <b>73</b> |
| <b>Bab 6: Proyek Visualisasi detak jantung dengan menggunakan interface komputer berbasis Arduino dan Processing .....</b> | <b>78</b> |



|                                                                                                                                |            |
|--------------------------------------------------------------------------------------------------------------------------------|------------|
| 6.1 Bahan.....                                                                                                                 | 79         |
| 6.2 Alat .....                                                                                                                 | 80         |
| 6.3 Perancangan Sistem.....                                                                                                    | 81         |
| 6.3.1 Perancangan Perangkat Keras.....                                                                                         | 82         |
| 6.3.2 Perancangan Perangkat Lunak.....                                                                                         | 84         |
| 6.4 Implementasi Sistem .....                                                                                                  | 90         |
| 6.5 Hasil dan Pengujian Data.....                                                                                              | 92         |
| <b>BAB 7: Aplikasi Mobile Phone Untuk Monitoring Tingkat Kebugaran Seseorang Dengan Algoritma Fast Fourier Transform .....</b> | <b>106</b> |
| 7.1 Bahan Penelitian.....                                                                                                      | 107        |
| 7.2 Alat Penelitian.....                                                                                                       | 108        |
| 7.3 Perancangan Sistem.....                                                                                                    | 108        |
| 7.3.1 Perancangan Perangkat Keras.....                                                                                         | 109        |
| 7.3.2 Tata Cara Pemeriksaan Detak Jantung .....                                                                                | 114        |
| 7.3.3 Perancangan Perangkat Lunak .....                                                                                        | 115        |
| 7.4 Pengujian Sistem.....                                                                                                      | 124        |
| 7.4.1 Pengujian Perangkat Keras .....                                                                                          | 124        |
| 7.4.2 Pengujian Perangkat Lunak .....                                                                                          | 134        |
| <b>BAB 8: Rancang Bangun Sistem Pemantauan Infus Berbasis Android .....</b>                                                    | <b>165</b> |
| 8.1 Bahan Penelitian.....                                                                                                      | 166        |
| 8.2 Alat Penelitian.....                                                                                                       | 166        |
| 8.2.1 Perangkat keras .....                                                                                                    | 166        |

|                                                                                                                 |            |
|-----------------------------------------------------------------------------------------------------------------|------------|
| 8.2.2 Perangkat lunak .....                                                                                     | 167        |
| 8.3 Perancangan Sistem.....                                                                                     | 168        |
| 8.3.1 Perancangan Perangkat Keras.....                                                                          | 168        |
| 8.3.2 Perancangan Perangkat Lunak.....                                                                          | 172        |
| 8.4 Pengujian Sistem.....                                                                                       | 190        |
| 8.4.1 Pengujian Perangkat Keras.....                                                                            | 190        |
| 8.4.2 Pengujian perangkat lunak.....                                                                            | 190        |
| 8.4.3 Pengujian Keseluruhan alat.....                                                                           | 190        |
| 8.5 Implementasi Sistem .....                                                                                   | 191        |
| 8.5.1 Pengujian Perangkat Keras.....                                                                            | 192        |
| 8.6 Hasil dan Pengujian Data.....                                                                               | 203        |
| 8.6.1 Pengujian sensor pendeteksi tetesan .....                                                                 | 204        |
| 8.6.2 Pengujian Pengamatan Sisa Cairan Infus .....                                                              | 205        |
| <b>BAB 9: Aplikasi Pemantau Terapi Jari Pasca Stroke Menggunakan Sarung Tangan Pintar Secara Real Time.....</b> | <b>208</b> |
| 9.1. Subyek Penelitian .....                                                                                    | 209        |
| 9.2. Bahan Penelitian.....                                                                                      | 209        |
| 9.3. Alat Penelitian.....                                                                                       | 210        |
| 9.4. Perancangan Sistem .....                                                                                   | 211        |
| 9.4.1. Perancangan Perangkat Keras.....                                                                         | 211        |
| 9.4.2 Perancangan Perangkat Lunak.....                                                                          | 213        |
| 9.5. Pengujian Sistem.....                                                                                      | 220        |
| 9.5.1 Pengujian Sensor Flex.....                                                                                | 220        |

|                                                          |            |
|----------------------------------------------------------|------------|
| <b>9.5.2 Pengujian Sistem Secara Keseluruhan .....</b>   | <b>222</b> |
| <b>9.5.3 Pengujian Software Secara Keseluruhan .....</b> | <b>248</b> |
| <b>Referensi .....</b>                                   | <b>250</b> |
| <b>Profil Penulis .....</b>                              | <b>251</b> |

# Bab 1: Pendahuluan

## **Bab 1: Pendahuluan**

Perkembangan teknologi komputer yang sangat cepat memicu perkembangan teknologi lainnya. Dalam artian bahwa sangat banyak teknologi pada sektor lain yang memanfaatkan perkembangan teknologi komputer untuk proses pengembangan teknologi itu sendiri. Oleh karena itu, saat ini jarang sekali ditemukan perangkat - perangkat teknologi terbaru yang tidak mengandung unsur komputer di dalamnya. Begitu juga halnya dengan bidang kesehatan, penggunaan komputer sangat diperlukan untuk mempercepat dan mempermudah proses monitoring kondisi pasien secara terus menerus. Pada buku ini akan diuraikan tentang beberapa proyek instrumentasi medis berbasis *internet of things*, yang mana pada proyek ini akan menggabungkan teknologi sensor yang digunakan untuk bidang biomedis dengan teknologi *internet of things* yang akan menghubungkan sensor-sensor tersebut dengan aplikasi pada *smartphone* maupun komputer pribadi.

**Proyek ini dibagi menjadi 3 bagian utama yaitu:**

1. Proyek deteksi dan visualisasi detak jantung seseorang untuk mengukur tingkat kebugaran akan diulas pada bab 6 & 7.

2. Proyek *internet of things* untuk memantau kondisi infus dengan menggunakan interface *smartphone* android akan diulas pada bab 8.
3. Proyek Aplikasi pemantau terapi jari pasca stroke dengan *interface* komputer secara *real time* akan diulas pada bab 9.

### 1.1 Pengantar proyek deteksi dan visualisasi detak jantung

Dengan mengetahui denyut jantung seseorang, maka kita dapat melihat kondisi kesehatan jantung seseorang dengan sangat mudah. Jantung sebagai organ vital manusia berfungsi untuk memompa darah ke seluruh tubuh. Dalam proses ini, timbul bunyi akibat pergerakan katup-katup dari jantung. Untuk mendeteksi denyut jantung maka biasanya digunakan sensor pendeteksi yang kemudian digabungkan dengan peranan mikrokontroler.

Diagnosa denyut jantung berdasarkan suara yang direkam oleh *Pulse Heart Sensor* ditempelkan pada jari pasien yang dapat mengakibatkan subyektifitas dokter dalam memvisualisasikan sinyal tersebut. Dengan menggunakan komponen elektronika seperti mikrokontroler serta *pulse heart sensor* akan memungkinkan untuk merancang suatu alat yang dapat menghitung detak jantung pada manusia sehingga lebih efisien untuk penggunaannya serta harga yang relatif murah. Di sisi lain, penggunaan *electrocardiograph* (ECG) masih

tergolong relatif mahal jika digunakan sebagai alat diagnosa jantung. Proyek ini berfokus pada perancangan visualisasi isyarat detak jantung yang ditangkap oleh sensor untuk mempermudah proses diagnosa oleh dokter.

Dari masalah yang dihadapi perlu dilakukan perancangan suatu alat dengan merancang sebuah sistem menggunakan komputer yang dapat memantau sinyal detak jantung dengan menggunakan mikrokontroler berbasis Arduino dan *Pulse Heart Sensor*. Sensor akan ditempelkan pada ujung jari pasien untuk mendeteksi sinyal detak jantung, data akan dikirim ke mikrokontroler untuk diproses dan kemudian akan ditampilkan pada komputer. Pada saat sinyal detak jantung berada pada kondisi yang telah ditetapkan maka akan muncul hasil berupa jumlah detak jantung per menit dan pulsa yang ditampilkan pada layar monitor kemudian dapat disimpan dalam memori, sehingga data hasil pembacaan tersebut dapat dibandingkan dengan data referensi dari alat ukur detak jantung standar.

Pada penelitian yang dilakukan, perekaman detak jantung dengan menggunakan *Pulse Heart Sensor* dan selanjutnya data dari pembacaan sensor diproses oleh arduino dengan pin analog (A0), menggunakan fitur *Analog to Digital Converter (ADC)*, kemudian selanjutnya diolah menjadi satuan bpm (*Beats Per Minute*) untuk dapat divisualisasikan menjadi isyarat detak jantung. Dengan alat

sederhana yang dirancang pada proyek ini, beragam gelombang suara dari detak jantung pasien dapat ditampilkan di monitor komputer sehingga tidak perlu lagi untuk mengukur detak jantung secara manual.

### 1.1.1 Ulasan Singkat Tentang Jantung

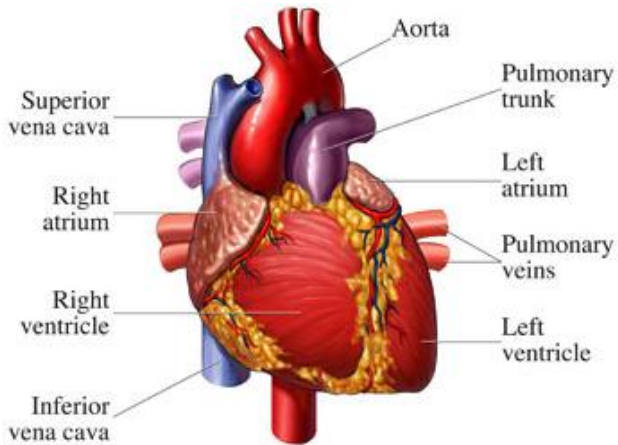
Jantung (Latin, cor) adalah rongga organ berotot yang berfungsi untuk memompa darah melalui pembuluh darah oleh kontraksi berirama yang berulang. Jantung adalah salah satu organ tubuh manusia yang berperan dalam sistem peredaran darah. Jantung merupakan otot tunggal yang terdiri dari lapisan endothelium. Jantung terletak dalam rongga thoracic, di balik tulang dada/sternum. Letak jantung berbelok ke bawah dan sedikit ke arah kiri.

Hampir seluruh permukaan jantung diselubungi oleh paru-paru, tapi tertutup oleh selaput ganda yang disebut perikardium, dan tertempel pada diafragma. Lapisan pertama menempel sangat erat dengan jantung, sedangkan lapisan luarnya lebih longgar dan berair, fungsinya adalah untuk menghindari gesekan antar organ tubuh yang terjadi karena pergerakan memompa yang dilakukan oleh jantung.

Jantung dijaga di tempatnya oleh pembuluh darah yang menyelubungi seluruh permukaan jantung, seperti di bagian bawah dan di bagian samping. Dua garis pembelah (terbentuk dari otot)



pada lapisan luar jantung menunjukkan di mana dinding pemisah di antara sebelah kiri dan kanan serambi (atrium) & bilik (ventrikel). Gambar 1.1 menunjukkan struktur jantung.



**Gambar 1.1** Struktur internal jantung

Dengan meraba denyut pada bagian arteri, kita dapat menghitung kecepatan detak jantung yang berbeda-beda yang disebabkan oleh aktifitas, makanan, umur, dan emosi seseorang. Standar jumlah denyut jantung manusia ditunjukkan dalam tabel 2.2.

**Tabel 2.2** Perbedaan denyut jantung manusia

| No | Umur                 | Jumlah denyut / menit (BPM) |
|----|----------------------|-----------------------------|
| 1  | Bayi baru lahir      | 140                         |
| 2  | Selama tahun pertama | 120                         |
| 3  | Selama tahun kedua   | 110                         |
| 4  | Pada umur 5 tahun    | 96-100                      |

|   |                    |       |
|---|--------------------|-------|
| 5 | Pada umur 10 tahun | 80-90 |
| 6 | Dewasa             | 60-80 |

Sumber: Pearce, 2000: 127-128

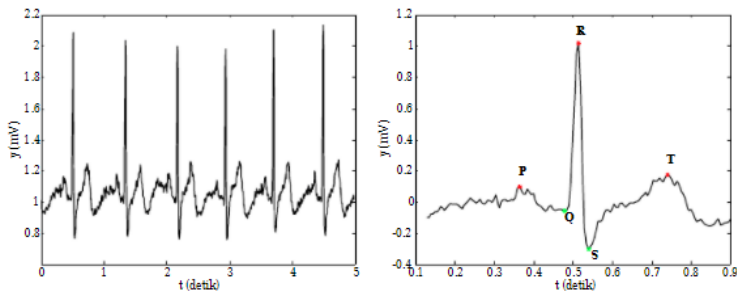
### 1.1.2 Cara Kerja Jantung

Fungsi utama jantung adalah untuk menyediakan suplai oksigen ke seluruh bagian tubuh dan membersihkan tubuh dari hasil meta-bolisme (CO<sub>2</sub>). Jantung menjalankan fungsi tersebut dengan mengumpulkan darah yang kurang oksigen dari seluruh tubuh dan memompanya ke dalam paru-paru, dimana darah akan mengambil oksigen dan membuang CO<sub>2</sub>. Pada saat jantung berdenyut, setiap ruang pada jantung mengendur dan terisi oleh darah (disebut sebagai periode diastolik). Selanjutnya jantung berkontraksi dan memompa darah keluar dari jantung (disebut sebagai periode sistolik). Dalam rangkaian sistemik, darah yang kaya oksigen akan meninggalkan tubuh melalui ventrikel kiri ke aorta, dan dari sana darah akan masuk arteri dan kapiler dimana jaringan tubuh disuplai dengan oksigen.

### 1.1.3 Sinyal Detak Jantung

*Electrocardiograph* (ECG) adalah serangkaian gambaran yang mencerminkan aktivitas listrik jantung. Sebuah lead yang dilengkapi dengan bahan konduktif ditempatkan pada bagian tubuh yang berbeda, sehingga mungkin untuk melacak sinyal listrik pada jantung

dari beberapa bagian yang berbeda. Jika arus listrik jantung mengarah menuju ke lead menghasilkan garis yang naik pada grafik (defleksi positif). Jika arus listrik jantung mengalir menjauhi lead maka akan menghasilkan garis turun (defleksi negatif). Gambaran arus listrik jantung ini kemudian digambarkan dengan tampilan grafik. Pada jantung yang sehat dan normal, gambaran ECG yang mewakili satu detakan jantung lengkap dapat dilihat pada Gambar 1.2.



**Gambar 1.2** Data detak jantung (kiri) dan Detak jantung dalam satu gelombang (kanan)

Pada Gambar 1.2 terlihat titik-titik puncak maksimum lokal dan minimum lokal. Puncak - puncak tersebut mem-punyai makna fisi yang disimbolkan sebagai P, Q, R, S dan T sebagaimana ditunjukkan pada Gambar 1.2 kanan. Irama jantung normal dapat dikatakan sebagai irama sinus yaitu irama yang terletak pada sekitar Vena Cava Superior di atrium kanan jantung. Irama jantung yang teratur yang

berarti jarak antara gelombang yang relatif sama dan teratur. Hubungan P dengan Q, R dan S adalah bertujuan untuk membedakan suatu irama jantung, bentuk dan durasi pada puncak merupakan pembesaran pada atrium jantung. Sedangkan pada puncak-puncak Q, R dan S ditujukan untuk mendeteksi suatu irama jantung, abnormalitas konduksi. Gelombang T menggambarkan bahwa adanya kembali proses pemompaan kedalam ventrikel jantung. Pada umumnya puncak T bernilai positif apabila puncak T negatif atau terbalik maka bisa terjadi ketidaknormalan pada jantung. Berdasarkan Gambar 1.2 terlihat bahwa posisi puncak ke puncak S terhadap potensial selalu berada pada posisi negatif sedangkan untuk puncak-puncak T selalu berada pada posisi positif. Data yang telah diukur merupakan data untuk jantung yang sehat. Pada penelitian ini ditentukan posisi P, Q, R, S dan T untuk keseluruhan data yang diukur. Untuk itu, diperlukan periode satu gelombang. Untuk menentukan periode satu gelombang maka syarat utama dari satu gelombang adalah satu gelombang harus memuat puncak-puncak P, Q, R, S dan T. Dengan contoh satu gelombang ditunjukkan pada Gambar 1.2 kanan.

#### **1.1.4 Detak Jantung Normal Saat Berolahraga**

Detak jantung dapat digunakan sebagai indikator untuk menentukan bahwa seseorang melakukan aktivitas terlalu keras.

Detak jantung normal orang saat berolahraga perlu dikenali, agar tidak berlebihan dalam melakukan aktivitas. Pada saat berolahraga detak jantung setiap manusia berbeda - beda tergantung berdasarkan usia.

## 1.2 Pengantar proyek sistem pemantau infus berbasis Android

### 1.2.1 Latar Belakang

Perkembangan sains dan teknologi saat ini begitu pesat, seiring dengan pesatnya perkembangan di bidang elektronika dan mikrokontroller, Arduino menjadi salah satu papan pengembangan berbasis mikrokontroller yang paling banyak diminati akhir-akhir ini. Arduino merupakan papan pengembangan berbasis mikrokontroller yang relatif mudah untuk diprogram, Arduino dilengkapi dengan pin/port yang dapat digunakan untuk berkomunikasi dengan perangkat lain. Dengan kata lain arduino adalah sebuah papan pengembangan mikrokontroller yang sangat populer saat ini. Akhir-akhir ini juga sedang marak penggunaan *smartphone* android sebagai penunjang aktifitas sehari-hari. Android merupakan nama system operasi yang bersifat "*Open Source*" yang pada umumnya ditujukan untuk *smartphone* dan juga tablet. Sebagai perangkat yang bergerak, Android dapat terhubung dengan peralatan elektronika untuk mengendalikan ataupun menerima informasi dari peralatan

elektronika. Untuk memicu berkembangnya inovasi baru yang memanfaatkan arduino dan android yang dirancang untuk mempermudah pekerjaan manusia. Salah satunya adalah di bidang kesehatan.

perkembangan teknologi ini diharapkan dapat mempermudah pekerjaan perawat/dokter dalam proses pemantauan kondisi pasien. Salah satu contohnya adalah dalam proses pemantauan infus. infus berfungsi untuk memberikan sejumlah cairan khusus kedalam tubuh pasien melalui media jarum kedalam pembuluh vena (pembuluh balik) yang bertujuan untuk mengganti kehilangan cairan atau zat makanan dari tubuh saat pasien sakit.

pada dasarnya infus sering sekali digunakan oleh pasien, oleh karena itu infus haruslah selalu dipantau kondisinya. Hingga saat ini proses pemantauan infus masih dilakukan secara manual, dimana perawat/dokter harus mengecek kondisi infus pasien secara langsung dan berkala. Saat cairan infus habis, perawat tidak bisa langsung mengganti/mengisi ulang cairan infus karena harus lalu lalang dari kamar pasien satu ke kamar pasien yang lain, hal ini menyebabkan infus bisa terlambat untuk diganti/diisi ulang, keterlambatan penggantian infus ini dapat menyebabkan penyumbatan udara, sehingga dapat berakibat fatal bagi pasien. Penyumbatan udara

terjadi apabila terdapat gelembung udara yang turut serta masuk ke dalam sistem sirkulasi pasien.

Berdasarkan permasalahan tersebut, untuk dapat memecahkan masalah maka peneliti merancang sebuah sistem menggunakan *Smartphone* android yang dapat memantau tetesan infus dengan menggunakan mikrokontroler berbasis arduino dan sensor photodiode. Untuk mendeteksi adanya tetesan pada infus, sensor photodiode dipasang pada chamber infus, kemudian data hasil pembacaan sensor dikirimkan ke mikrokontroler untuk diolah, kemudian data hasil pengolahan data oleh mikrokontroler dikirim ke ponsel Android melalui bluetooth. pada saat cairan infus berada pada batas bawah yang telah ditetapkan, maka mikrokontroler akan mengirimkan peringatan atau pemberitahuan ke ponsel perawat.

### 1.2.2 Pengertian Infus

*intravenous fluids infusion* (infus) adalah proses pemberian sejumlah cairan ke dalam tubuh pasien melalui sebuah jarum ke dalam pembuluh vena untuk menggantikan kehilangan cairan atau zat makanan dari tubuh saat pasien menderita sakit tertentu. Secara umum, keadaan tubuh yang memerlukan pemberian cairan infus adalah:

1. Perdarahan dalam jumlah yang relatif banyak (kehilangan cairan tubuh dan komponen darah)

2. Trauma abdomen (perut) berat (kehilangan cairan tubuh dan komponen darah)
3. Fraktur (patah tulang), khususnya di pelvis (panggul) dan femur (paha) (kehilangan cairan tubuh dan komponen darah)
4. “Serangan panas” (*heat stroke*) (kehilangan cairan tubuh pada dehidrasi)
5. Diare dan demam (mengakibatkan dehidrasi)
6. Luka bakar yang cukup luas (kehilangan banyak cairan tubuh)
7. Semua trauma kepala, dada, dan tulang punggung (kehilangan cairan tubuh dan komponen darah)

**Pada infus terdapat beberapa bagian yang harus diketahui. Adapun bagian – bagian infus adalah sebagai berikut:**

1. Botol infus: merupakan wadah dari cairan infus, biasa dijumpai dan dijual dalam tiga varian ukuran 500mL, 1000mL dan 1500mL.
2. Selang infus: merupakan alat untuk mengalirkan cairan infus
3. Klem selang infus: merupakan bagian untuk mengatur laju aliran dari cairan infus, dengan mempersempit atau memperlebar jalur aliran pada selang.
4. Jarum infus: alat untuk memasukkan cairan infus dari selang infus ke pembuluh vena.



### 1.2.3 Prinsip Kerja Infus

Prinsip kerja dari cairan infus adalah sama dengan sifat dari air biasa yaitu mengalir dari tempat yang lebih tinggi ke tempat yang lebih rendah, dipengaruhi oleh gaya gravitasi bumi sehingga cairan akan selalu jatuh kebawah. Pada sistem infus laju aliran infus dapat diatur melalui klem selang infus, jika klem digerakan untuk mempersempit jalur aliran pada selang maka laju cairan akan menjadi lebih lambat, ini ditandai dengan sedikitnya jumlah tetesan infus/menit yang keluar dan sebaliknya bila klem digerakan untuk memperlebar jalur aliran pada selang infus maka laju cairan infus akan menjadi lebih cepat dan ditandai dengan banyaknya jumlah tetesan infus/menit.

### 1.2.4 Cara Menghitung Tetesan Infus

Untuk menghitung jumlah tetesan cairan infus, kita perlu mengetahui Istilah yang sering digunakan dan juga rumus tetap perhitungan tetes infus. Dalam pemasangan infus diketahui istilah dan rumus tetap sebagai berikut:

- gtt = makro tetes
- mgtt = mikro tetes
- jumlah tetesan = banyaknya tetesan dalam satu menit
- 1 gtt = 3 mgtt
- 1 cc = 20 gtt

- 1 cc = 60 mgtt
- 1 kolf = 1 labu = 500 cc
- 1 cc = 1 mL
- mggt/menit = cc/jam

Untuk lebih memahami istilah-istilah tersebut, kita harus terlebih dahulu mengetahui formula untuk menghitung jumlah tetes cairan infus dalam hitungan menit dan jam.

- faktor tetes Otsuka — 1cc = 15 tetes
- faktor tetes Terumo — 1 cc = 20 tetes

(Kebutuhan cairan x faktor tetes) = Jumlah tetesan/menit  
(jumlah jam x 60menit).

## 1.3 Pengantar proyek pemantau terapi jari pasca stroke secara real-time

### 1.3.1 Latar Belakang

*Stroke* merupakan istilah yang digunakan untuk menamakan *sindroma* (kumpulan gejala) *hemiparesis* atau *hemiparalisis* (lumpuh sebelah) akibat kerusakan pada pembuluh darah yang bisa bangkit dalam waktu beberapa detik hingga beberapa hari, tergantung pada jenis penyakit yang menjadi penyebabnya. Sehingga, sebenarnya *stroke* bukanlah nama penyakit, melainkan istilah untuk menjelaskan

kumpulan beberapa gejala yang muncul akibat adanya kerusakan pada pembuluh darah dalam otak.

*Stroke* termasuk dalam penyakit *serebrovaskuler* (pembuluh darah otak) yang ditandai dengan matinya jaringan otak (*infark serebral*) yang terjadi akibat berkurangnya aliran darah dan oksigen ke bagian otak. *World health organization* (WHO) mendefinisikan *stroke* sebagai gejala-gejala defisit dari fungsi susunan saraf yang diakibatkan oleh adanya penyakit pada pembuluh darah dalam otak, bukan karena alasan yang lain.

*Stroke* sebenarnya diambil dari kata *strike*, pada umumnya merupakan serangan mendadak yang dapat berakibat fatal tanpa adanya gejala-gejala yang bisa dirasakan sebelumnya. Penderita *stroke* pada umumnya tidak tahu kapan dan bagaimana ia akan mendapatkan serangan *stroke*. Berdasarkan penelitian yang dilakukan di *Oxfordshire Community Stroke Project*, tidak ada hubungan *stroke* berulang dengan umur, jenis kelamin, tipe patologi *stroke*, dan riwayat penyakit jantung atau *fibrilasi atrium*.

Seseorang yang punya riwayat terserang penyakit *stroke* mempunyai kecenderungan lebih besar akan mengalami serangan *stroke* susulan/lanjutan, apalagi jika faktor resiko *stroke* yang ada, tidak ditanggulangi dengan baik. Oleh karena itu, perlu diupayakan prevensi sekunder yang meliputi gaya hidup sehat dan pengendalian

faktor risiko yang bertujuan untuk mencegah terjadinya serangan *stroke* berulang pada seseorang yang sebelumnya pernah terserang *stroke*.

Oleh karena itu, kita patut untuk berwaspada dalam menjalankan pola hidup. Usia muda tidak menjamin kita untuk terhindar dari serangan *stroke* jika tidak dibarengi dengan pola hidup yang sehat. Seseorang yang mengkonsumsi banyak gula dalam minuman dan makanan cepat saji memiliki resiko terserang *stroke* pada usia yang relatif tidak terlalu tua.

### 1.3.2 Penyebab Stroke

Menurut dr. Abdul Ghofir, S.p.S, munculnya serangan *stroke* dipicu oleh besarnya faktor risiko penyakit pendukung lainnya seperti penyakit jantung, saraf, diabetes melitus, darah tinggi, *dislipidemia*, obesitas dan usia yang sudah relatif tua, yang mana menyebabkan fungsi motorik, sensorik, saraf *kranialis*, dan fungsi *kognitif* lainnya menjadi lemah. Selain itu, gaya hidup yang kurang sehat juga menjadi faktor yang mempercepat datangnya serangan penyakit *stroke*, seperti kebiasaan merokok, konsumsi alkohol, Konsumsi makanan berkolesterol tinggi, dll. Selain itu, jenis kelamin dan ras juga menjadi salah satu faktor penentu timbulnya serangan penyakit *stroke*. Angka kejadian penyakit *stroke* lebih banyak dialami oleh wanita daripada laki-laki dikarenakan perbedaan profil faktor resiko *vaskular* dan

*substipe* dari *stroke*. Hal itu disebabkan karena wanita memiliki kecacatan *stroke* yang lebih berat dibandingkan dengan laki-laki. Berikut adalah beberapa penyebab terjadinya serangan *stroke*:

### 1. Hipertensi

Orang yang memiliki tekanan darah tinggi (*hipertensi*) memiliki peluang yang lebih besar untuk terkena *stroke*. Bahkan, *hipertensi* merupakan penyebab terbesar dari serangan penyakit *stroke*. *Hipertensi* mampu meningkatkan risiko terkena *stroke* 2-4 kali lipat, dan tidak tergantung pada faktor risiko yang lainnya. Alasannya adalah karena dalam kasus *hipertensi*, dapat terjadi gangguan aliran darah dalam tubuh, dimana diameter pembuluh darah akan mengecil, sehingga intensitas darah yang mengalir ke otak akan berkurang. Dengan berkurangnya intensitas aliran darah otak (ADO), maka otak akan mengalami kekurangan suplai oksigen dan glukosa atau disebut dengan *hipoksia*. Karena suplai darah yang berkurang secara terus-menerus, sehingga lama-kelamaan jaringan otak akan mengalami kematian. Seseorang disebut mengalami *hipertensi* apabila tekanan darahnya lebih tinggi dari 140/90 mmHg atau lebih tinggi dari 135/85 mmHg.

## 2. Diabetes Melitus

Penderita *Diabetes Melitus* atau penyakit kencing manis juga memiliki risiko untuk mengalami *stroke*. Hal ini disebabkan karena pembuluh darah penderita *diabetes* yang umumnya menjadi lebih kaku atau kehilangan kelenturan. Adanya peningkatan ataupun penurunan kadar glukosa darah yang terjadi secara tiba-tiba juga dapat menyebabkan kematian pada jaringan otak. Seseorang dianggap memiliki penyakit *diabetes mellitus* apabila kadar glukosa plasma darahnya lebih tinggi dari 126 mg% dan diperiksa pada dua waktu yang berbeda.

## 3. Merokok

Berdasarkan hasil penelitian, didapatkan bahwa orang yang merokok memiliki kadar *fibrinogen* darah yang lebih tinggi daripada orang yang tidak merokok. Peningkatan kadar *fibrinogen* ini dapat mempermudah terjadinya penebalan pada pembuluh darah, sehingga pembuluh darah menyempit dan kaku, dengan demikian hal tersebut dapat menyebabkan gangguan pada aliran darah. Merokok dapat meningkatkan risiko *stroke* sebesar 1,5 kali lipat setelah faktor risiko lainnya dikendalikan. Asap rokok juga dapat meningkatkan terjadinya *aterosklerosis* terutama pada *arteri karotis*. Pada pasien yang merokok, kerusakan organ yang diakibatkan oleh *stroke* jauh lebih parah karena dinding bagian dalam (*endothelial*) pada sistem

pembuluh darah otak (*serebrovaskular*) biasanya sudah menjadi lemah, sehingga dapat menyebabkan kerusakan yang lebih parah lagi pada otak sebagai akibat apabila terjadi serangan *stroke* tahap kedua

#### **4. Kurang Olahraga/Aktifitas Fisik**

Kurangnya aktivitas fisik seseorang merupakan salah satu faktor risiko yang menyebabkan terjadinya *stroke* dan penyakit jantung. Olahraga rutin rata-rata 30 menit/hari dapat menurunkan risiko terjadinya serangan *stroke*. Olahraga juga dapat mengendalikan beberapa masalah kesehatan lainnya, seperti obesitas, diabetes melitus, meningkatkan kadar kolesterol HDL, dan pada sekelompok individu dapat menurunkan tekanan darah. Hubungan antara aktivitas fisik dan *stroke* telah diteliti secara terus-menerus dan dilaporkan sejak 38 tahun silam. hal-hal yang mempunyai kaitan dengan faktor risiko *stroke* lain, yaitu merokok dan minum alkohol secara berlebihan.

#### **5. Obesitas (kegemukan)**

Obesitas merupakan salah satu faktor yang menyebabkan terjadinya *stroke*. *Obesitas* memberi risiko *stroke* 2 kali lipat. Hal tersebut disebabkan karena *Obesitas* memberi beban yang lebih berat kepada jantung, selain itu *obesitas* juga merupakan *predisposisi* untuk meningkatnya kadar kolesterol total, *trigliserid*, *hipertensi*, menurunnya kadar kolesterol HDL, dan *diabetes mellitus*. *Obesitas*

bisa dikontrol dengan melakukan diet dan olahraga secara teratur dengan disiplin yang tinggi. Namun, ada juga *obesitas* yang didasari oleh faktor genetik, sehingga dengan usaha apa pun berat badan sulit untuk diturunkan secara drastis.

### 1.3.3 Upaya Rehabilitasi Pasien Stroke

Perawatan terhadap penderita *stroke* memiliki dua tujuan utama, yang pertama adalah untuk meminimalkan cedera pada jaringan otak dan yang kedua adalah untuk mengobati komplikasi yang dapat terjadi pada pasien pasca *stroke*, baik kerusakan syaraf maupun kerusakan fisik. Pasien penderita *stroke* membutuhkan upaya rehabilitasi agar ke depannya mereka mampu mandiri untuk beraktifitas dalam kehidupan sehari-hari tanpa harus terus-menerus menjadi beban bagi keluarganya.

Upaya rehabilitasi atau pemulihan bagi pasien yang pernah menderita serangan *stroke*, sebenarnya tidak harus selalu dilakukan di fasilitas rehabilitasi rumah sakit. Secara umum tahap rehabilitasi pasca *stroke* dapat ditangani dengan tahapan rehabilitasi sederhana yang dapat dilakukan oleh tenaga kesehatan maupun keluarga pasien, karena hal tahapan rehabilitasi sebenarnya tidak membutuhkan peralatan yang canggih. Upaya rehabilitasi tersebut bertujuan untuk mencegah terjadinya komplikasi karena kelumpuhan, yang dapat membawa dampak buruk bagi kondisi



pasien *stroke*. Selain itu rehabilitasi juga bertujuan untuk mengembalikan kemandirian pasien dalam melakukan aktivitas sehari-hari. Dengan upaya ini diharapkan pasien *stroke* dapat menjalani hidup yang lebih berkualitas.

Rehabilitasi pada pasien penderita *stroke* akan memberikan hasil yang maksimal jika dilakukan dalam 3 bulan pertama pasca *stroke*. Meskipun perkembangan pemulihan yang optimal bisa didapatkan dalam jangka waktu 3 bulan pertama, namun proses pemulihan perlu terus dilakukan seumur hidup untuk mengembalikan kemandirian pasien secara optimal. Oleh karena itu, sangatlah penting untuk memulai melakukan rehabilitasi sedini mungkin dan dilakukan secara berkesinambungan.

Salah satu prinsip umum dalam melakukan upaya rehabilitasi pada pasien *stroke* adalah dengan memperbanyak gerakan pada bagian tubuh yang mengalami sakit maupun tidak. Hal tersebut merupakan terapi yang sangat baik. Karena gerakan dapat meningkatkan proses pembentukan sirkuit saraf di dalam otak. Salah satu contoh gerakan yang perlu dilakukan antara lain adalah gerakan meraih, memegang dan mengangkat gelas ke mulut. Bila bagian tubuh yang terkena *stroke* masih terlalu lemah, maka perlu diberikan bantuan tenaga secukupnya, namun pasien *stroke* masih tetap menggunakan ototnya secara aktif. Gerakan ini dapat dilakukan pada

posisi duduk maupun berdiri. Waktu latihan dapat berkisar antara 45-60 menit, dengan melakukan pengulangan sesering mungkin. Latihan mencapai lingkup gerak penuh pada semua persendian disertai latihan regangan otot sedikitnya 2 kali per hari diperlukan. Upayakan lengan selalu dalam posisi terbuka saat sedang tidak melakukan latihan.

#### 1.3.4 Enam Terapi Dasar Pemulihan Pasca Stroke

Pasien yang telah sembuh dari stroke Sangat dianjurkan untuk sedini mungkin memulai langkah-langkah dalam rangka rehabilitasi. Bahkan pada penderita stroke yang mengalami koma sekalipun, dapat dibantu memulai gerakan-gerakan latihan secara pasif (dengan bantuan orang lain) jika kondisi penderita sudah mulai stabil. Saat penderita telah sadar, maka selanjutnya dapat dilanjutkan dengan latihan aktif yang dilakukan oleh penderita itu sendiri. Rehabilitasi pasca stroke dapat dimulai saat penderita masih dalam proses perawatan oleh dokter di rumah sakit, sehingga tidak perlu menunggu sampai penderita pulang ke rumah.

Adapun 6 terapi dasar pasca stroke adalah sebagai berikut:

##### 1. Terapi Fisik

6 bulan pertama setelah *stroke* merupakan *gold period* (masa keemasan/masa terbaik) untuk melakukan proses rehabilitasi pasca

*stroke*. Oleh karena itu, tidak ada lagi alasan untuk menunda-nunda memulai latihan. Beberapa manfaat yang dapat diperoleh dengan melakukan latihan fisik, antara lain adalah membantu meningkatkan penggunaan ekstremitas/anggota gerak tubuh, memperkuat otot yang lemah setelah *stroke*, mengaktifkan kembali fungsi tubuh yang mengalami kelumpuhan, mandiri dan tidak bergantung pada orang lain, meningkatkan daya tahan tubuh dan mencegah terjadinya depresi.

Latihan fisik yang akan dilakukan oleh seorang yang pernah menderita *stroke* haruslah mengikuti beberapa aturan dasar supaya hasilnya dapat optimal. Beberapa hal yang perlu diperhatikan ketika melakukan latihan fisik antara lain; menitikberatkan pada latihan kekuatan, koordinasi, keseimbangan dan kestabilan, memulai latihan dengan pemanasan terlebih dahulu agar otot dan sendi tidak menjadi kaku, tidak memaksakan kemampuan diri sampai titik yang tidak dikuasai, memakai alat bantu terlebih dahulu dan secara perlahan mencoba untuk melepas alat bantu tersebut.

Latihan fisik secara bertahap ini dapat dimulai ketika penderita *stroke* masih terbaring di tempat tidur, namun dengan kondisi yang sudah dinyatakan stabil oleh dokter. Diawali dengan gerakan berbaring miring dengan dibantu orang lain (keluarga, perawat, maupun ahli fisioterapi) dari posisi lurus kemudian menekuk. Apabila

sudah memungkinkan, maka latihlah penderita untuk duduk secara mandiri, tentunya dengan dibantu terlebih dahulu kemudian lama kelamaan dapat dilakukan secara mandiri.

Di sela-sela istirahat, pasien dapat melakukan latihan pada jari-jari tangan, seperti menekuk jari, menjepit dan memegang sesuatu. Semakin sering melakukan latihan maka hasil yang didapatkan akan semakin optimal. Usahakanlah untuk memaksimalkan peran aktif dari pasien, sedangkan peran keluarga/perawat/ahli fisioterapis hanya membantu dan memberikan dukungan.

## **2. Terapi Okupasi**

Terapi okupasi bertujuan untuk meningkatkan kemampuan rawat mandiri bagi pasien dan mengupayakan pasien mampu melakukan aktivitas harian dengan mandiri. Tahap ini dapat dimulai apabila pasien sudah dapat melakukan beberapa gerakan aktif, seperti berjalan perlahan (meski masih memakai alat bantu), memegang, dan sebagainya. Dengan dukungan dan kasih sayang keluarga, maka pasien *stroke* akan mampu menjalankan aktivitas hariannya dengan baik meski dengan segala keterbatasan. Dengan demikian terapi okupasi ini sangatlah penting untuk diberikan kepada pasien selama masih dalam proses penyembuhan.

### **3. Terapi Wicara**

Terapi wicara pada umumnya melibatkan ahli atau terapis wicara. Namun demikian, dukungan keluarga juga tetap memegang peranan penting. Misalnya dengan tetap melakukan komunikasi dengan berbicara, walaupun pasien masih belum mampu meresponnya. Berikan himbauan kepada seluruh penghuni rumah untuk menghargai penderita dan menginformasikan apapun yang akan dikerjakan, misalnya meminta izin ketika akan menggantinya seprei, memakaikan baju, dan lain-lain.

### **4. Terapi Psikologis**

Terapi psikologis dilakukan untuk mengurangi tekanan/stress pada psikologis pasien karena memikirkan kondisi kesehatannya. Dibutuhkan suasana yang hangat dalam keluarga supaya pasien bahagia dan merasa diperhatikan. Ketika berbicara, hendaknya kita mendekat pada mereka, tidak dengan berteriak atau bersuara keras. Jangan sesekali membentak mereka, karena hal tersebut akan sangat melukai hati pasien dan menurunkan kondisi psikologisnya.

### **5. Terapi Hobi**

Terapi hobi merupakan salah satu penunjang dalam keberhasilan pemulihan pasien *stroke*. Dukung dan temani mereka untuk melakukan hobinya, misalnya seperti membaca, menyulam, atau memasak. Dengan demikian, pasien akan terhindar dari stress

dan dapat mengisi waktu luangnya dengan kegiatan yang bermanfaat dan menyenangkan. Selama hobi tersebut tidak membahayakan, maka berikanlah dukungan dan bantuan, karena pada dasarnya kemampuan fisik pasien pasca *stroke* akan berkurang.

## **6. Terapi Spiritual**

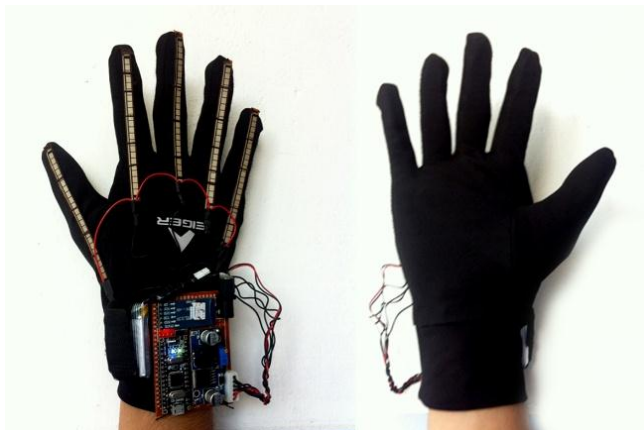
Kebutuhan spiritual seseorang yang telah mengalami *stroke* sangatlah penting untuk menjadi perhatian. Ingatkan mereka untuk mengerjakan shalat/ibadah meski tidak dengan posisi orang normal (jika tidak mampu berdiri) dan ajak mereka untuk mengikuti kajian/pengajian jika kondisinya memungkinkan. Apabila kita hendak mengingatkan atau menyampaikan nasihat, hendaklah dengan cara yang sopan dan halus. Jangan sampai penderita berputus asa dengan kondisi kesehatannya. Berikan selalu semangat dan ingatkan agar selalu bersabar supaya mendapatkan pahala dari Allah. Sampaikanlah bahwa kondisi sakit yang dialaminya dapat menghapuskan dosa-dosanya selama ia menerima takdir dengan sabar.

Selain dengan cara tersebut kita juga dapat membawa pasien pasca *stroke* kepada ustadz/pemuka agama untuk diberikan nasihat agar selalu bersabar dan tabah, sehingga kondisi spiritual pasien lama-kelamaan akan semakin membaik.

### 1.3.5 Terapi Jari untuk Pemulihan Pasca Stroke

Salah satu dari sekian banyak terapi yang dapat dilakukan untuk para penderita *stroke* di rumah adalah dengan melakukan gerakan terapi berupa terapi pada jari-jari tangan. Adapun terapi jari-jari tangan dapat dilakukan dengan melakukan gerakan-gerakan ringan menggunakan jari-jari tangan seperti:

1. Meluruskan semua jari sehingga semua jari dalam keadaan lurus sempurna atau membentuk isyarat angka 5 seperti yang ditunjukkan pada Gambar 1.3



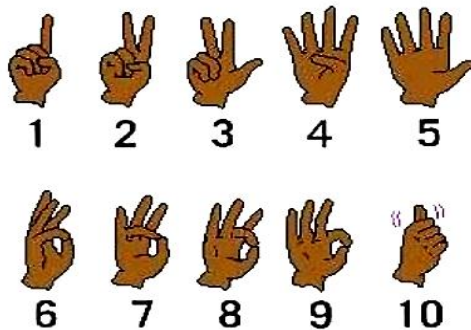
**Gambar 1.3** Meluruskan Semua Jari atau Isyarat Anka 5

2. Menggenggam penuh semua jari tangan sehingga semua jari dalam posisi terlipat seperti yang ditunjukkan pada Gambar 1.4



**Gambar 1.4** Menggenggam Penuh Semua Jari

3. Menekuk jari tengah, jari manis, kelingking dan jempol sehingga membentuk isyarat angka 1, kemudian mengganti tekukan jari sehingga membentuk beberapa isyarat angka 2,3,4,5,6,7,8,9 dan 10 seperti yang ditunjukkan pada Gambar 1.5



**Gambar 1.5** Macam-macam Isyarat Angka Menggunakan Jari tangan



4. Serta beberapa gerakan terapi jari lainnya dapat dilakukan di rumah penderita *stroke* dengan dibantu keluarga. Beberapa gerakan lainnya dapat dilihat pada Gambar 1.6



**Gambar 1.6** Terapi Jari

Beberapa contoh di atas merupakan latihan tangan dan jari-jari untuk pasien *stroke* yang dapat dilakukan di rumah. Untuk itu dampingan seorang fisioterapis sangat penting agar meminimalisir pembelajaran fungsi dan gerak yang salah. Semua terapi harus dengan catatan hindari *stretching* dan rasa sakit agar otot-otot tidak cepat memendek.

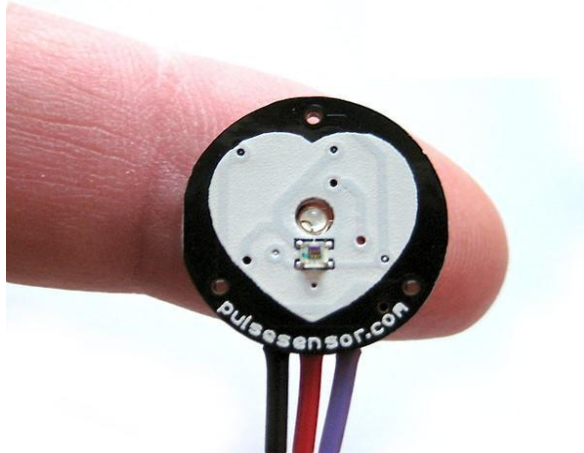
## BAB 2: Sensor untuk Proyek Instrumentasi Medis

## **BAB 2: Sensor untuk Instrumentasi Medis**

Pada bab ini akan dijelaskan secara singkat mengenai sensor-sensor yang digunakan untuk membuat project instrumentasi medis yang dibahas pada buku ini.

### **2.1 Sensor Detak Jantung**

*HeartRate Sensor* adalah perangkat medis yang berfungsi untuk mendeteksi/membaca denyut jantung manusia. Rangkaian dasar dari sensor ini terdiri dari *phototransistor* dan *LED*. Sensor ini bekerja berdasarkan prinsip dasar pantulan sinar *LED*, kemudian Kulit menjadi permukaan reflektif untuk sinar *LED*, kepadatan darah pada kulit akan mempengaruhi reflektifitas sinar *LED*. Aksi pemompaan jantung mengakibatkan kepadatan darah meningkat, sehingga Pada saat jantung memompa darah, darah akan mengalir melalui pembuluh arteri dari bagian yang besar hingga ke bagian yang kecil seperti pada ujung jari. Volume darah pada ujung jari akan bertambah, sehingga intensitas cahaya yang mengenai *phototransistor* akan kecil karena terhalang oleh darah, begitu pula sebaliknya. Output sinyal dari *phototransistor* kemudian akan dikuatkan oleh rangkaian *Op-Amp* (penguat sinyal) sehingga dapat dibaca oleh fasilitas ADC pada mikrokontroler. Gambar 2.1 menunjukkan bentuk fisik dari sensor *HeartRate*.



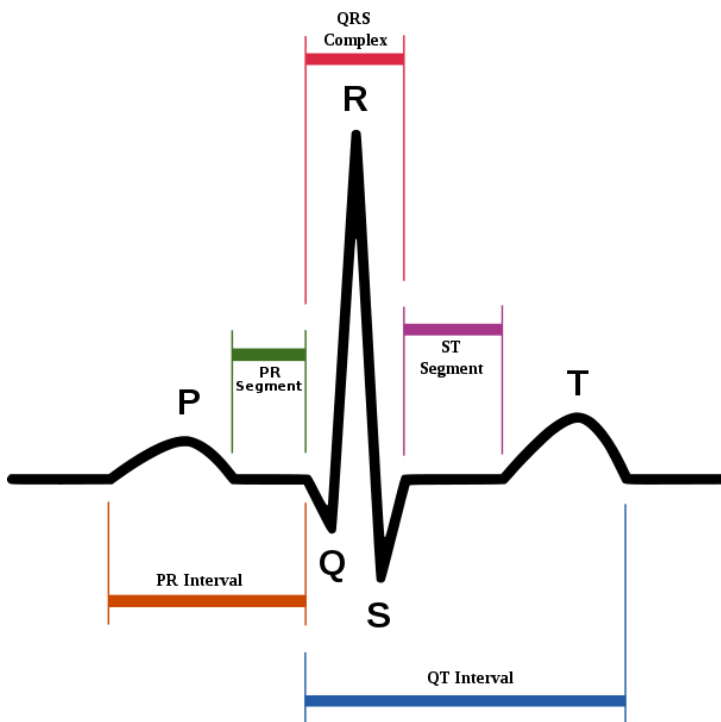
**Gambar 2.1** *HeartRate sensor* atau sensor detak jantung

### **2.1.1 Cara Kerja Sensor Heart Rate**

Cara kerja dari sensor *HeartRate* yaitu dengan cara sensor diletakkan di salah satu ujung jari tangan, dimana cahaya *LED* yang menembus jari diterima oleh rangkaian penerima atau *receiver*. Rangkaian *receiver* ini terdiri dari resistor sebagai pembatas arus dan *photodiode*. Selain *receiver* adapun juga rangkaian *transmitter* yang terdiri dari resistor sebagai pembatas arus dan *LED infrared*, kemudian diteruskan dalam rangkaian penguat mikrokontroler Arduino untuk diproses dalam program.

### 2.1.2 Gelombang Elektrokardiogram

EKG merupakan rekaman aktivitas listrik dari jantung. Dimana aktivitas listrik atrium digambarkan oleh gelombang P dan aktivitas listrik ventrikel digambarkan oleh gelombang Q, R, S dan T. Gambar 2.2. menunjukkan gambaran dari gelombang PQRST jantung.



**Gambar 2.2** Gelombang Sinyal EKG

Pada dasarnya terdapat 5 syarat dasar yang harus dimiliki oleh sebuah gelombang EKG yang normal, antara lain:

- a. **Gelombang P.** Gelombang ini pada umumnya berukuran kecil dan merupakan hasil depolarisasi atrium kanan dan kiri. Kelainan pada atrium akan menyebabkan kelainan bentuk pada gelombang ini. Gelombang ini diukur dari permulaan gelombang P sampai permulaan gelombang QRS.
- b. **Segmen PR.** Segmen ini merupakan garis isoelektrik yang menghubungkan gelombang P dan gelombang QRS. Bagian ini menggambarkan aktivitas listrik dari atrium ke ventrikel.
- c. **Gelombang Kompleks QRS.** Gelombang kompleks QRS adalah satu kelompok gelombang yang merupakan hasil depolarisasi ventrikel kanan dan kiri.
- d. **Gelombang ST.** Segmen ini merupakan garis isoelektrik yang menghubungkan kompleks QRS dan gelombang T.
- e. **Gelombang T.** Gelombang ini merupakan potensial repolarisasi ventrikel kanan dan kiri.

Selain itu, sinyal EKG juga memiliki parameter dari setiap gelombang. Parameter EKG dapat dilihat pada Tabel 2.1.

**Tabel 2.1** Parameter EKG

| <b>Gelombang EKG</b> | <b>Amplitudo</b> | <b>EKG Interval</b> | <b>Durasi</b>     |
|----------------------|------------------|---------------------|-------------------|
| P                    | < 0,3 mV         | P – R               | 0,12 – 0,20 detik |
| R                    | 1,6 – 3 mV       | Q – T               | 0,35 – 0,44 detik |
| Q                    | 25% dari R       | S – T               | 0,05 – 0,15 detik |
| T                    | 0,1 – 0,5 mV     | Q – R – S           | 0,06 – 0,10 detik |

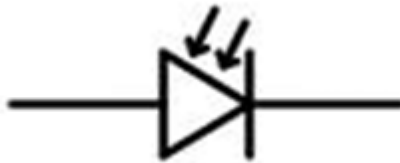
(Sumber: Suryana & Aziz, 2017)

Berdasarkan interval antara R–R kita dapat mengetahui periode detak jantung yang dapat dikonversikan menjadi satuan *Heart Rate* (HR). Pada dasarnya interval R–R relatif konstan dari detak ke detak. Perubahan pada interval R–R ini menandakan adanya kecepatan detak jantung yang tak wajar. Hal tersebut dapat terjadi dikarenakan beberapa faktor diantaranya; kondisi kesehatan, tingkat aktivitas, suhu udara, dan emosi seseorang. Faktor-faktor tersebut sangat mempengaruhi kinerja jantung.

## **2.2 Sensor Photodiode**

Photodiode adalah salah satu komponen elektronika yang berasal dari keluarga diode yang memiliki fungsi sebagai pendeteksi

cahaya. Meskipun merupakan jenis dioda, tetapi prinsip kerjanya merupakan kebalikan dari dioda pada umumnya. Photodiode akan mengubah cahaya menjadi arus listrik. Komponen elektronika ini mampu mendeteksi berbagai jenis cahaya, mulai dari cahaya infra merah, cahaya tampak, sinar ultra violet sampai dengan Sinar-X. Photodiode dapat digunakan untuk banyak aplikasi mulai dari penghitung kendaraan di jalan umum secara otomatis, pengukur cahaya pada kamera serta beberapa aplikasi pada peralatan di bidang medis. Simbol photodiode dapat dilihat pada Gambar 2.3.

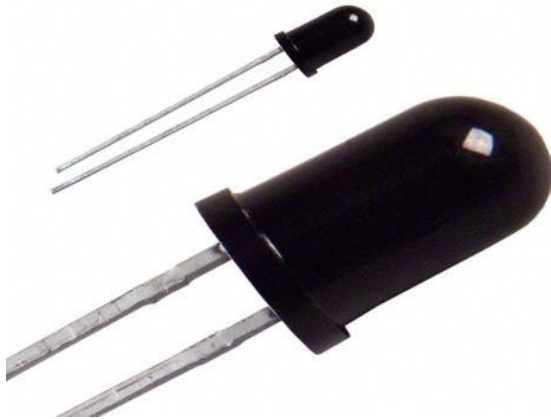


**Gambar 2.3** Simbol Photodiode

Photodiode merupakan sensor cahaya yang dapat mengalirkan arus listrik satu arah (DC) dari satu sisi ke sisi lainnya, ketika Photodiode menyerap cahaya. Semakin banyak cahaya yang diserap, maka semakin banyak pula arus yang dapat mengalir. Photodiode juga biasa digunakan untuk mendeteksi pulsa cahaya dalam serat optik yang sangat sensitif terhadap pergerakan cahaya. prinsip kerja



photodiode pada dasarnya merupakan kebalikan dari prinsip kerja LED. Bentuk fisik photodiode dapat dilihat pada Gambar 2.4.



**Gambar 2.4** Bentuk fisik photodiode

### **2.3 Sensor Flex**

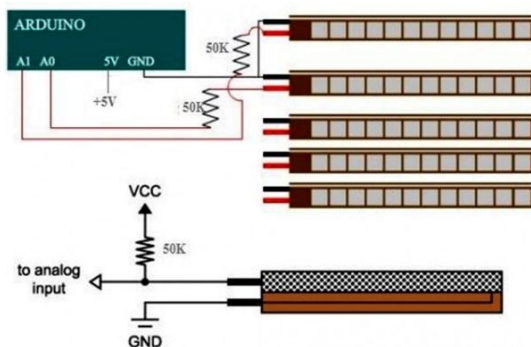
Sensor *flex* adalah sensor yang berfungsi untuk mendeteksi suatu kelengkungan atau tekukan jari-jari tangan. Prinsip kerjanya sama dengan *potensiometer*. Untuk menggunakan sensor *flex*, kita membutuhkan bantuan *arduino nano* yang digunakan sebagai mikrokontroler untuk membaca data ADC berupa tegangan yang dikeluarkan dari suatu *flex* sensor. Sensor *flex* dapat di aplikasikan pada beberapa perangkat, biasanya digunakan untuk kendali robot, sebagai pembaca isarat tangan digital, serta sebagai pembaca gerakan jari-jari tangan yang diterapkan pada penelitian ini.

Gambar 2.5 merupakan wujud fisik dari sensor *flex* yang sering digunakan.



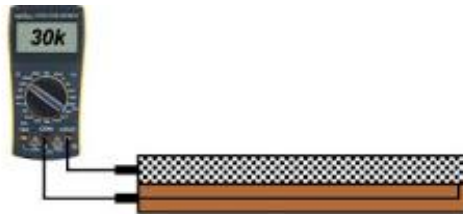
**Gambar 2.5** bentuk fisik sensor Flex

Sensor *flex* yang kita gunakan dalam proyek ini adalah sensor *flex* berukuran 4,5 inci. Arduino nano digunakan sebagai prosesor dari sistem yang dirancang, pin yang dimanfaatkan untuk pembacaan sensor yaitu *analog pin* dari *pin* 0 sampai 4 (A0-A4). Gambar 2.6 merupakan rangkain skematik dari sensor *flex* yang dipasangkan ke arduino nano.

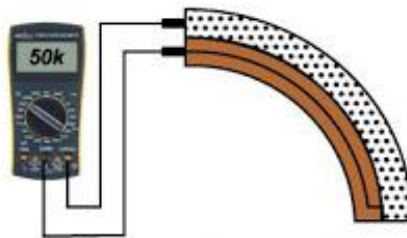


**Gambar 2.6** Skematik Pemasangan Sensor *Flex*

Adapun prinsi kerja dari sensor *flex* dapat dilihat seperti yang dijelaskan pada Gambar 2.7 di bawah ini.



Saat antar partikel saling berdekatan maka hambatan akan kecil yaitu 30KOhm



Saat antar partikel saling berjauhan maka hambatan akan besar yaitu 50KOhm

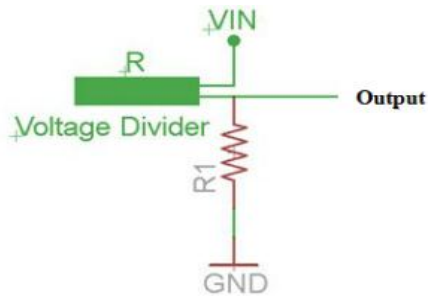
**Gambar 2.7** Prinsip Kerja Sensor *Flex*

Prinsip kerja dasar dari sensor *flex* adalah ketika sensor *flex* tidak ditekuk (dalam kondisi lurus) maka partikel di dalam sensor saling berdekatan, sehingga nilai hambatan akan menjadi kecil yaitu berkisar 30 KOhm. Sedangkan ketika sensor ditekuk maka partikel di dalam sensor akan saling berjauhan sehingga mengakibatkan membesarnya nilai hambatan, yaitu 50 KOhm.

### - Rangkaian Sensor *Flex*

Sensor *flex* atau disebut juga sensor kelenturan berfungsi untuk membaca fleksibilitas dan lekukan dari sensor tersebut. *Output* dari sensor tersebut berupa tegangan analog yang dapat langsung dihubungkan dan di baca oleh mikrokontroler melalui fasilitas ADC.

Rangkaian dasar sensor *flex* dapat dilihat pada Gambar 2.8



**Gambar 2.8** Rangkaian Sensor *Flex*



## Bab 3: Pengenalan Arduino

## Bab 3: Pengenalan Arduino

Pada bab ini akan dibahas secara singkat mengenai platform Arduino baik terkait *hardware* maupun *software*.

### 3.1 Apa itu Arduino?

Arduino adalah sebuah platform *open source* berbasis mikrokontroler yang banyak digunakan untuk membuat berbagai macam proyek elektronika. Arduino dibagi menjadi 2 bagian utama yaitu sebuah papan sirkuit fisik (atau bisa disebut *board* mikrokontroler) dan sebuah perangkat lunak/*software* berupa IDE (*Integrated Development Environment*) yang dapat dijalankan pada perangkat komputer/laptop. Perangkat lunak ini digunakan untuk membuat dan meng-upload kode program dari komputer ke *board* Arduino.

Arduino memiliki banyak varian atau jenis *board* yang dapat kita gunakan untuk pembuatan proyek tergantung dari kebutuhan port input/output dan kecepatan/*speed clock* yang kita butuhkan untuk proyek yang ingin kita rancang. Pada Bab ini ulasan singkat tentang arduino akan diuraikan.

### 3.2 Jenis-jenis board Arduino

Seperti yang dikatakan pada sub-bab sebelumnya bahwa arduino memiliki banyak sekali varian *board*, yang mana setiap varian

memiliki fitur dan fungsi yang berbeda-beda, namun dengan metode pemrograman yang sama. Berikut adalah beberapa varian *board* dari keluarga Arduino:

- 1. Arduino Uno**
- 2. Arduino Duemilanove**
- 3. Arduino Leonardo**
- 4. Arduino Mega2560**
- 5. Arduino Intel Galile**
- 6. Arduino Pro Micro AT**
- 7. Arduino Nano R3**
- 8. Arduino mini Atmega**
- 9. Arduino Mega ADK**
- 10. Arduino Esplora**

Untuk detail dari jenis-jenis *board* Arduino yang telah disebutkan dapat di lihat pada halaman web resmi Arduino yaitu:

[www.arduino.cc](http://www.arduino.cc)



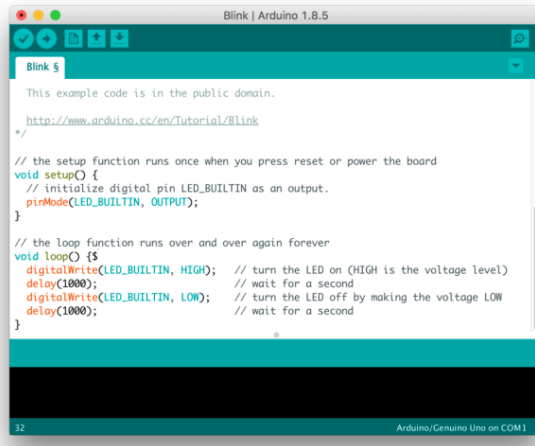
### 3.3 Arduino IDE

Sesuai dengan topik yang dibahas dalam buku ini *software* utama yang akan kita gunakan adalah Arduino IDE. Arduino IDE merupakan komponen utama yang tidak dipisahkan dengan perangkat keras Arduino, arduino IDE menggunakan basis bahasa C++. Arduino IDE terdiri dari beberapa bagian utama antara lain':

1. **Editor program**, adalah sebuah jendela yang memungkinkan pengguna untuk menulis dan mengedit kode program dalam bahasa C++.
2. **Compiler**, adalah sebuah modul yang mengubah kode program yang ditulis dengan Bahasa C++ menjadi kode biner. Karena pada dasarnya sebuah mikrokontroler tidak bisa memahami bahasa C++, namun mikrokontroler hanya dapat memahami kode biner saja Itulah sebabnya modul *compiler* diperlukan dalam hal ini.
3. **Uploader**, adalah sebuah modul yang digunakan untuk memuat kode biner yang telah di *compile* dari komputer ke dalam memori Arduino.

Kode program Arduino pada umumnya disebut dengan istilah *sketch*. Kata "*sketch*" digunakan secara bergantian dengan "kode program" dimana pada dasarnya kedua istilah tersebut memiliki arti yang sama.

Tampilan dari Arduino IDE dengan sebuah *sketch* editor dapat kita lihat pada gambar 3.1

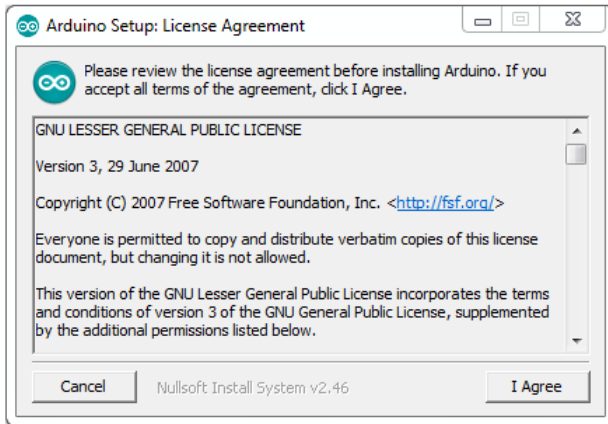


**Gambar 3.1.** Tampilan Arduino IDE

### 3.4 Menginstall Software Arduino pada PC

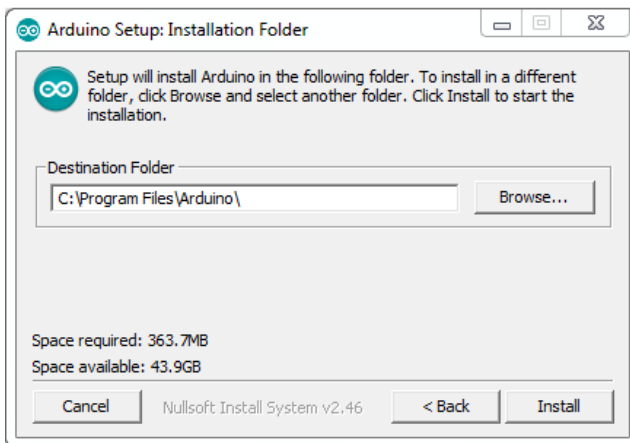
Software Arduino IDE dapat diunduh secara gratis disitus [www.arduino.cc](http://www.arduino.cc) untuk memudahkan pengguna, adapun prosedur menginstal program tersebut seperti berikut:

1. Setelah berhasil mengunduh file installer, Klik ganda pada *software* arduino IDE untuk segera memulai proses instalasi.
2. Setelah file installer dijalankan, akan muncul jendela “*License Agreement*”. Klik saja tombol “*I Agree*” di sajikan pada Gambar 3.2



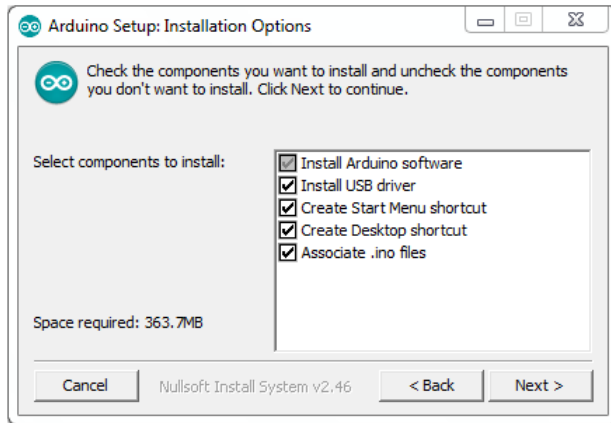
**Gambar 3.2** Proses instal *software* Arduino

3. Berikut akan diminta memasukkan folder instalasi Arduino. Biarkan *default* di C:\Program Files\Arduino, tetapi bisa diganti dengan folder yang lain disajikan pada Gambar 3.3



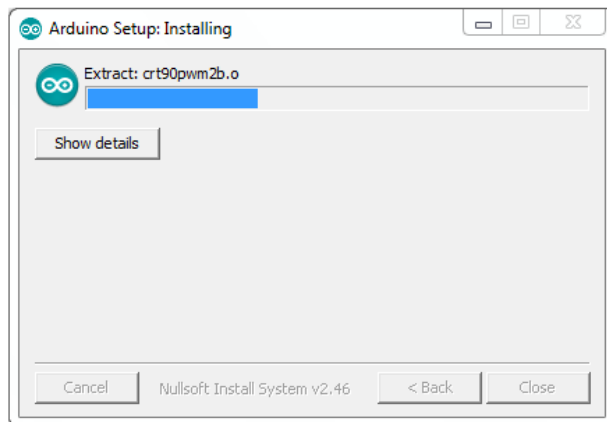
**Gambar 3.3** *Default* folder instalasi

- Setelah itu akan muncul jendela “*Setup Installation Option*”, sebaiknya centang semua opsinya disajikan pada Gambar 3.4



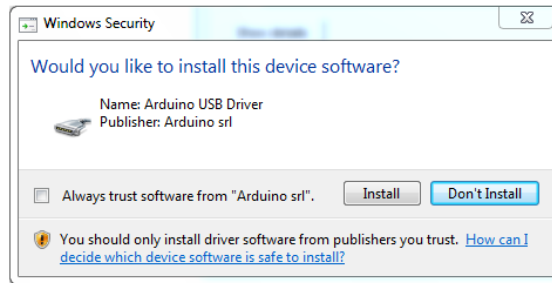
**Gambar 3.4** Pemilihan komponen untuk di instal

- Selanjutnya proses instalasi akan dimulai dan menunggu agar proses instalasinya selesai disajikan pada Gambar 3.5



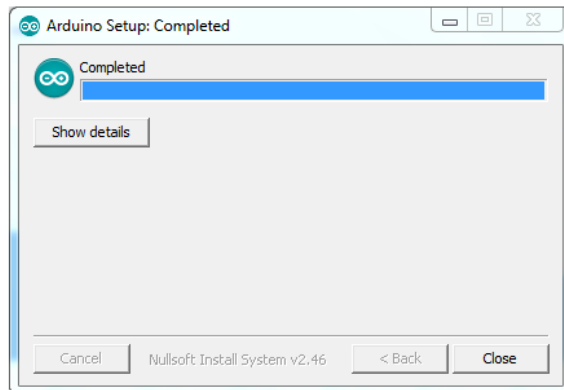
**Gambar 3.5** Proses Instalasi *software* Arduino

6. Di tengah proses instalasi, jika komputer yang digunakan belum terinstal driver USB, maka akan muncul jendela “*security Warning*” sebagai berikut pilih saja tombol ‘*Instal*’ yang disajikan pada Gambar 3.6



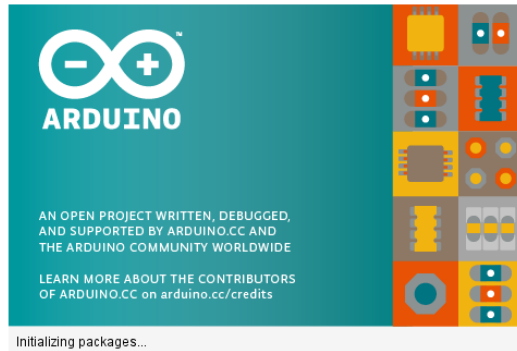
**Gambar 3.6** Proses instal USB *driver*

7. Tunggu sampai proses instalasi ‘*Completed*’ yang disajikan pada Gambar 3.7



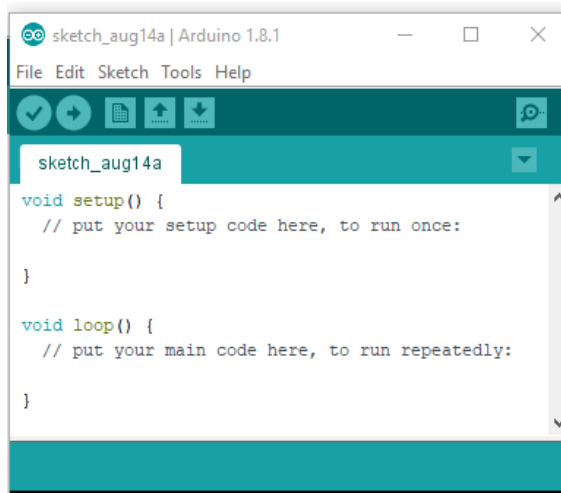
**Gambar 3.7** Proses instalasi *completed*

8. Pada tahap ini *software* IDE Arduino sudah terinstal, coba cek di start Menu Windows dan seharusnya akan muncul ikon Arduino, kemudian jalankan aplikasi tersebut sehingga akan menampilkan aplikasi yang disajikan pada Gambar 3.8



**Gambar 3.8** Tampilan awal aplikasi Arduino

9. Beberapa detik kemudian, jendela IDE Arduino akan muncul jendela untuk tempat pemrograman yang disajikan pada Gambar 3.9



**Gambar 3.9** Jendela utama papan pemrograman

# Bab 4: Pengenalan Processing



## Bab 4: Pengenalan Processing



**Gambar 4.1** Icon Processing IDE

Processing adalah pustaka grafis *open-source* berbasis *Integrated development and environment* (IDE) yang dibuat untuk keperluan seni elektronik (membuat visualisasi grafis), seni media baru, dan komunitas desain visual dengan tujuan untuk mengajar orang awam atau non-programmer dasar-dasar pemrograman komputer dalam konteks visual.

Processing menggunakan bahasa Java, dengan penyederhanaan tambahan seperti kelas tambahan dan fungsi dan operasi matematika. Selain itu, Processing juga memiliki antarmuka pengguna grafis untuk menyederhanakan tahap kompilasi dan eksekusi. Bahasa dan IDE Processing adalah pendahulu untuk proyek-proyek lain termasuk Arduino, Wiring dan p5.js.

## 4.1 Sejarah

Proyek Processing dimulai pada tahun 2001 oleh Casey Reas dan Ben Fry, dimana keduanya berasal dari Aesthetics and Computation Group di MIT Media Lab. Pada 2012, mereka memulai Yayasan Processing bersama Daniel Shiffman, yang kemudian bergabung sebagai pemimpin proyek ketiga. Johanna Hedva bergabung dengan Yayasan pada tahun 2014 sebagai Direktur Advokasi.

Awalnya, Processing memiliki URL di `proce55ing.net`, karena domain Processing diambil. Akhirnya Reas dan Fry memperoleh domain `processing.org`. Meskipun nama itu memiliki kombinasi huruf dan angka, itu masih diucapkan Processing. Mereka tidak suka lingkungan yang disebut sebagai `Proce55ing`. Meskipun perubahan nama domain, Processing masih menggunakan istilah `p5` kadang-kadang sebagai nama singkat (`p5` khusus digunakan, bukan `p55`), misalnya `p5.js` adalah referensi untuk itu.

## 4.2 Fitur

Processing termasuk buku sketsa, alternatif IDE untuk mengatur proyek. Setiap sketsa Processing sebenarnya adalah subkelas dari kelas Java `PApplet` (sebelumnya merupakan subkelas

dari Applet bawaan Java) yang mengimplementasikan sebagian besar fitur bahasa Processing.

Saat memprogram dalam Processing, semua kelas tambahan yang ditentukan akan diperlakukan sebagai kelas dalam ketika kode diterjemahkan ke dalam Java murni sebelum dikompilasi. Ini berarti bahwa penggunaan variabel statis dan metode di kelas dilarang kecuali Processing secara eksplisit diberitahu untuk kode dalam mode Java murni.

Processing juga memungkinkan bagi pengguna untuk membuat kelas mereka sendiri dalam sketsa PApplet. Hal ini memungkinkan untuk tipe data kompleks yang dapat menyertakan sejumlah argumen dan menghindari batasan hanya menggunakan tipe data standar seperti: int (integer), char (karakter), float (angka nyata), dan warna (RGB, RGBA, hex).

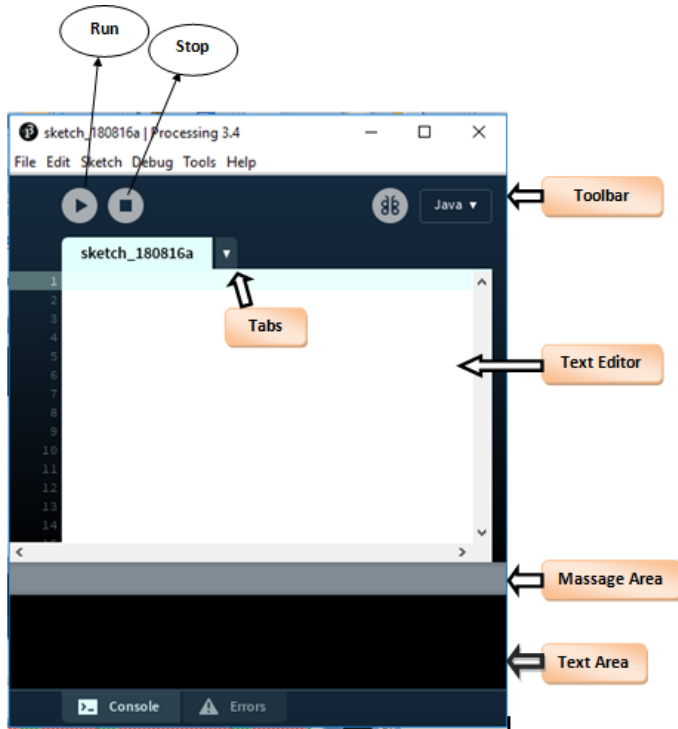
### 4.3 Contoh Pemrograman

Versi paling sederhana yang mungkin dari program "Hello World" dalam Processing adalah:

```
//This prints "Hello World." to the IDE console.  
println("Hello World.");
```

Namun, karena sifat Processing yang lebih berorientasi visual, kode berikut adalah contoh yang lebih baik dari tampilan dan nuansa bahasa.

```
//Hello mouse.  
void setup() {  
  size(400, 400);  
  stroke(255);  
  background(192, 64, 0); }  
void draw()  
{  
  line(150, 25, mouseX, mouseY);  
}
```



**Gambar 4.2** Tampilan utama *software processing*

Gambar 4.2 merupakan tampilan awal *software processing* ada beberapa menu yang perlu diketahui fungsinya.

- Run** digunakan untuk mengeksekusi *sketch* pada *software processing*
- Stop** Menghentikan *sketch* yang sedang dijalankan
- Toolbar** adalah bagian dimana tombol-tombol kontrol ditempatkan.

- d. **Tabs** adalah tempat dimana file yang terdiri dari banyak *sketch* ditempatkan.
- e. **Area Text Editor** adalah tempat untuk menuliskan program.
- f. **Message Area** merupakan tempat dimana berbagai pesan dari processing dikeluarkan, misalnya terjadi *error* ataupun kesalahan dalam format memasukan program.
- g. **Text Area** merupakan area tempat ditampilkannya output-output program, misalnya dalam menampilkan hasil perhitungan, data berupa numerik dan string, serta data komunikasi secara serial.

# Bab 5: Mengkoneksikan Arduino dengan Processing

## **Bab 5: Mengkoneksikan Arduino dengan Processing**

Pada bab sebelumnya telah dibahas terkait pengenalan Arduino dan processing secara singkat, namun pada proyek yang dibahas dalam buku ini, salah satu hal terpenting yang perlu kita ketahui adalah bagaimana cara untuk mengkoneksikan kedua platform tersebut sehingga proyek yang dikerjakan dapat berjalan dengan baik. Pada bab ini kita akan membahas bagaimana cara mengkoneksikan Arduino dengan Processing.

Pada bab ini kita akan belajar:

- Cara mengirim data dari Arduino ke Processing melalui port serial
- Cara menerima data dari Arduino di Processing
- Cara mengirim data dari Processing ke Arduino
- Cara menerima data dari Processing di Arduino

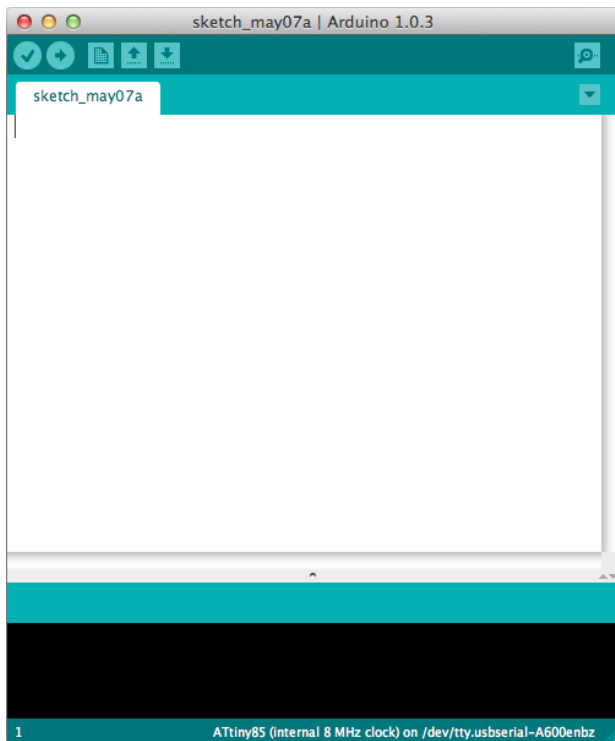
### **5.1 Cara mengirim data dari Arduino ke Processing melalui port serial**

Mari kita mulai dengan sisi Arduino. Kami akan menunjukkan kepada Anda dasar-dasar cara mengatur sketsa Arduino Anda untuk mengirim informasi melalui serial.



Pada titik ini Anda harus sudah menginstal perangkat lunak Arduino, board Arduino, dan kabel. Setelah itu adalah tahap menulis kode program. Jangan khawatir ini cukup mudah untuk dilakukan!

Buka perangkat lunak Arduino. Akan muncul tampilan aplikasi seperti pada gambar 5.1



**Gambar 5.1** Tampilan awal *software* Arduino

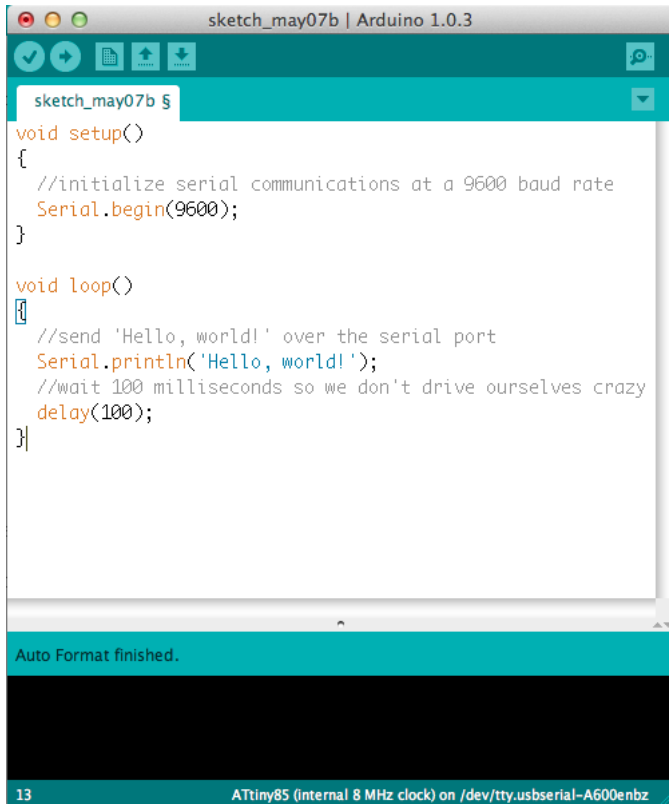
Ruang putih besar yang bagus adalah tempat kita akan menulis kode. Klik di area putih dan ketikkan kode program berikut:

```
void setup()
{
    Serial.begin(9600);
}
```

Ini disebut metode pengaturan kita. Di situlah kita bisa 'mengatur' program yang kita ingin buat. Di sini, kita menggunakannya untuk memulai komunikasi serial dari Arduino ke komputer pada *baud rate* 9600. Untuk saat ini, yang Anda butuhkan sekarang tentang *baud rate* adalah bahwa pada dasarnya itu adalah tingkat di mana kita mengirimkan data ke komputer, dan jika kita mengirim dan menerima data pada tingkat yang berbeda, semuanya akan berjalan dengan baik.

Setelah metode `setup ()`, kita membutuhkan metode yang disebut `loop ()`, dimana bagian ini akan berulang-ulang selama program berjalan. Untuk contoh pertama, kita hanya akan mengirim string 'Hello, world!' melalui port serial, berulang-ulang. Ketikkan kode program beriku ini di *sketch* Arduino Anda:

```
void loop()
{
    Serial.println("Hello, world!");
    delay(100);
}
```



**Gambar 5.2** tampilan sketch untuk mengirim data dari Arduino ke Processing

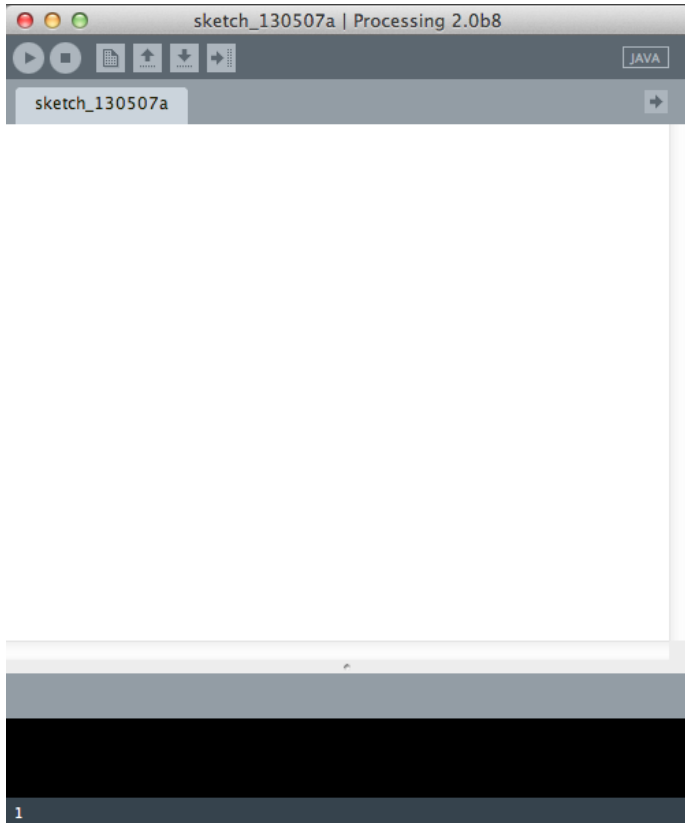
Hanya itu yang kita butuhkan dari sisi Arduino untuk contoh pertama yang kita buat. Kita sedang menyiapkan komunikasi serial dari Arduino dan akan mengirim data setiap 100 milidetik. *sketch* Arduino Anda sekarang akan terlihat seperti gambar 5.2

Kemudia yang perlu dilakukan hanyalah menyam-bungkan *board* Arduino Anda, pilih jenis *board* Anda (di bawah *tools* -> *Board Type*) dan port Serial Anda (di bawah *Tools* -> *Serial Port*) dan tekan tombol 'upload' untuk memuat kode program Anda ke *board* Arduino.

Selanjutnya kita siap untuk melihat apakah kita dapat secara ajaib atau melalui kode mendeteksi string 'hello, world!' yang kami kirim ke Processing.

## 5.2 Cara menerima data dari Arduino di Processing

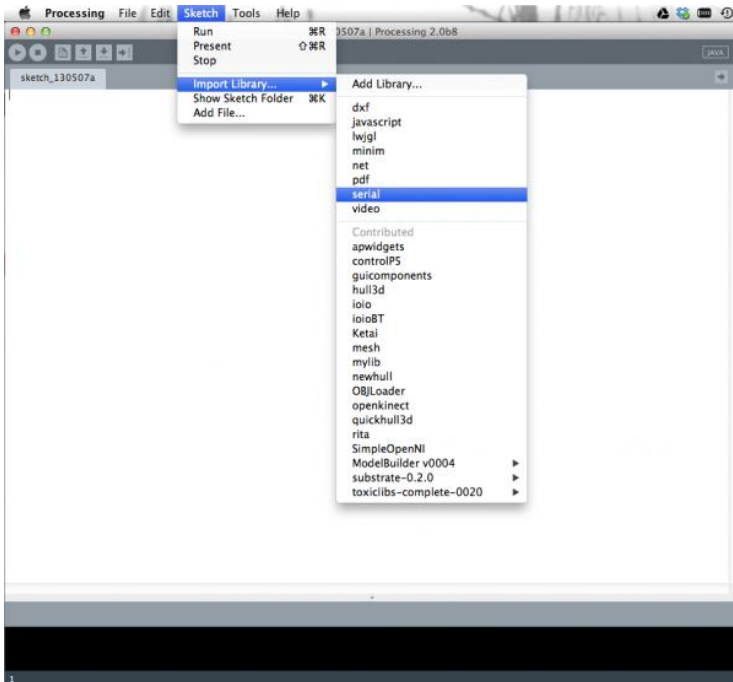
Tugas kita sekarang adalah menemukan cara untuk mendapatkan data yang dikirim dari *sketch* Arduino yang telah kita buat. Untungnya, Processing hadir dengan pustaka (*library*) Serial yang dirancang untuk hal semacam ini! Jika Anda tidak memiliki versi Processing, pastikan Anda pergi ke [Processing.org](http://Processing.org) dan unduh versi terbarunya untuk sistem operasi Anda. Setelah Processing diinstal, kemudia buka. Anda akan melihat tampilan seperti pada gambar 5.3



**Gambar 5.3** tampilan awal Processing IDE

Tampak sangat mirip Arduino, ya? Perangkat Lunak Arduino sebenarnya didasarkan pada bagian dari Processing - itulah keindahan proyek dengan sumber terbuka (*open source*). Setelah kita membuat *Sketch* terbuka, langkah pertama yang perlu kita lakukan

adalah mengimpor *library* Serial. Pergi ke *Sketch*-> *Import Library*-> *Serial*, seperti yang ditunjukkan pada gambar 5.4



**Gambar 5.3** cara import *library* pada Processing

Anda seharusnya sekarang melihat baris seperti import `processing.serial.*`; di bagian atas sketsa Anda. Di bawah pernyataan impor, kita perlu mendeklarasikan beberapa variabel global. Semua ini berarti bahwa variabel-variabel ini dapat digunakan di mana saja dalam *sketch* yang kita buat. Tambahkan dua baris ini di bawah pernyataan impor:

```
Serial myPort;  
String val;
```

Untuk mendapatkan data dari komunikasi serial, kita harus mendapatkan objek Serial (kita dapat menyebutnya myPort; tetapi Anda juga dapat menamainya sesuka Anda), yang memungkinkan kita mendapatkan data pada port serial di komputer untuk setiap data yang masuk. Kita juga membutuhkan variabel untuk menerima data aktual yang masuk. Dalam hal ini, karena kita mengirim jenis data String (urutan karakter 'Hello, World!') Dari Arduino, kita juga ingin menerima String dalam processing. Sama seperti Arduino memiliki `setup ()` dan `loop ()`, Processing memiliki `setup ()` dan `draw ()` (bukan `loop`).

Untuk metode `setup ()` dalam Processing, kita akan menemukan port serial yang terhubung dengan Arduino dan kita dapat mengatur objek Serial untuk mendapatkan data dari port itu.

```
void setup()  
{  
  myPort = new Serial(this, portName, 9600);  
}
```

Ingat bagaimana kita mengatur `Serial.begin (9600)` di Arduino? Nah, jika kita tidak ingin *gobbledy-gook* yang saya bicarakan, kita sebaiknya meletakkan 9600 sebagai argumen terakhir dalam objek

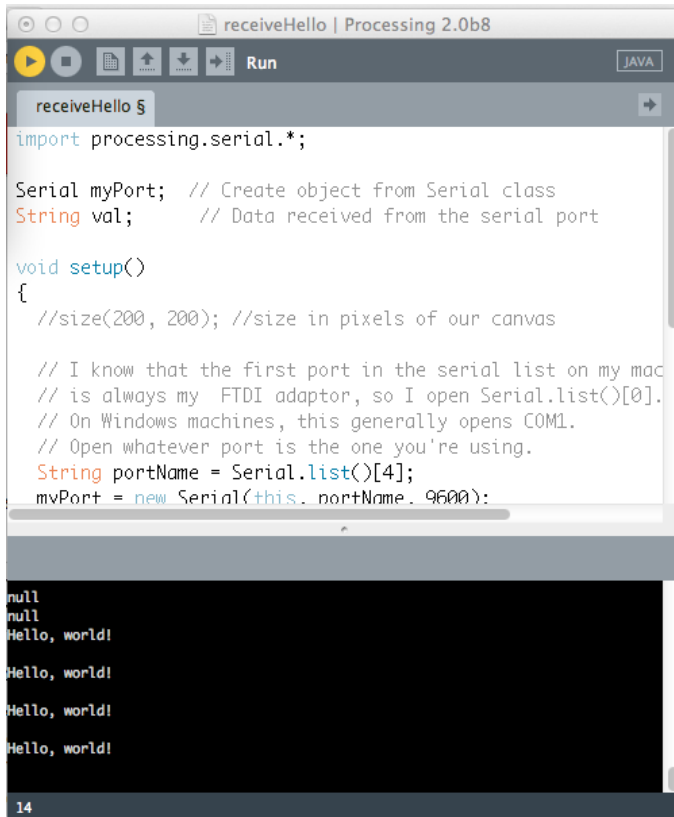
Serial kami di Processing juga. Dengan cara ini Arduino dan Processing berkomunikasi pada tingkat yang sama.

Dalam loop draw (), kita akan mendapatkan data dari port Serial dan kita mendapatkan sesuatu, menempelkan sesuatu di variabel val kita dan mencetaknya ke konsol (area hitam di bagian bawah *sketch* Processing Anda).

```
void draw()
{
    if ( myPort.available() > 0)
    { // If data is available,
      val = myPort.readStringUntil('\n');
    }
    println(val);
}
```

Jika Anda menekan tombol 'run' (dan Arduino Anda terhubung dengan kode pada halaman sebelumnya dimuat), Anda akan melihat jendela pop-up kecil, dan setelah beberapa detik Anda akan melihat 'Hello, World! ' muncul di konsol Processing. Lagi dan lagi. Seperti pada gambar 5.4.





```
receiveHello | Processing 2.0b8
Run JAVA
receiveHello $
import processing.serial.*;

Serial myPort; // Create object from Serial class
String val;    // Data received from the serial port

void setup()
{
  //size(200, 200); //size in pixels of our canvas

  // I know that the first port in the serial list on my mac
  // is always my FTDI adaptor, so I open Serial.list()[0].
  // On Windows machines, this generally opens COM1.
  // Open whatever port is the one you're using.
  String portName = Serial.list()[4];
  myPort = new Serial(this, portName, 9600);
}

null
null
Hello, world!
Hello, world!
Hello, world!
Hello, world!
14
```

**Gambar 5.4** tampilan konsol Processing setelah menerima data dari Arduino

Kita sekarang telah menaklukkan cara untuk mengirim data dari Arduino ke Processing. Langkah selanjutnya adalah mencari tahu cara sebaliknya yaitu mengirim data dari Processing ke Arduino.

### 5.3 Cara mengirim data dari Processing ke Arduino

Jadi kita telah mengirim data dari Arduino ke Processing, tetapi bagaimana jika kami ingin mengirim data dengan cara lain, yaitu dari Processing ke Arduino?

Mari kita mulai dengan sisi Processing. Ini dimulai seperti *sketch* terakhir kita: kami mengimpor *library* Serial dan mendeklarasikan variabel objek Serial global untuk port kita di bagian atas, dan dalam metode `setup()` kita menemukan port dan menginisialisasi komunikasi Serial pada port tersebut dengan variabel Serial di *baud-rate* 9600. Kita juga akan menggunakan perintah `size()`, untuk memberi kita sedikit jendela untuk mengklik, yang akan memicu *sketch* kita untuk mengirim sesuatu melalui port Serial ke Arduino.

```
import processing.serial.*;

Serial myPort;

void setup()
{
  size(200,200);
  String portName = Serial.list()[0];
  myPort = new Serial(this, portName, 9600);
}
```

Dalam loop `draw ()`, kita bisa mengirim apa pun yang kita inginkan melalui port serial dengan menggunakan metode tulis dari *library* Serial Processing. Untuk *sketch* ini, kita akan mengirim string '1' setiap kali kita mengklik mouse di jendela Processing. Kami juga akan mencetaknya di konsol, hanya untuk memastikan bahwa kita benar-benar telah mengirim sesuatu. Jika tidak mengklik, kita akan mengirim string '0'.

```
void draw()
{
  if (mousePressed == true)
  {
    myPort.write('1');
//Lanjutan kode pada halaman sebelumnya
    println("1");
  } else
  {
    myPort.write('0');
  }
}
```

Tampilan Kode akhir dari step ini terlihat pada gambar 5.5

```
Serial myPort; // Create object from Serial class

void setup()
{
  size(200,200); //make our canvas 200 x 200 pixels big
  String portName = Serial.list()[0]; //change the 0 to a 1 or 2
  myPort = new Serial(this, portName, 9600);
}

void draw() {
  if (mousePressed == true)
  {
    myPort.write('1'); //if we clicked in the window
    println("1"); //send a 1
  } else
  {
    myPort.write('0'); //otherwise
    println("0"); //send a 0
  }
}
```

**Gambar 5.5** tampilan kode akhir untuk mengirimkan data dari processing

Jika Anda menjalankan kode ini, Anda akan melihat sekelompok angka 1 muncul di area konsol setiap kali Anda mengklik mouse. Rapi! Tapi bagaimana kita dapat menerima data ini di Arduino? Dan apa yang bisa kita lakukan dengannya?

### 5.4 Cara menerima data dari Processing di Arduino

Pada bagian ini kita akan mencari data yang di kirim dari Processing, dan jika kita melihatnya, kita akan menyalakan LED pada

pin 13 (pada beberapa Arduino, seperti Uno, pin 13 adalah untuk LED *on board*, jadi Anda tidak perlu menambah LED eksternal untuk melakukan pekerjaan ini).

Di bagian atas *sketch* Arduino, kita membutuhkan dua variabel global; satu untuk menyimpan data yang berasal dari Processing, dan yang lainnya untuk memberi tahu Arduino terkait Pin LED terhubung.

```
char val;  
int ledPin = 13;
```

Selanjutnya, dalam metode `setup()`, kita akan mengatur pin LED ke output, karena kita akan menyalakan LED, dan kita akan memulai komunikasi serial di *baud-rate* 9600.

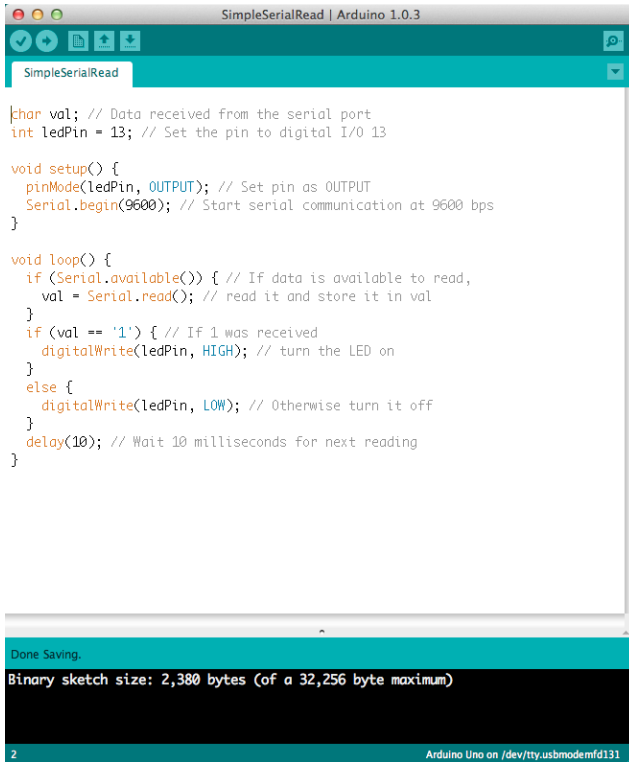
```
void setup()  
{  
    pinMode(ledPin, OUTPUT);  
    Serial.begin(9600);  
}
```

Akhirnya, dalam metode `loop()`, kita akan melihat data serial yang masuk. Jika kita melihat angka '1', maka kita menetapkan LED ke nilai TINGGI (atau aktif), dan jika tidak (misalnya kami melihat '0'), kita mi akan mematikan LED. Pada akhir loop, kita melakukan

penundaan kecil untuk membantu Arduino mengikuti aliran data serial.

```
void loop() {
  if (Serial.available())
  {
    val = Serial.read();
  }
  if (val == '1')
  {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
  //Lanjutan kode pada halaman sebelumnya
  }
  delay(10); // Wait 10 milliseconds for
next reading
}
```

Tampilan Kode akhir dari step ini terlihat pada gambar 5.6



```
SimpleSerialRead | Arduino 1.0.3

SimpleSerialRead

char val; // Data received from the serial port
int ledPin = 13; // Set the pin to digital I/O 13

void setup() {
  pinMode(ledPin, OUTPUT); // Set pin as OUTPUT
  Serial.begin(9600); // Start serial communication at 9600 bps
}

void loop() {
  if (Serial.available()) { // If data is available to read,
    val = Serial.read(); // read it and store it in val
  }
  if (val == '1') { // If 1 was received
    digitalWrite(ledPin, HIGH); // turn the LED on
  }
  else {
    digitalWrite(ledPin, LOW); // Otherwise turn it off
  }
  delay(10); // Wait 10 milliseconds for next reading
}

Done Saving.
Binary sketch size: 2,380 bytes (of a 32,256 byte maximum)

2 Arduino Uno on /dev/tty.usbmodemfd131
```

**Gambar 5.6** tampilan kode akhir untuk mengirimkan data dari processing

Voila! Jika kita mengupload kode ini ke Arduino, dan menjalankan *sketch* processing dari halaman sebelumnya, Anda harusnya dapat menyalakan LED yang terpasang pada pin 13 Arduino Anda, cukup dengan mengklik apapun di dalam konsol processing.





# Bab 6: Proyek Visualisasi detak jantung dengan menggunakan interface komputer berbasis Arduino dan Processing

## **Bab 6: Proyek visualisasi detak jantung dengan menggunakan interface komputer berbasis Arduino dan Processing**

### **6.1 Bahan**

Pembuatan sistem pendeteksi respons sinyal detak Jantung ini diperlukan beberapa bahan penelitian atau komponen yang berfungsi untuk merancang rangkaian. Perancangan merupakan suatu tahap yang penting dalam proses realisasi suatu alat. Bahan penelitian yang akan digunakan mencakup komponen-komponen elektronika yang akan dirangkai menjadi suatu alat yang dirancang untuk mampu memvisualisasikan sinyal detak jantung. Berikut bahan atau komponennya:

- a. Arduino uno R3
- b. *Pulse Heart Sensor*
- c. Kabel pelangi
- d. 2 titik Velcro, dan Tali velcro
- e. Aplikasi processing 3
- f. *Support Package Hardware*

Fungsi dari beberapa komponen diatas adalah:

1. **Arduino uno R3** berfungsi sebagai piranti untuk melakukan pemrograman visualisasi sinyal detak jantung.

2. **Pulse Heart Sensor** berfungsi sebagai sensor untuk pendeteksi sinyal detak jantung.
3. **Kabel kode warna 24-inci**, dengan konektor header (laki-laki) untuk memudahkan penanaman sensor kedalam proyek, dan terhubung ke Arduino.
4. **2 titik Velcro dan Tali velcro** ini adalah sisi pengait dan juga berukuran sempurna untuk sensor yang sangat berguna untuk membungkus sensor pulsa di sekitar jari.
5. Perangkat lunak dengan bahasa pemrograman yang digunakan yaitu aplikasi processing untuk menghubungkan koneksi serial dari Arduino.’

## 6.2 Alat

Alat penelitian meliputi perangkat untuk perancangan dan pengujian. Berikut perangkat yang digunakan dalam penelitian, yaitu:

### - Perangkat keras

1. **Laptop/PC/computer** dengan system operasi Microsoft Windows 10, dengan ArduinoIDE yang digunakan untuk memprogram arduino serta mengaplikasikan pada aplikasi processing untuk menampilkan grafik.

2. **Solder dan timah patri** (timah solder) alat ini digunakan untuk menghubungkan dan melepas komponen.

- **Perangkat lunak**

**1. IDE Arduino**

Software ini berperan dalam proses menjalankan program yang akan dimasukan ke Arduino supaya dapat berjalan sebagaimana mestinya.

**2. Processing**

*Software* ini berfungsi untuk membuat dan memvisualisasikan program pada komputer.

### 6.3 Perancangan Sistem

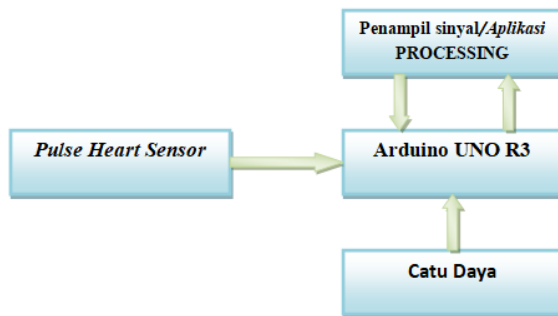
Dalam proses perancangan sistem, kami membagi proses perancangan menjadi dua tahap, yaitu; (1) perancangan perangkat keras dan (2) perancangan perangkat lunak. Untuk menghasilkan sistem yang ideal, perancangan sistem dilakukan dengan mengacu pada beberapa teori didapatkan dari penelitian yang telah dilakukan sebelumnya.

Perancangan alat pendeteksi sinyal detak jantung ini berbasis teknologi yang dimulai dengan membuat sistem Arduino UNO R3.

Selanjutnya pembuatan *software* untuk mengoperasikan sistem untuk menghubungkan koneksi serial ke Arduino UNO R3.

### 6.3.1 Perancangan Perangkat Keras

Perancangan *hardware* dan program pada monitoring visualisasi sinyal detak jantung berbasis Arduino Uno ini dibuat dalam bentuk diagram blok. Untuk lebih jelasnya pada perancangan sistem perangkat keras dapat dilihat pada Gambar 6.1



**Gambar 6.1** Diagram blok rancangan perangkat keras

#### Keterangan diagram blok:

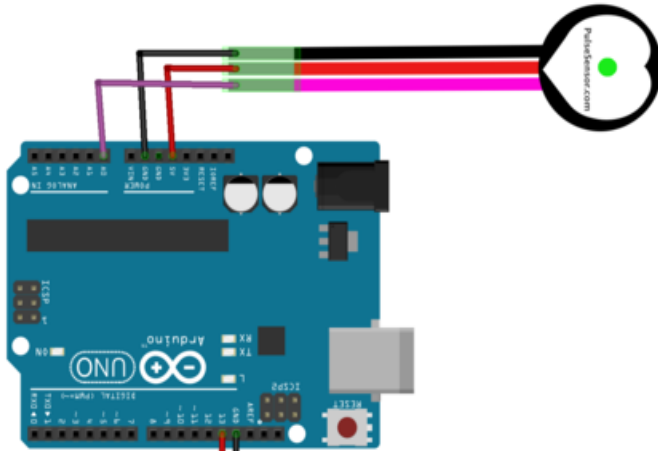
1. *Pulse Heart Sensor* akan mengirimkan sinyal informasi berupa data frekuensi dan pulsa dari detak jantung yang diterima dari jari pengguna.
2. Sinyal tersebut kemudian diproses oleh *microcontroller Arduino Uno* untuk diproses sebagai input data sinyal BPM.

3. *Microcontroller* kemudian mengirim data mentah ke program processing melalui *serial port* untuk diolah kembali untuk menampilkan data dalam bentuk grafik yang ditampilkan pada layar monitor dengan media kabel serial USB.

Aplikasi Processing sebagai bahan untuk mengolah data agar dapat membaca nilai sensor sampai memunculkan grafik.

Diagram blok pada sistem yang dirancang terdiri dari 3 komponen utama yaitu; input, proses dan output. Pada bagian input sistem ini terdiri dari rangkaian Sensor *heart rate* sebagai pendeteksi adanya detak yang diterima dan terintegrasi oleh LED ketika detak jantung terdeteksi. Kemudian data tersebut akan masuk kedalam Arduino Uno R3 untuk dirubah menjadi data digital dan diproses sehingga dapat menampilkan informasi nilai detak jantung dengan tampilan grafik seseorang pada kondisi *real-time* yang dapat dilihat pada layar monitor/*output*.

Berikut rangkaian perangkat keras *Heart Rate Monitor* dapat dilihat pada Gambar 6.2



**Gambar 6.2** Rangkaian Perangkat Keras *Heart Rate Monitor*

### 6.3.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada penelitian ini dibangun menggunakan perangkat lunak yang mendukung pemrograman Arduino Uno yaitu Arduino IDE, setelah itu membuat GUI menggunakan *software processing* yang kemudian dihubungkan ke *software* Arduino Uno.

#### - **Pemrograman Arduino**

Proses pemrograman dengan *software* Arduino IDE terbagi menjadi beberapa bagian dan akan dijelaskan pada sub-bab ini.

## - Listing Program

Pada listing program yang di *upload* ke dalam chipset mikrokontroler Arduino agar data dari *Pulse Sensor* dapat diterima dan dikirimkan ke processing.

### 1. Kode program pembacaan sensor

Pada listing program pembacaan sensor merupakan barisan kode untuk membaca nilai pulsa dari sensor. kode program pembacaan sensor dapat dilihat pada listing 6.1

```
#define USE_ARDUINO_INTERRUPTS true
#include <PulseSensorPlayground.h>
const int OUTPUT_TYPE =
PROCESSING_VISUALIZER;
const int PIN_INPUT = A0;
const int PIN_BLINK = 13;
const int PIN_FADE = 5;
const int THRESHOLD = 550;
PulseSensorPlayground pulseSensor;
void setup() {
    Serial.begin(115200);
```

**Listing 6.1** kode program untuk pembacaan sensor



Penulisan kode program arduino dilakukan untuk menjalankan perintah yang kita inginkan pada perangkat Arduino Uno. Saat *board* Arduino dicatu tegangan dalam hal ini dihubungkan dengan catu daya, kemudian mikrokontroler akan memulai proses inialisasi *input* maupun *output* serta variabel yang dibutuhkan sesuai dengan kode program/perintah yang telah dibuat. Kode program pada bagian `Serial.begin(115200)` digunakan sebagai inialisasi komunikasi serial untuk terhubung dengan *matlab* dan sebagai pengatur kecepatan aliran data menggunakan *baudrate* sebesar 115200 bps. Data yang masuk ke dalam arduino selanjutnya akan diolah dan dikirim melalui *software* matlab untuk ditampilkan sebagai visualisasi.

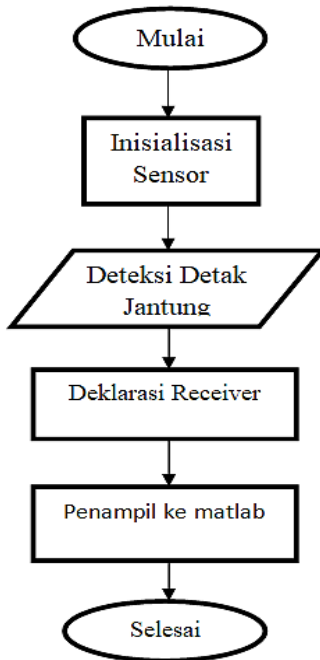
## 2. Kode program pembacaan pada Processing

*Processing* merupakan sebuah *software* yang menggunakan pemrograman untuk merancang sebuah *software* aplikasi yang berfungsi sebagai *receiver* atau penerima data pada visualisasi sinyal detak jantung. *Processing* dibuat dengan tujuan untuk mempermudah pemrograman grafis bersifat interaktif dengan *output* 2D, 3D atau PDF.

|                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> import processing.serial.*;  PFont font;  PFont portsFont;  Scrollbar scaleBar;  Serial port;  int Sensor;  int IBI;  int BPM;  int[ ] RawY;  int[ ] ScaledY;  int[ ] rate;  float zoom;  float offset;  color eggshell = color(255, 253, 248);  int heart = 0;  int PulseWindowWidth = 490;  int PulseWindowHeight = 512;  int BPMWindowWidth = 180;  int BPMWindowHeight = 340;  boolean beat = false; </pre> | <pre> String serialPort;  String[] serialPorts = new String[Serial.list().length];  boolean serialPortFound = false;  Radio[] button = new Radio[Serial.list().length*2];  int numPorts = serialPorts.length;  boolean refreshPorts = false;  void setup() {      size(700, 600); // Stage size      frameRate(100);      font = loadFont("Arial-BoldMT- 24.vlw");      textFont(font);      textAlign(CENTER);      rectMode(CENTER);      ellipseMode(CENTER);      scaleBar = new Scrollbar (400, 575, 180, 12, 0.5, 1.0);      RawY = new int[PulseWindowWidth      ScaledY = new int[PulseWindowWidth];      rate = new int[BPMWindowWidth];      zoom = 0.75; </pre> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Listing 6.2** Listing program *processing*

Adapun bentuk *flowchart* dapat dilihat pada Gambar 6.3

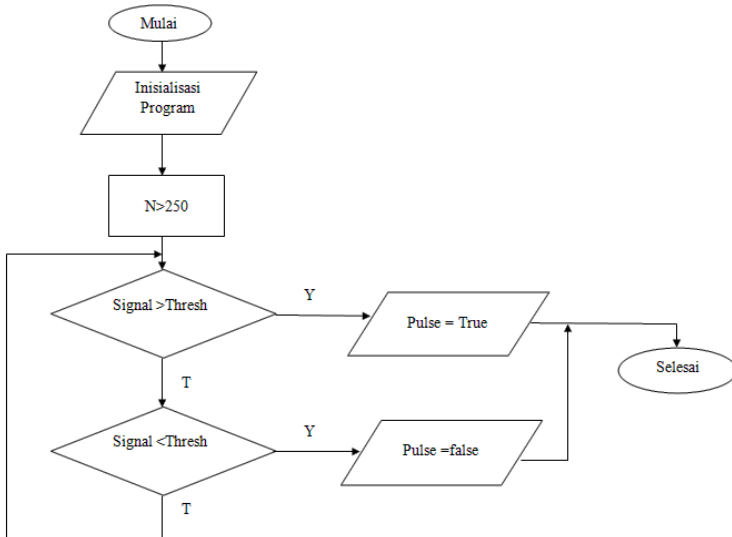


**Gambar 6.3** Diagram alir proses kerja alat secara umum

**Penjelasan :**

Pada saat alat menerima catu daya dari *power supply* maka program akan mulai bekerja dan sensor akan berinisialisasi kemudian data input sensor akan diterima oleh arduino. Kemudian pada waktu yang sama data akan dikirim ke program aplikasi processing setelah *receiver* memberikan perintah.

Berikutnya adalah diagram alir dari *Timer Interrupt* pada program dapat dilihat pada Gambar 6.4



**Gambar 6.4** Flowchart timer interrupt

Proses dari *timer interrupt* terjadi pada fungsi `InterruptSetup()`, dimana fungsinya adalah untuk menginisialisasi/memulai algoritma yang digunakan untuk mendeteksi kondisi dimana jantung berdetak. dari diagram alir yang dapat dilihat pada gambar 6.3, variabel `N` digunakan sebagai variabel yang berfungsi untuk menghitung waktu sejak detak terakhir yang terdeteksi untuk mengurangi gangguan. Nilai dari variabel `N` ditentukan oleh nilai dari variabel *sample*

*Counter* dikurangi dengan nilai dari variabel *last Beat Time*. Ketika nilai variabel *N* lebih besar dari 250 dan jika nilai variabel *signal* lebih besar dari nilai variabel *thresh* maka sensor akan mendeteksi adanya detak jantung. Namun jika nilai variabel *signal* lebih kecil dari nilai variabel *thresh* maka sensor dianggap tidak mendeteksi adanya detak jantung.

## 6.4 Implementasi Sistem

Sistem visualisasi sinyal detak jantung tergambar via PC dibuat untuk membantu pekerjaan dokter ataupun perawat dalam memonitoring sinyal detak jantung pada pasien. Sistem deteksi sinyal detak jantung ini, dirancang menggunakan *Pulse Sensor*. Sensor pada sistem ini mengirimkan sinyal informasi berupa data frekuensi dan pulsa dari detak jantung yang diterima dari jari pasien. Sistem ini menggunakan arduino uno untuk proses pengolahan *input* yang berasal dari *Pulse Sensor*. Berikut *Prototype* pendeteksi sinyal detak jantung dilihat pada gambar 6.5



**Gambar 6.5** Purwarupa pendeteksi sinyal detak jantung

Cara kerja dari sistem ini adalah sensor akan mengirimkan sinyal informasi berupa data frekuensi dan pulsa dari detak jantung yang diterima dari jari pasien ke *microcontroller Arduino Uno*. Sinyal tersebut kemudian diolah oleh mikrokontroler untuk diproses sebagai input data sinyal BPM (*Beat Per Menit*). Setelah data tersebut diproses (di dalam Arduino IDE), kemudian *microcontroller* menghasilkan perintah mengolah input data dari *Pulse Sensor*. *Microcontroller* kemudian akan mengirimkan data mentah ke program processing melalui *serial port* untuk diolah kembali menampilkan data dalam bentuk grafik yang ditampilkan pada layar monitor dengan media kabel serial USB.

Setelah melakukan proses perancangan *software* dan *hardware* selesai, maka tahap selanjutnya adalah melakukan

pengujian kinerja pada sistem yang telah dirancang. Pengujian dilakukan agar bisa mendapatkan data dari sistem tersebut, sehingga menghasilkan alat yang diharapkan.

### 6.5 Hasil dan Pengujian Data

Pada pengujian data alat visualisasi sinyal detak jantung ini dengan cara pengujian perangkat keras dan perangkat lunak. Untuk menguji alat ini di aplikasikan secara langsung menggunakan alat yang telah dirancang. Pengujian alat dapat dilihat gambar 6.6

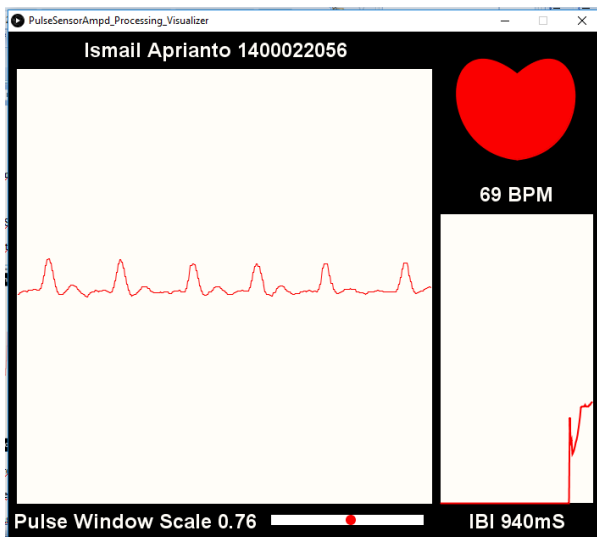


**Gambar 6.6** Gambaran secara keseluruhan

Data yang diperoleh dari tahap diperiksa pasien kemudian di sajikan dalam bentuk tabel data hasil penelitian, proses selanjutnya dilakukan pembahasan berdasarkan data yang telah didapatkan.

- **Pengujian Respon Sensor *Heart rate* Terhadap Detak Jantung**

Pada alat pendeteksi sinyal detak jantung ini pengujian dilakukan dengan menguji kemampuan sensor ketika mendeteksi sinyal pada detak jantung. Berikut hasil pengujian respon sensor ketika mendeteksi denyut jantung dapat dilihat Gambar 6.7



**Gambar 6.7** Pengujian respon sensor heart rate terhadap detak jantung

Pada Gambar 6.7 kondisi gelombang terlihat pada layar monitor yang mendeteksi adanya sinyal detak jantung melalui sensor *heart rate*. Gelombang yang ditampilkan terlihat stabil dikarenakan sensor *heart rate* mendeteksi adanya detak jantung dengan baik. Jika



dilihat dari kondisi nilai *Beat Per Minute (BPM)* saat mendeteksi detak jantung menunjukkan nilai jantung dengan keadaan normal.

- **Pengujian Sensor *Heart Rate* pada Titik Objek**

Pada tahap ini sensor *heart rate* diletakkan pada titik objek yaitu ujung jari dengan tujuan untuk mendapatkan hasil dari detak jantung yang dibaca sensor. Berikut gambar pengujian sensor *heart rate* dengan titik objek pada Gambar 6.8



**Gambar 6.8** Pengujian sensor heart rate pada ujung jari

Pada gambar 6.8 menunjukkan tampilan bahwa pengujian sensor *heart rate* dilakukan dengan meletakkan sensor di salah satu ujung jari yang kemudian dapat diproses oleh arduino untuk mengirimkan sebuah data digital yang kemudian ditampilkan melalui *software processing*.

Setelah melakukan pengujian dalam mendeteksi sinyal detak jantung dengan meletakkan sensor pada ujung jari maka didapatkan hasil berupa nilai detak jantung *beat per menit* (BPM). Hasil yang didapat dapat dilihat pada tabel 6.1.

**Tabel 6.1** Hasil pengujian sensor *heart rate* pada objek ujung jari

| No | Detak Jantung (BPM) | IBI (mS) | Perbandingan dengan <i>stethoscope</i> |
|----|---------------------|----------|----------------------------------------|
| 1  | 63                  | 926      | 62 / menit                             |
| 2  | 63                  | 916      | 62 / menit                             |
| 3  | 65                  | 802      | 63 / menit                             |
| 4  | 67                  | 778      | 65 / menit                             |
| 5  | 68                  | 1064     | 65 / menit                             |

Berdasarkan hasil dari pengujian sensor *heart rate* pada Tabel 6.1 pengujian satu sampai seterusnya terdapat nilai selisih yang tidak jauh berbeda. Kemudian pemeriksaan dengan stetoskop juga diperlukan untuk mencari perbandingan dari alat yang telah dirancang. Cara mendengarkan bunyi jantung dengan stetoskop ialah mendengarkan bunyi yang dihasilkan oleh organ jantung selama satu menit penuh. Untuk hasil rata - rata jumlah detak jantung pada ujung jari dapat dilihat sebagai berikut:

*Rata – rata detak jantung pada ujung jari*

$$= \frac{\text{Jumlah detak jantung pada ujung jari}}{\text{Jumlah data percobaan}}$$

**Maka:**

- Rata-rata detak jantung pada jari  $326/5 = 65.2$  kali

**Kemudian:**

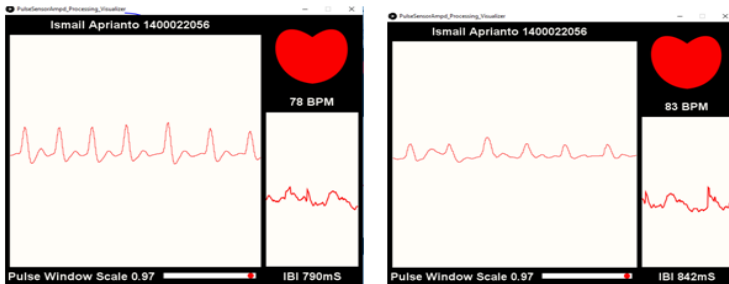
- Rata-rata Perbandingan dengan *stethoscope*  $317/5 = 63.4$  kali

Dari hasil rata - rata jumlah detak jantung pada objek ujung jari dapat disimpulkan bahwa detak jantung dalam setiap menit adalah 65,2 kali jantung berdetak. Kemudian untuk rata - rata dari perhitungan menggunakan stetoskop bahwa detak jantung dalam setiap menit adalah 63,4 kali jantung berdetak. Maka untuk perhitungan alat dengan perhitungan secara manual dengan stetoskop dapat diambil selisih yang tidak jauh berbeda, dan hasil tersebut disimpulkan dengan keadaan yang normal.

- **Pengujian Alat Terhadap Pasien**

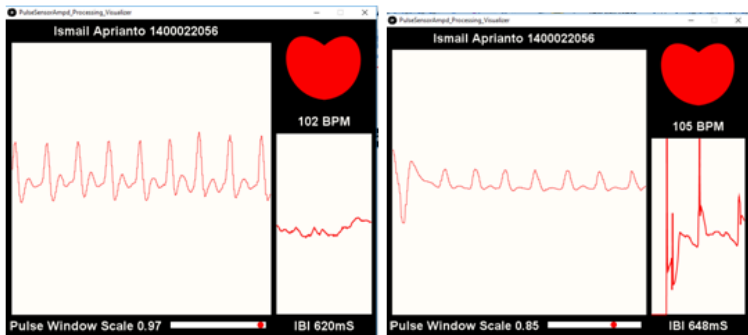
Pengujian ini dilakukan untuk mengetahui detak jantung berdasarkan kondisi seseorang dengan parameter yang berbeda yaitu pada kondisi istirahat dan setelah berolahraga. Untuk kondisi rileks pasien akan diberikan waktu untuk tidak melakukan aktifitas apapun sehingga objek berada dalam kondisi stabil sedangkan untuk kondisi beraktifitas berat yaitu ketika pengujian pasien setelah berolahraga. Untuk mendapatkan hasil yang maksimal pada tahap pengujian ini dilakukan 3 kali percobaan setiap menitnya pada

pasien. Adapun hasil pengujian detak jantung pada orang pertama berdasarkan kondisi istirahat dapat dilihat pada gambar Gambar 6.9



**Gambar 6.9** Pengujian alat detak jantung pada kondisi istirahat

Pada gambar 6.9 merupakan tampilan hasil pengujian alat pada objek dalam kondisi istirahat, dimana dalam percobaan ini dilakukan 3 kali percobaan pada setiap pasien agar mendapatkan hasil yang akurat untuk perbandingan. Jumlah dari detak jantung pada saat diperiksa yaitu 78, 83, dan 84 85 *Beat Per Minute* (BPM) dan *Inter Beat Interva I*(IBI) atau kecepatan detak jantung juga berbeda – beda. Kemudian hasil pengujian detak jantung pada kondisi setelah berolahraga dapat dilihat pada gambar 6.10.



**Gambar 6.10** Pengujian alat detak jantung pada kondisi setelah berolahraga

Pada gambar 6.10 merupakan tampilan hasil pengujian alat pada objek dalam kondisi setelah melakukan aktivitas olahraga. Jumlah dari hasil detak jantung setelah selesai melakukan olahraga adalah 102, 105, 96 *Beat Per Minute* (BPM) dan *Inter Beat Interval* (IBI) atau kecepatan detak jantung juga berbeda – beda.

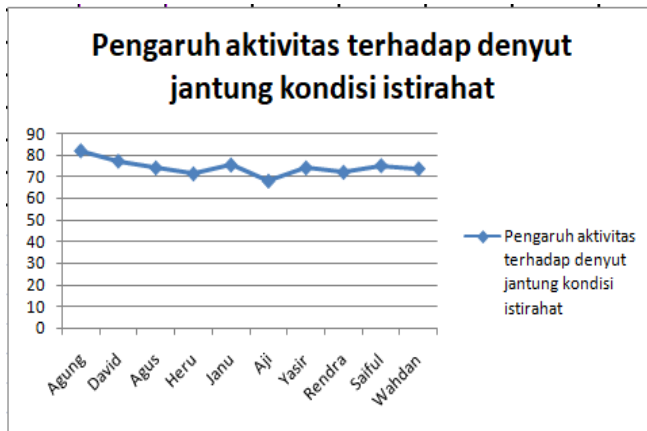
Kemudian setelah melakukan pengujian alat dengan 10 orang dengan kondisi pengujian yang berbeda yaitu pada saat kondisi istirahat dan setelah melakukan aktivitas olahraga, didapat hasil berupa nilai detak jantung pada setiap percobaan. Berikut hasil pengujian yang dapat dilihat pada tabel 6.2.

**Tabel 6.2** Hasil pengujian alat pada kondisi istirahat dan kondisi setelah beraktivitas

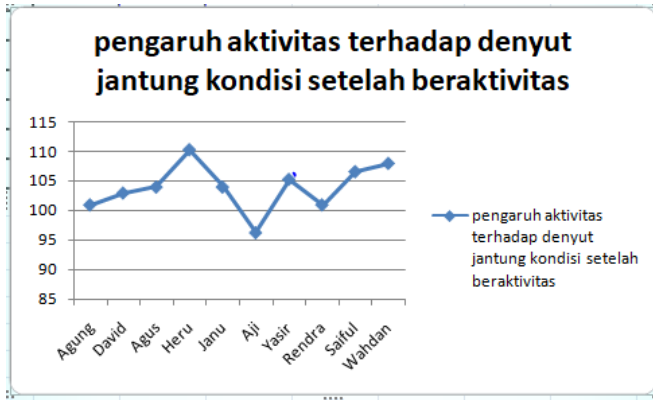
| No  | Nama           | Jumlah denyut jantung |    |    |           |                  |     |     |           |
|-----|----------------|-----------------------|----|----|-----------|------------------|-----|-----|-----------|
|     |                | Waktu istirahat       |    |    |           | Setelah olahraga |     |     |           |
|     |                | 1                     | 2  | 3  | Rata-rata | 1                | 2   | 3   | Rata-rata |
| 1.  | Tri Admojo AL  | 84                    | 83 | 78 | 81.7      | 102              | 105 | 96  | 101.0     |
| 2.  | David Ardi W   | 80                    | 77 | 74 | 77.0      | 109              | 105 | 95  | 103.0     |
| 3.  | Agus Priyatno  | 76                    | 75 | 71 | 74.0      | 113              | 102 | 98  | 104.0     |
| 4.  | Heru Hartanto  | 70                    | 74 | 70 | 71.3      | 109              | 109 | 113 | 110.3     |
| 5.  | Janu Prasetyo  | 76                    | 77 | 73 | 75.3      | 105              | 105 | 102 | 104.0     |
| 6.  | Aji Surya KP   | 63                    | 70 | 71 | 68.0      | 96               | 95  | 98  | 96.3      |
| 7.  | M. Andri Yasir | 72                    | 76 | 74 | 74.0      | 105              | 109 | 102 | 105.3     |
| 8.  | Mey Rendra     | 74                    | 73 | 69 | 72.0      | 96               | 102 | 105 | 101.0     |
| 9.  | Armina SI      | 73                    | 76 | 76 | 75.0      | 98               | 109 | 113 | 106.6     |
| 10. | Wahdan M       | 72                    | 74 | 75 | 73.6      | 109              | 113 | 102 | 108.0     |

Tabel 6.2 pada percobaan ini, jumlah denyut jantung per menit masing-masing pasien dihitung dengan mendapatkan dua perlakuan

berbeda yakni, keadaan normal (istirahat) berkisar antara 60 sampai 80 *Beat PerMinute* (BPM), dan pada keadaan setelah melakukan aktivitas fisik (olahraga) berkisar antara 100 sampai 140 *Beat PerMinute* (BPM). Untuk IBI (*Inter Beat Interval*) sendiri adalah kecepatan detak jantung antara detakan pertama dengan detak jantung selanjutnya. Berdasarkan data hasil percobaan, dapat diketahui bahwa rata - rata jumlah denyut jantung per menit antara sebelum dan sesudah melakukan aktivitas fisik pada masing - masing orang memiliki perbedaan yang signifikan. Perbedaan jumlah denyut jantung dapat dilihat pada grafik pada gambar 6.11 dan 6.12



**Gambar 6.11** Grafik pengaruh aktivitas terhadap denyut jantung kondisi istirahat

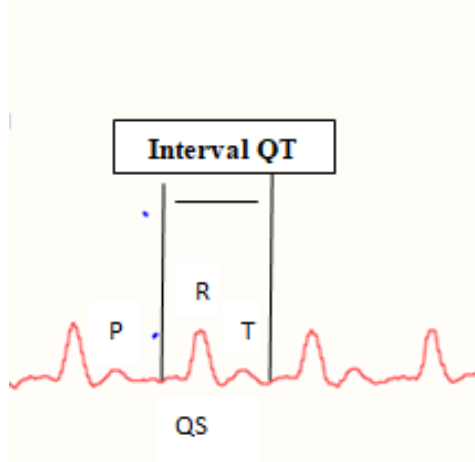


**Gambar 6.12** Grafik pengaruh aktivitas terhadap denyut jantung kondisi setelah beraktivitas

#### - Analisis Data Hasil Pengujian Alat Detak Jantung

Pada tahap ini dilakukan untuk menganalisa data hasil pengujian dengan cara menghitung dari frekuensi karena frekuensi adalah jumlah kompleks gelombang yang muncul dalam 60 detik (1 menit), maka dengan mengetahui jumlah kompleks QRS (*depolarisasi ventrikel*) dalam 6 detik, frekuensi dapat diketahui. Untuk menentukan periode satu gelombang maka syarat utama dari satu gelombang adalah harus memuat puncak-puncak P, Q, R, S dan T.





**Gambar 6.13** Gelombang detak jantung berdasarkan P, Q, R, S dan T

Dari Gambar 6.13, akan diambil data gelombang detak jantung saat kondisi dimana tubuh beristirahat. Sesuai dengan syarat yang digunakan untuk menentukan satu gelombang penuh yang didapatkan dari titik puncak P, Q, R, S dan T. Pada Gambar 6.13 terlihat bahwa setiap memulai satu gelombang, garis gelombang terlihat sedikit naik yang ditandai dengan puncak P, hal ini dikarenakan sensor heart rate telah menerima sinyal detak jantung. Terdapat juga interval PQ yang merupakan perlambatan sinyal. Perlambatan ini memberikan waktu bagi atrium untuk mengosongkan darah yang ada di dalamnya ke dalam ventrikel. Sinyal ini kemudian menuju ke titik Q melalui atrium yang

menyebabkan kedua atrium berkontraksi dan mendorong darah ke ventrikel yang berada di bawahnya. Gelombang R menandai puncak ketika detak jantung berkontraksi dan memompa darah ke paru-paru dan seluruh tubuh. Setelah jantung berkontraksi maka akan ada selang waktu atau waktu tunda untuk beralih ke detak jantung selanjutnya sampai ke titik S. Setelah itu sinyal detak jantung menuju titik T merupakan proses yang menggambarkan ketika ventrikel mengalami repolarisasi. Dari gambaran elektrokardiogram inilah, kita bisa mengetahui normal tidaknya aktivitas listrik jantung. Jika aktivitas listrik jantung tidak normal, ini menunjukkan bahwa jantung mungkin juga tidak normal. Dalam EKG, masing – masing unsur tersebut memiliki nama dan dapat dilihat pada Tabel 6.3.

**Tabel 6.3** Unsur – unsur dalam EKG

|              |                                                                                                                                                                                                                                 |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Gel P        | <ul style="list-style-type: none"> <li>• Gelombang yang tampak pertama</li> <li>• Bentuk melengkung kecil ke atas</li> <li>• Menunjukkan depolarisasi atrium</li> <li>• Kelainan gel P = kelainan atrium</li> </ul>             |
| Interval QRS | <ul style="list-style-type: none"> <li>• Jarak antara gelombang P dan permulaan kompleks QRS</li> <li>• Untuk mengukur waktu perjalanan depolarisasi dari atrium ke ventrikel</li> <li>• <b>Normalnya: 0,12-0,22</b></li> </ul> |

|                                |                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Interval<br/><b>QRS</b></p> | <ul style="list-style-type: none"> <li>• Tiga defleksi yang mengikuti gelombang P</li> <li>• Mengindikasikan depolarisasi (dan kontraksi) ventrikel</li> <li>• Gel Q : defleksi negative pertama setelah P</li> <li>• Gel R : defleksi positif pertama setelah P</li> <li>• Gel S : defleksi negative pertama setelah R</li> <li>• <b>Normalnya kurang dari 0,12 detik</b></li> </ul> |
| <p>Segmen<br/><b>ST</b></p>    | <ul style="list-style-type: none"> <li>• Jarak antara gelombang S dan permulaan gelombang T</li> <li>• Menunjukkan repolarisasi ventrikel</li> </ul>                                                                                                                                                                                                                                  |

*Sumber: Setianto, 2008*



# BAB 7: Aplikasi Mobile Phone Untuk Monitoring Tingkat Kebugaran Seseorang Dengan Algoritma Fast Fourier Transform

## **BAB 7: Aplikasi Mobile Phone Untuk Monitoring Tingkat Kebugaran Seseorang Dengan Algoritma Fast Fourier Transform**

### **7.1 Bahan Penelitian**

Bahan-bahan yang dibutuhkan dalam pembuatan aplikasi untuk memonitoring tingkat kebugaran seseorang dengan algoritma FFT yaitu:

1. **Arduino Uno R3** digunakan sebagai sarana menerima masukan dari sensor kadar oksigen dan kadar oksigen yang berupa pembacaan nilai dari kedua unsur yang ingin diketahui nilainya.
2. **AD8232** digunakan untuk mengukur aktivitas listrik pada jantung.
3. **Handphone Android** digunakan sebagai piranti yang dapat melihat hasil analisa detak jantung.
4. **MATLAB** digunakan untuk memrogram dengan algoritma FFT.
5. **ESP8266-01** digunakan sebagai media Arduino Uno R3 untuk dapat mengakses internet.
6. **Pulse Sensor** digunakan untuk mendeteksi detak jantung.
7. **LCD 16x2** digunakan untuk menampilkan hasil deteksi detak jantung.

## 7.2 Alat Penelitian

Penelitian ini memerlukan beberapa peralatan untuk memonitoring tingkat kebugaran seseorang dengan algoritma FFT yaitu:

1. **ECG** digunakan sebagai analisis ekstraksi ciri atau fitur dari EKG.
2. **Komputer/PC** digunakan untuk simulasi detak jantung yang diperoleh dari *Pulse Sensor* dan sensor AD8232.
3. **Kabel jumper** digunakan untuk menghubungkan pin-pin pada Arduino dengan AD8232.
4. **Perkakas elektronik** digunakan untuk memasang komponen elektronik pada alat yang akan dibuat, perkakas tersebut diantaranya adalah solder, tenol, atraktor, obeng dan lain-lain.
5. **Bor** untuk melubangi PCB dan juga *box* alat.
6. **Tang potong** digunakan untuk memotong kabel.

## 7.3 Perancangan Sistem

Dalam tahap ini, akan diuraikan proses perancangan sistem dari proyek yang dibuat. Dalam merancang sistem ini, kita membagi proses menjadi dua tahap perancangan yaitu perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*). Untuk menghasilkan sistem yang relevan, perancangan sistem

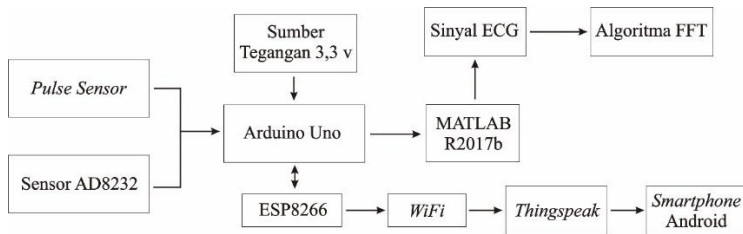
dilakukan dengan mengacu pada teori–teori dan penelitian yang telah dilakukan sebelumnya.

### 7.3.1 Perancangan Perangkat Keras

Perancangan perangkat keras sistem untuk memonitoring tingkat kebugaran seseorang dengan algoritma FFT awalnya dilakukan dengan mendeteksi detak jantung seseorang, pendeteksian detak jantung menggunakan sensor AD8232, yang nantinya sensor tersebut merupakan masukan bagi mikrokontroler yang peneliti gunakan. Pada penelitian ini peneliti menggunakan mikrokontroler ATmega328P yang dikemas dalam bentuk board mikrokontroler yaitu Arduino Uno R3. Arduino Uno R3 dipilih karena board ini memiliki banyak kompatibilitas dengan berbagai jenis sensor dan juga modul lain.

Data yang dikirimkan berupa nilai hasil detak jantung yang diukur menggunakan sensor AD8232 dan akan dikirimkan secara berkala. Data yang sudah masuk dan tersimpan di Arduino Uno kemudian dapat diakses menggunakan aplikasi Android. Android dipilih karena sistem operasi ini merupakan sistem operasi yang familiar dan banyak digunakan oleh penduduk di Indonesia. Blok diagram dari perancangan alat ini ditunjukkan pada Gambar 3.1.



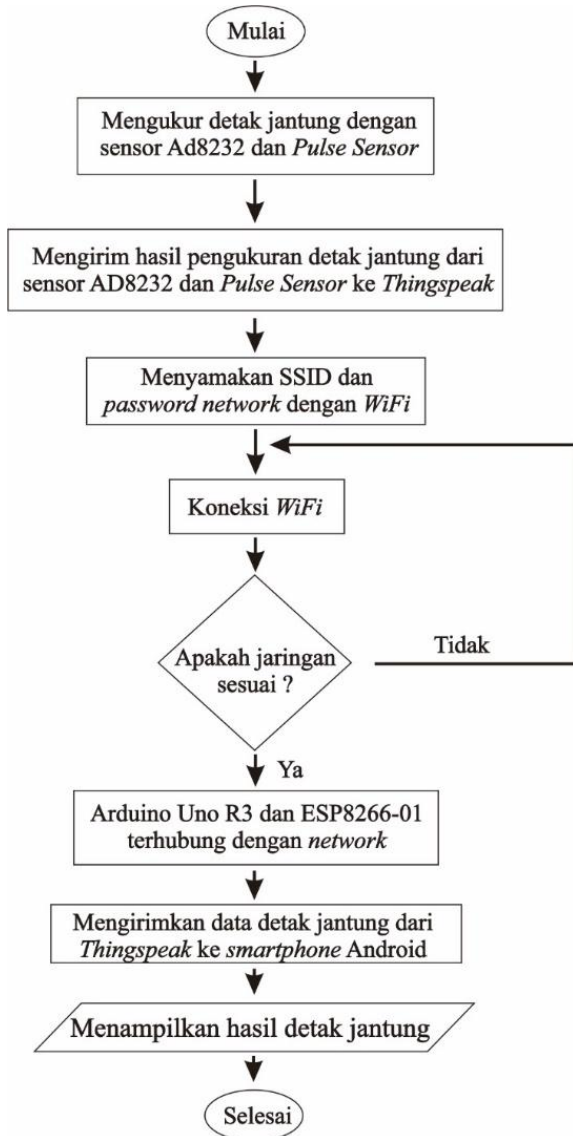


**Gambar 7.1** Blok Diagram Rancangan Perangkat Keras

Pada Gambar 7.1 terlihat bahwa Arduino Uno akan terhubung dengan sumber tegangan. Sensor AD8232 dan Pulse Sensor akan mengirimkan data ke Arduino Uno dan diproses untuk menampilkan grafik detak jantung dan jumlah BPM. Hasil rekaman detak jantung dari sensor AD8232 akan diproses ke MATLAB untuk pengujian algoritma FFT. Sedangkan Pulse Sensor akan mengirimkan jumlah BPM ke Arduino Uno dan Thingspeak yang sudah terhubung dengan jaringan internet. Setelah data diproses, jumlah BPM akan dikirimkan ke aplikasi android yang sudah terhubung dengan modul wifi, dimana aplikasi tersebut akan terlihat seseorang bugar atau tidak.

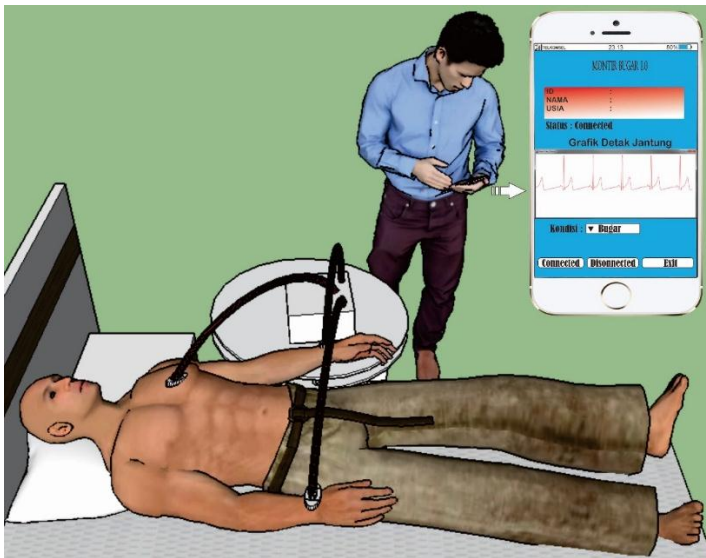
Gambar 7.2 menjelaskan tentang diagram alir dari rancangan perangkat keras tingkat kebugaran seseorang. Pada Gambar 7.2 terlihat bahwa proses pertama yaitu Arduino Uno R3 dihubungkan dengan sumber tegangan, setelah itu sensor AD8232 dihubungkan dengan Arduino Uno R3 dimana sensor AD8232

memiliki 3 kabel konektor yang dipasang di tangan kanan, kiri, dan dada.



**Gambar 7.2** Diagram Alir Rancangan Perangkat Keras

Pembacaan data dari sensor AD8232 dikirimkan ke Arduino Uno R3 yang kemudian setelah diproses di Arduino, data tersebut dikirimkan kembali ke Node MCU melalui sambungan internet yang kemudian diteruskan ke *smartphone* Android. Kemudian desain monitoring tingkat kebugaran seseorang dengan algoritma FFT akan terlihat seperti Gambar 7.3.



**Gambar 7.3** Desain Monitoring Detak Jantung

Pada Gambar 7.3 terlihat seseorang yang sedang berbaring dimana orang tersebut akan diperiksa kesehatannya dengan menempelkan ECG *Electrode* yang sudah terhubung dengan sensor

detak jantung. Hasil dari pemeriksaan tersebut akan terlihat pada aplikasi yang sudah dirancang.

### 7.3.2 Tata Cara Pemeriksaan Detak Jantung

Pemeriksaan detak jantung dapat dilakukan pada pergelangan tangan menggunakan sisi ibu jari. Pergelangan tangan relatif mudah untuk diperiksa dan sering digunakan untuk pemeriksaan. Berikut ini beberapa cara pemeriksaan detak jantung (Anonim, 2018):

1. Cuci tangan pemeriksa.
2. Minta responden untuk melipat baju yang menutupi lengan.
3. Pada posisi duduk, tangan diletakan pada paha dan lengan ekstensi. Pada posisi tidur, kedua lengan ekstensi menghadap ke atas.
4. Letakkan jari telunjuk dan jari tengah pemeriksa pada titik-titik area tubuh yang telah disebutkan sebelumnya.
5. Tekanlah jari telunjuk dan jari tengah pemeriksa pada titik tersebut dengan cukup kuat agar bisa merasakan denyut jantung. Akan tetapi, jangan menekan terlalu keras karena denyut jantung tidak akan terasa.
6. Tetapkan alat yang Anda gunakan sebagai penghitung waktu, biasanya menggunakan stopwatch. Setel waktu selama 30 detik.

7. Mulailah menekan stopwatch sebagai tanda dimulainya penghitungan detak jantung normal.
8. Hitunglah jumlah denyut jantung pasien selama 30 detik tersebut.
9. Catat hasilnya, kemudian kalikan dengan 2 untuk mengetahui jumlah detak jantung pada kondisi normal per menit.

Detak jantung yang terlalu cepat atau lambat merupakan hasil dari kondisi jantung, sehingga dapat mengambil langkah untuk menjalani gaya hidup sehat biasanya akan meningkatkan kesehatan secara keseluruhan.

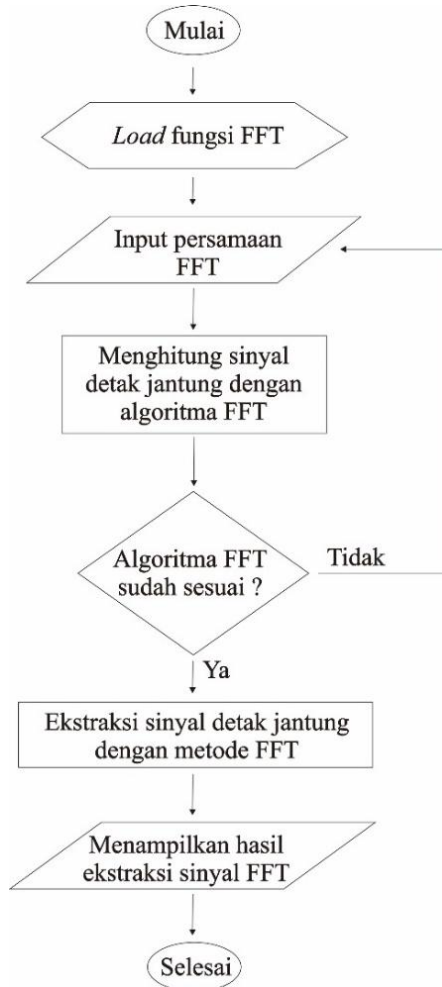
### 7.3.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada penelitian ini dilakukan terbagi menjadi tiga bagian yaitu perancangan untuk program MATLAB, perancangan program untuk Arduino Uno dan juga untuk aplikasi *mobile phone* berbasis Android.

#### - Pemrograman MATLAB

MATLAB atau *Matrix Laboratory* adalah sebuah *software* komputasi numerikal dan bahasa pemrograman komputer generasi keempat (“Equalization - MATLAB & Simulink - MathWorks India,” n.d.). Pada perangkat lunak yang dirancang pada proyek ini, proses

pengolahan data menggunakan metode FFT. Bahasa yang digunakan di dalam MATLAB adalah bahasa C.



**Gambar 7.4** Diagram alir program MATLAB

Pada Matlab 2017b telah tersedia fungsi FFT. Fungsi FFT pada Matlab adalah *tools signal processing* yang disebut SPTool. Untuk memanggil fungsi *signal processing* pada Matlab cukup dengan mengetikkan fungsi “sptool” pada command windows atau jika menggunakan Matlab 2017b dapat diakses dengan memilih option *signal analisys* pada *toolbar* Matlab.

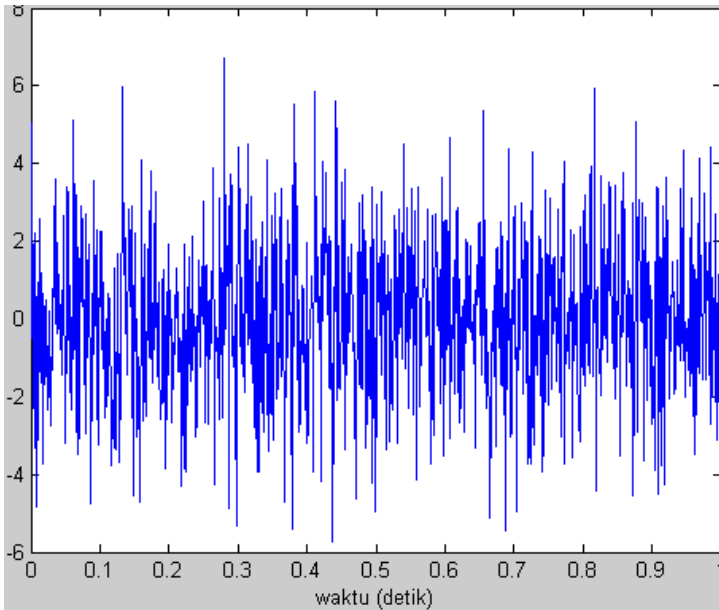
Gambar 7.4 menjelaskan tentang diagram alir dari program MATLAB. Rancangan diawali dengan persiapan data EKG yang berupa interval R-R. Kemudian dilanjutkan dengan proses ekstraksi fitur, pada proses ini dilakukan penentuan fitur dengan melakukan pengolahan sinyal FFT. FFT merupakan teknik cepat dari metode alih bentuk sinyal yang semula analog menjadi diskrit. FFT menggunakan data interval R-R yang kemudian melalui proses transformasi *fourier* diubah menjadi frekuensi. Hasil dari proses transformasi sinyal kemudian diidentifikasi. Hasil dari identifikasi ini merupakan karakteristik fitur yang nantinya akan dilakukan variasi dan pengujian data guna mendapatkan fitur dengan hasil kinerja terbaik.



#### - Implementasi FFT pada Gelombang EKG

proses implementasi algoritma FFT pada gelombang EKG adalah dengan menentukan frekuensi cuplik ( $f_s$ ) dan jumlah cuplikan ( $2N$ ) yang akan dihitung terlebih dahulu. Input berasal dari gelombang EKG yang sudah direkam menggunakan sensor AD8232. Cuplikan gelombang yang akan dihitung disimpan terlebih dahulu dalam *workspace* Matlab. Setelah nilai cuplikan didapatkan maka proses selanjutnya adalah melakukan perhitungan dengan algoritma FFT. Hasil perhitungan tersebut kemudian disimpan dalam *workspace* dan ditampilkan dalam bentuk grafik magnitude pada kawasan frekuensi. Perhitungan dilakukan pada bentuk sinyal dan frekuensi yang sama dengan jumlah cuplikan berbeda-beda dan kemudian dibandingkan dengan hasil dari masing-masing perhitungan.

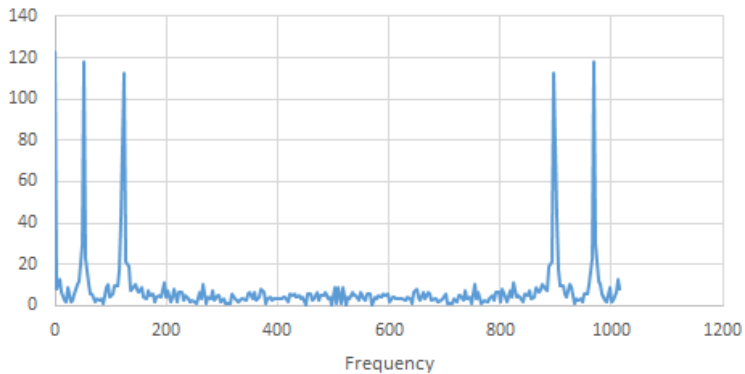
FFT erat kaitannya dengan periode dan frekuensi. Secara umum periode didefinisikan sebagai waktu yang dibutuhkan untuk sebuah isyarat atau gelombang dan mencapai satu gelombang penuh. Sedangkan frekuensi diartikan sebagai jumlah gelombang yang terjadi dalam 1 detik. Periode dan frekuensi merupakan fungsi utama dari algoritma FFT. Gelombang yang memiliki keberagaman periode dan frekuensi dapat dilihat pada Gambar 7.5.



**Gambar 7.5** Gelombang Gabungan dari 4 Isyarat

Pada Gambar 7.5 dapat dilihat bahwa ada empat isyarat yang digabungkan dalam satu gelombang dan memiliki frekuensi yang berbeda-beda. Isyarat yang dihasilkan benar-benar acak. Isyarat yang akan diolah harus bersifat stasioner dan aperiodis (tidak periodis). Isyarat kontinu dapat dianalisa menggunakan deret Fourier. Untuk itu dibutuhkan algoritma FFT untuk memisahkan isyarat tersebut menjadi beberapa komponen frekuensi agar data mentah lebih mudah dikelompokkan berdasarkan perioditasnya. Data yang

diekstraksi oleh algoritma FFT bisa mewakili data asli. Hasil dari ekstraksi ciri dengan algoritma FFT dapat dilihat pada Gambar 3.6.



**Gambar 7.6** Hasil Ekstraksi Ciri dengan FFT

Pada Gambar 7.6 dapat kita lihat bahwa dengan menggunakan algoritma FFT, isyarat yang merupakan hasil penjumlahan 4 buah isyarat yang memiliki frekuensi berbeda-beda dapat dipisahkan dengan baik, sehingga keempat komponen frekuensi bisa terdeteksi dengan baik juga, sedangkan data lain yang menjadi bagian dari keempat komponen frekuensi tersebut akan ditampilkan sebagai spektrum frekuensi dengan nilai yang magnitudenya sangat kecil atau mendekati 0.

Contoh implementasi dari algoritma FFT lainnya antara lain adalah untuk bidang statistik, medis, pengolahan citra, pengolahan suara dan telekomunikasi. Algoritma lainnya yang bernama MFCC

juga menggunakan algoritma FFT atau transformasi fourier sebagai syarat awal untuk memproses sinyal yang akan ditampilkan.

#### - Arduino Uno R3

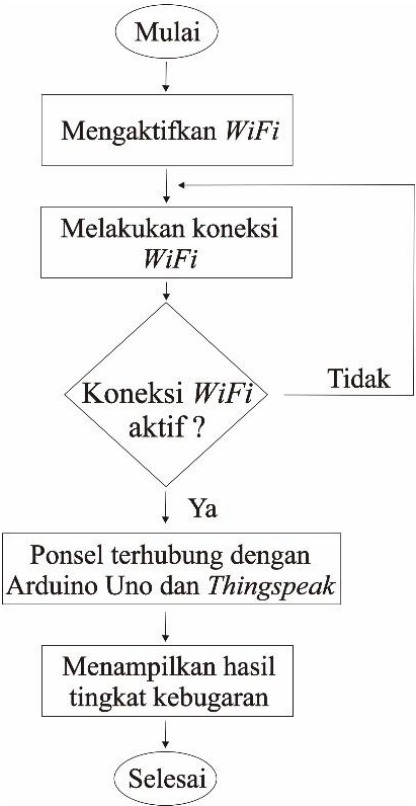
Pada pembuatan program yang akan diunduh kedalam *board* Arduino Uno, peneliti menggunakan software Arduino IDE. Arduino IDE merupakan sebuah *software* pemrograman bawaan dari *board* Arduino Uno. Pada *software* ini inisiasi pengaturan *port* sebagai *input*, *output* dan fitur lainnya seperti PWM dan ADC dilakukan secara langsung dari Arduino IDE (Santoso, 2015).

#### - Pemrograman MIT App Inventor Aplikasi Mobile Phone

App Inventor adalah aplikasi berbasis web *open source* yang pada awalnya dikembangkan oleh perusahaan Google, namun saat ini dikelola oleh *Massachusetts Institute of Technology* (MIT). MIT App Inventor adalah sebuah aplikasi yang banyak digunakan untuk membuat aplikasi untuk *smartphone* dengan sistem operasi Android. App inventor memungkinkan pengguna yang awam terhadap pemrograman computer untuk merancang aplikasi Android secara mudah.

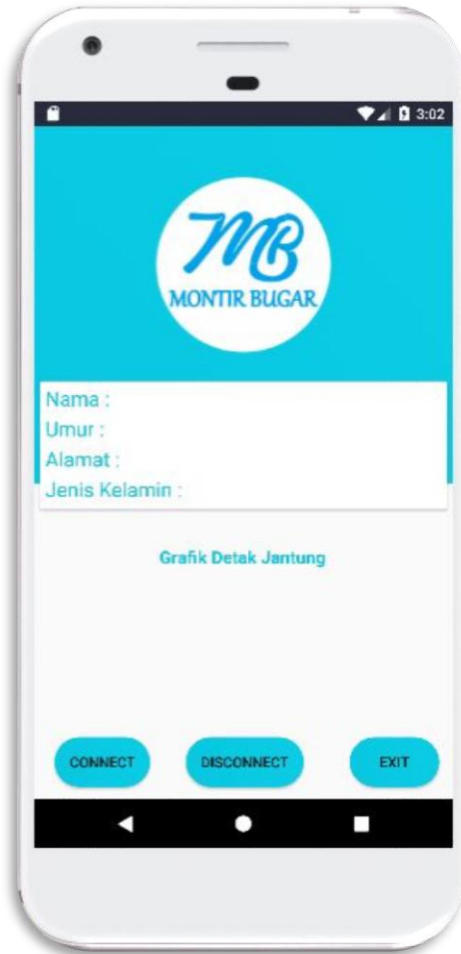
Dalam pembuatan aplikasi, kita akan membuat sebuah program untuk menyusun dan mengatur blok-blok untuk

menghasilkan sebuah aplikasi. Blok-blok yang sudah disusun, ditambahkan program untuk mengkoneksikan android dengan aplikasi *Thingspeak*. Diagram alir dari aplikasi Android yang dirancang dapat dilihat pada Gambar 7.7.



**Gambar 7.7** Diagram alir aplikasi *mobile phone*

Gambar 7.8 menjelaskan tentang desain dari aplikasi Android.



**Gambar 7.8** Desain Aplikasi Android

Pada Gambar 7.8 terlihat bahwa aplikasi Monitoring Tingkat Kebugaran seseorang bisa diakses dan dilihat menggunakan aplikasi

yang sudah dirancang. Sebelum masuk ke aplikasi tersebut, pasien harus mengisi data diri. Setelah data diri terisi, maka akan masuk ke menu utama yaitu data detak jantung, dimana dalam data tersebut akan terlihat kondisi seseorang bugar atau tidak bugar.

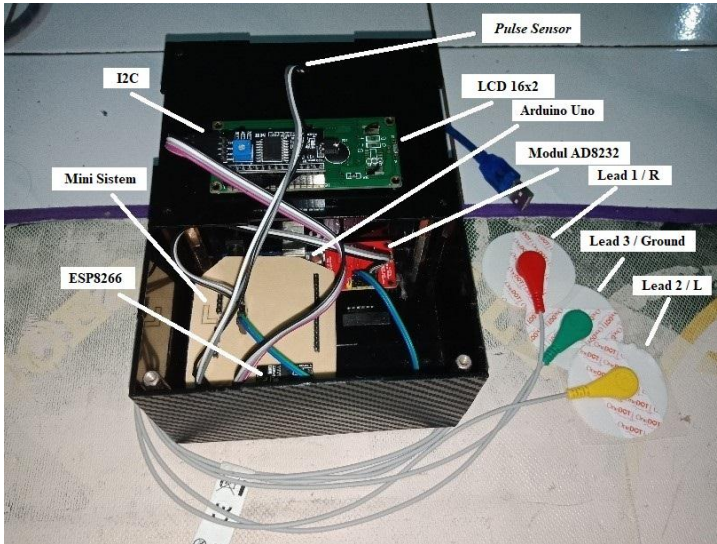
## 7.4 Pengujian Sistem

Setelah alat berhasil dibuat, maka alat tersebut akan diuji coba terlebih dahulu. Pengujian sistem ini dilakukan untuk memastikan bahwa sistem yang telah dibuat sesuai dengan apa yang diharapkan oleh peneliti. Tahapan pengujian sistem dimulai dalam beberapa tahap sebagai berikut:

### 7.4.1 Pengujian Perangkat Keras

Setelah melakukan proses perancangan perangkat keras, didapatkan hasil berupa perangkat keras monitoring tingkat kebugaran seseorang yang siap digunakan. Perangkat keras monitoring tingkat kebugaran dibuat untuk mempermudah seseorang untuk kondisi detak jantung berdasarkan usia dan jenis kelamin. Hasil pengujian tersebut dapat dilihat melalui LCD 16x2, web *DataBase*, dan aplikasi Android yang telah dibuat, sedangkan sinyal FFT di *filter* melalui *software* Matlab R2017b dengan mengambil sampel sinyal detak jantung seseorang dan hasil sampel tersebut disimpan dalam format *file* Excel dengan tipe file *.csv*.

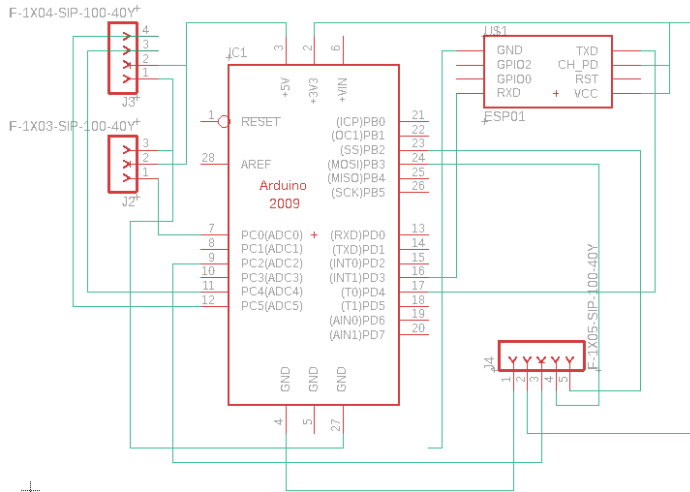
Perangkat keras monitoring tingkat kebugaran seseorang berbasis FFT dapat dilihat pada Gambar 7.9.



**Gambar 7.9** Perangkat keras monitoring tingkat kebugaran seseorang

Pada Gambar 7.9 terlihat bahwa perangkat keras monitoring tingkat kebugaran seseorang berbasis FFT terdiri dari Arduino Uno, modul AD8232, *Pulse Sensor*, ESP8266, dan LCD 16x2 yang telah dirangkai sedemikian rupa. Skeamtik dari modul deteksi tingkat kebugaran dapat dilihat pada Gambar 7.10





**Gambar 7.10** Skematik rangkaian monitoring tingkat kebugaran seseorang

Pengujian terhadap perangkat keras meliputi 3 bagian utama yaitu Arduino Uno, modul AD8232, dan ESP8266. Berikut ini hasil pengujian dari perangkat keras monitoring tingkat kebugaran seseorang:

#### 7.4.1.1 Pengujian Arduino Uno

Arduino Uno merupakan papan pengembangan mikrokontroler yang berbasis chip ATmega328P. Arduino sudah disediakan berbagai port yang dapat digunakan secara langsung melalui pin *header female* yang sudah ada pada papan tersebut. Arduino Uno memiliki

14 pin yang dapat dijadikan *input* maupun *output*, dimana 6 diantaranya dapat digunakan sebagai *output pulse width modulation*, selain itu terdapat juga 6 pin analog input, crystal oscillator 16 MHz, *socket jack power*, kepala ISP, tombol reset dan koneksi USB.

Pengujian Arduino Uno adalah dengan menggunakan software Arduino IDE dengan MATLAB R2017b dengan cara melakukan komunikasi secara serial kedua software tersebut. Komunikasi tersebut dilakukan menggunakan listing 7.1 dan 7.2.

```
void setup() {  
  // initialize the serial communication:  
  Serial.begin(9600);  
  pinMode(10, INPUT); // deteksi LO+ pada modul AD8232  
  pinMode(11, INPUT); // deteksi LO- pada modul AD8232  
}
```

**Listing 7.1** Listing program Arduino IDE

```
%buka komunikasi melalui port COM  
serialPort='COM3'; % Port yang digunakan  
s=serial(serialPort, 'BaudRate', 9600);  
disp('Tutup jendela grafik');  
fopen(s);  
fclose(s);
```

**Listing 7.2** Listing program MATLAB R2017b

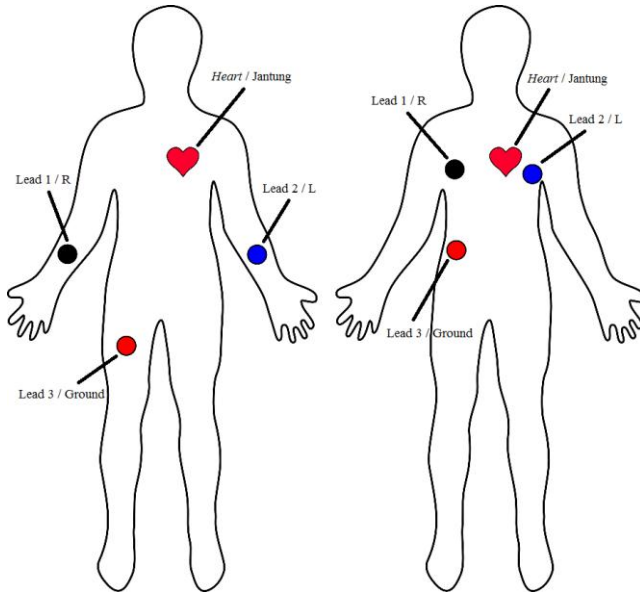
Pada Listing 7.1 dan Listing 7.2 dapat dilihat bahwa komunikasi yang terjadi antara Arduino IDE dengan MATLAB R2017b yaitu

dengan pembacaan PORT yang sama pada Arduino dan *BaudRate* 9600 yang digunakan.

#### **7.4.1.2 Pengujian modul AD8232**

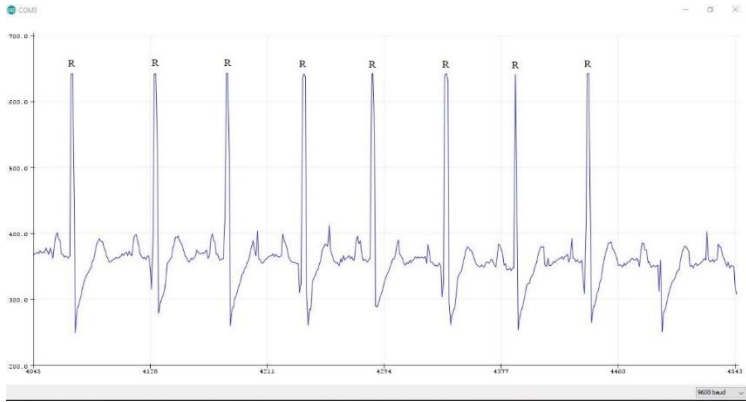
Sensor AD8232 merupakan sebuah modul sensor yang digunakan untuk mengukur aktivitas listrik yang terjadi dalam jantung manusia. Aktivitas listrik ini dapat dipetakan sebagai gelombang ECG dan dapat dibaca sebagai data analog. AD8232 bertindak sebagai op amp atau penguat sinyal untuk membantu mendapatkan sinyal yang jelas dari bagian gelombang PR dan interval QT dengan lebih mudah.

Modul AD8232 yang sudah dikoneksikan dengan board Arduino Uno R3 akan dilakukan pengujian apakah saat diberikan perlakuan tertentu seperti ditempelkan ditangan dan dada. Ada beberapa titik yang bisa ditempelkan ECG *Electrode* seperti pada Gambar 7.11.



**Gambar 7.11** Titik penempelan ECG Electrode

Berdasarkan Gambar 7.11, detak jantung seseorang bisa dideteksi berdasarkan titik penempelan ECG *Electrode* yang sudah dihubungkan dengan modul AD8232. Lampu indikator LED pada modul tersebut akan berdenyut dengan irama jantung berdetak. Irama detak jantung yang diperoleh dari modul AD8232 dapat dilihat pada Gambar 7.12



**Gambar 7.12** Irama Detak Jantung

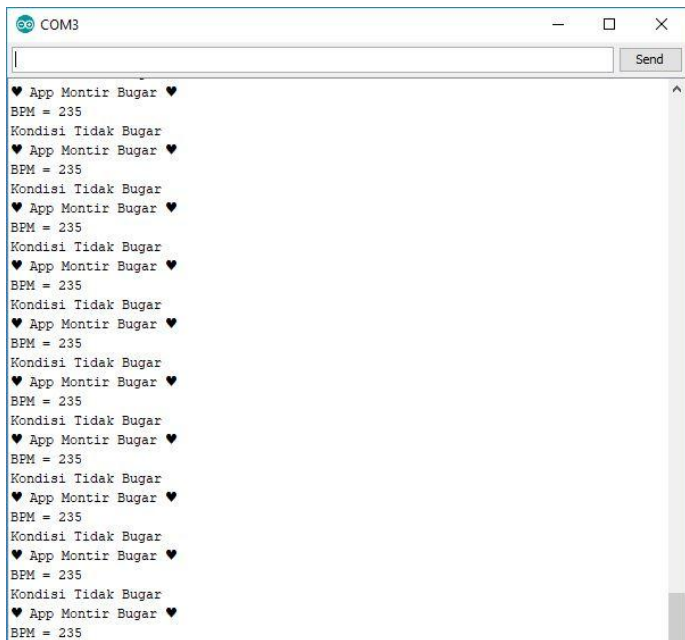
Pada Gambar 7.12 terlihat bahwa interval R–R relatif konstan dari detak ke detak dengan kondisi bugar. Kondisi detak jantung terlihat normal karena gelombang PQRST sesuai dengan parameter sinyal EKG.

#### **7.4.1.3** *Pengujian Pulse Sensor*

*Pulse Sensor* adalah sensor denyut jantung yang dirancang untuk Arduino yang dapat mendeteksi setiap denyut jantung dari kulit. Hadirnya *Pulse Sensor* dapat menjadikan kegiatan yang dilakukan sehari-hari terpantau dengan baik sesuai dengan yang diharapkan.

*Pulse Sensor* merespon perubahan relatif dalam satuan intensitas cahaya. Jika nilai intensitas cahaya rendah dan konstan,

maka nilai sinyal akan tetap berada di rentang 512 yang merupakan nilai tengah dari ADC 10 bit, dan semakin tinggi intensitas cahaya maka nilai ADC akan lebih besar. Pada sensor ini, cahaya LED hijau yang dipantulkan kembali ke sensor akan berubah pada setiap pulsa. Hasil pengujian *Pulse Sensor* dapat dilihat pada Gambar 7.13



7.13 Hasil Pengujian *Pulse Sensor*

Pada Gambar 7.13 terlihat bahwa banyak cahaya yang dipantulkan sehingga mendapatkan sinyal yang lebih tinggi dan

menampilkan kondisi tidak bugar karena rentang nilai BPM yang diperoleh lebih dari nilai maksimal kondisi kebugaran.

#### 7.4.1.4 Pengujian ESP8266

*Espressif Systems Smart Connectivity Platform (ESCP)* adalah perangkat yang dapat bekerja dengan kemampuan tinggi, jaringan Nirkabel SOCs dengan integrasi tinggi dirancang untuk kebutuhan *mobile* yang dibatasi oleh jarak dan daya. Data dari Arduino IDE ke Thingspeak dikirim melalui modul ESP8266 yang sudah diaktifkan *WiFi*-nya melalui Arduino IDE. Hasil pengujian pengiriman data ke Thingspeak dapat dilihat pada Tabel 7.1.

**Tabel 7.1** Hasil Pengujian Pengiriman Data ke Thingspeak

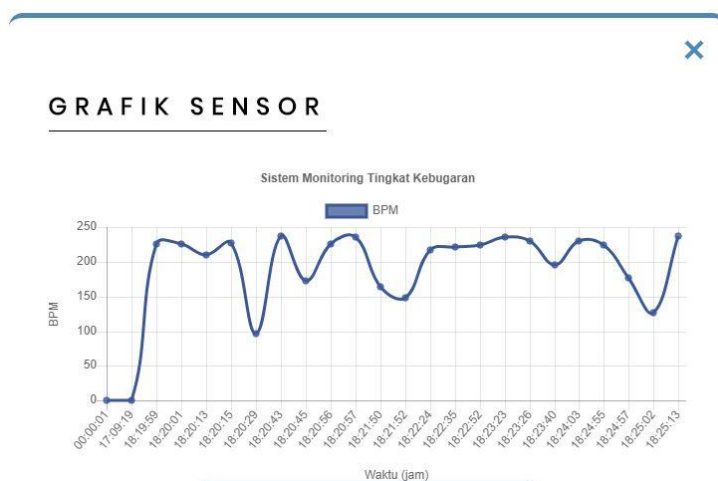
| No. | Tanggal    | Jam      | BPM | Keterangan   |
|-----|------------|----------|-----|--------------|
| 1   | 01/05/2019 | 18.19.59 | 226 | Tidak Bugar  |
| 2   | 01/05/2019 | 18.20.01 | 225 | Tidak Bugar  |
| 3   | 01/05/2019 | 18.20.13 | 210 | Tidak Bugar  |
| 4   | 01/05/2019 | 18.20.15 | 227 | Tidak Bugar  |
| 5   | 01/05/2019 | 18.20.29 | 97  | Kurang Bugar |
| 6   | 01/05/2019 | 18.20.43 | 237 | Tidak Bugar  |

|    |            |          |     |             |
|----|------------|----------|-----|-------------|
| 7  | 01/05/2019 | 18.20.45 | 172 | Tidak Bugar |
| 8  | 01/05/2019 | 18.20.56 | 226 | Tidak Bugar |
| 9  | 01/05/2019 | 18.20.57 | 235 | Tidak Bugar |
| 10 | 01/05/2019 | 18.21.50 | 164 | Tidak Bugar |
| 11 | 01/05/2019 | 18.21.52 | 148 | Tidak Bugar |
| 12 | 01/05/2019 | 18.22.24 | 217 | Tidak Bugar |
| 13 | 01/05/2019 | 18.22.35 | 222 | Tidak Bugar |
| 14 | 01/05/2019 | 18.22.52 | 224 | Tidak Bugar |
| 15 | 01/05/2019 | 18.23.23 | 235 | Tidak Bugar |
| 16 | 01/05/2019 | 18.23.26 | 230 | Tidak Bugar |
| 17 | 01/05/2019 | 18.23.40 | 196 | Tidak Bugar |
| 18 | 01/05/2019 | 18.24.03 | 230 | Tidak Bugar |
| 19 | 01/05/2019 | 18.24.55 | 224 | Tidak Bugar |
| 20 | 01/05/2019 | 18.24.57 | 177 | Tidak Bugar |
| 21 | 01/05/2019 | 18.25.02 | 127 | Tidak Bugar |



|    |            |          |     |             |
|----|------------|----------|-----|-------------|
| 22 | 01/05/2019 | 18.25.13 | 237 | Tidak Bugar |
|----|------------|----------|-----|-------------|

Tabel 7.1 didapatkan data sebanyak 22. Sensor mengirimkan data tersebut melalui modul *WiFi* ESP8266-01. Hasil pengiriman data Arduino IDE ke Thingspeak dapat dilihat pada Gambar7.14



**Gambar 7.14** Pengiriman Data Arduino IDE ke Thingspeak

## 7.4.2 Pengujian Perangkat Lunak

Pada pengujian tahap kedua ini merupakan pengujian yang berkaitan dengan program yang sudah dibuat apakah berjalan dengan baik, pengujian pada perangkat lunak ini terbagi menjadi beberapa yaitu:

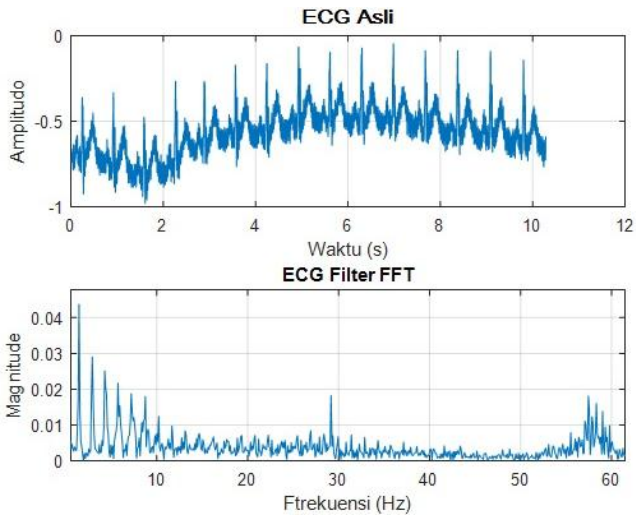
### 7.4.2.1 Pengujian FFT

Algoritma FFT (*Fast Fourier Transform*) adalah sumber dari suatu algoritma untuk menghitung fungsi *Discrete Fourier Transform* atau transformasi fourier diskrit (DFT) secara cepat, efisien dan inversnya. Pengujian FFT dilakukan dengan menggunakan rekaman sinyal EKG yang diambil dari tubuh seseorang. Sinyal tersebut dikirim melalui Arduino IDE ke Matlab R2017b yang kemudian disimpan dalam format Excel. Hasil rekaman tersebut akan di filter menggunakan algoritma FFT. Listing ekstraksi FFT yang digunakan dapat dilihat pada Listing 7.3.

```
Fs=100;
ekg=load('5.txt');
L=length(ekg);
t=(0:L-1)/Fs;
Y = fft(ekg);
f = Fs*(0:(L/2))/L;
P2 = abs(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);
subplot(2,1,2);
plot(f,P1)
xlabel('Frekuensi (Hz)');
ylabel('Magnitude');
title('ECG Filter FFT');
```

**Listing 7.3** Ekstraksi Gelombang EKG dengan FFT

Pada Listing 7.3 kita dapat mengambil sinyal EKG yang sudah direkam dan ditampilkan kembali menggunakan Matlab R2017b. Tampilan sinyal EKG asli dan sinyal ekstraksi dengan FFT dapat dilihat pada Gambar 7.15.



**Gambar 7.15** Filter sinyal EKG dengan frekuensi sampling 100 Hz

Pada Gambar 7.15 adalah pola dari bentuk gelombang jantung yang asli dan telah diekstraksi dengan algoritma FFT. Hasil dari ekstraksi menggambarkan suatu pola yang lebih sederhana dibandingkan dengan bentuk gelombang aslinya. Algoritma FFT digunakan untuk memisahkan isyarat dari gelombang EKG menjadi komponen penyusun frekuensi. Sedangkan data lain akan

ditampilkan sebagai spektrum frekuensi yang nilai magnitudenya sangat kecil atau mendekati nol.

#### 7.4.2.2 Pengujian Aplikasi Android



**Gambar 7.16** Tampilan panel Isi Data Diri

MIT App Inventor merupakan sebuah aplikasi untuk membuat aplikasi untuk ponsel yang menggunakan sistem operasi android. Dalam pembuatan aplikasi, peneliti membuat sebuah program untuk menyusun dan mengatur blok-blok untuk menghasilkan sebuah

aplikasi. Tampilan utama aplikasi android yang dibuat dapat dilihat pada Gambar 7.16.

Pada Gambar 7.16 terlihat bahwa tampilan isi data diri aplikasi menampilkan biodata seseorang yang dimulai dengan Nama, Jenis Kelamin, Umur, dan Alamat. Tampilan utama dari Aplikasi Montir Bugar dapat dilihat pada Gambar 7.17



**Gambar 7.17** Tampilan Utama Aplikasi

Pada Gambar 7.17 menampilkan biodata diri seseorang dan data detak jantung, dimana pada data detak jantung akan

menampilkan tingkat kebugaran seseorang yang dikirim melalui *Data Base* dan modul *WiFi ESP8266*. Selain itu, terdapat pula button yang akan digunakan untuk menampilkan form selanjutnya. Hasil Deteksi Tingkat Kebugaran

#### 7.4.2.3 Tampilan Deteksi Tingkat Kebugaran

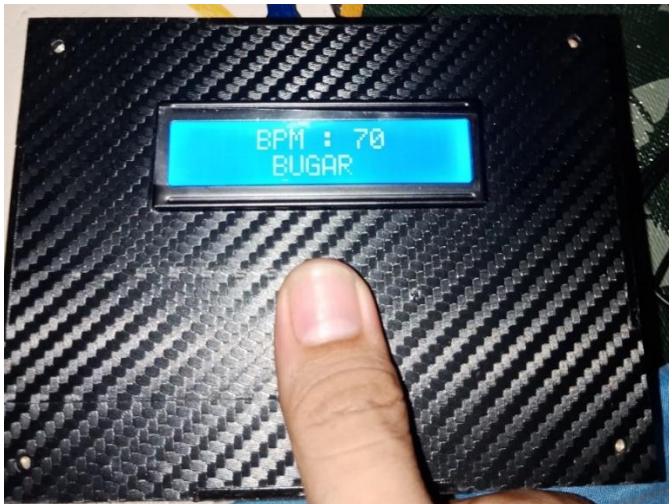
Deteksi tingkat kebugaran menggunakan *Pulse Sensor* yang tampilkan melalui LCD 16x2. Pada tampilan LCD, terdapat empat kategori tingkat kebugaran. Kategori tingkat kebugaran dapat dilihat pada Tabel 7.2.

**Tabel 7.2** Kategori Tingkat Kebugaran

| Kategori  | Umur (Tahun) | Detak Jantung (BPM) | Hasil        |
|-----------|--------------|---------------------|--------------|
| Laki-laki | 18-35        | 49-61               | Sangat Bugar |
|           |              | 62-85               | Bugar        |
|           |              | 86-101              | Kurang Bugar |
|           |              | 102                 | Tidak Bugar  |
| Perempuan | 18-35        | 54-65               | Sangat Bugar |
|           |              | 66-89               | Bugar        |
|           |              | 90-104              | Kurang Bugar |
|           |              | 105                 | Tidak Bugar  |

Tabel 7.2 menjelaskan tentang rentang BPM yang diperoleh melalui kriteria detak jantung laki-laki maupun perempuan (Hardian, 2018). Kebugaran seseorang dapat diketahui melalui aktivitas seseorang serta berolahraga. Selain itu, grafik EKG dengan kategori orang bugar akan stabil pada setiap gelombang R-R.

Nilai BPM bisa dideteksi menggunakan *Pulse Sensor* dengan cara menempelkan jari tangan ke sensor tersebut. Pembacaan nilai BPM berdasarkan kondisi sensor yang disentuh atau tidak. Hasil deteksi tingkat kebugaran yang ditampilkan LCD, dapat dilihat pada Gambar 7.18



**Gambar 7.18** Hasil Deteksi Tingkat Kebugaran

Pada Gambar 7.18 terlihat bahwa deteksi tingkat kebugaran menggunakan *Pulse Sensor* yang ditampilkan melalui LCD 16x2. *Pulse Sensor* menggunakan sinyal Analog dimana sinyal analog tersebut terdiri dari 10 bit (0-1023). *Pulse Sensor* akan memberikan nilai BPM sesuai dengan nilai BPM pada program.

#### 7.4.2.4 Perbandingan *Pulse Sensor* dan Manual

Proses pengambilan data dilakukan menggunakan modul yang dibuat dan secara manual. Pengambilan data menggunakan *Pulse Sensor* menggunakan jari tangan yang ditempel pada sensor. Hasil perbandingan deteksi detak jantung dapat dilihat pada Tabel 7.3.

**Tabel 7.3** Hasil Perbandingan Detak Jantung

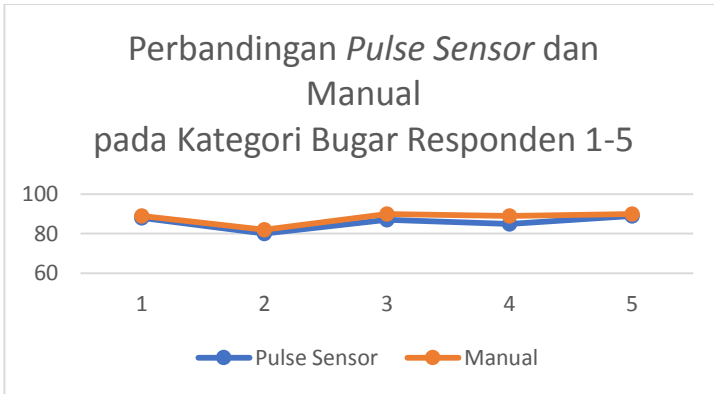
| No | Respon<br>-den<br>ke- | Detak Jantung           |            | Error<br>(%) | Sensitivitas<br>(%) | Spesivitas<br>(%) | Akurasi<br>(%) |
|----|-----------------------|-------------------------|------------|--------------|---------------------|-------------------|----------------|
|    |                       | <i>Pulse<br/>Sensor</i> | Manua<br>l |              |                     |                   |                |
| 1. | 1                     | 88                      | 89         | 1,12         | 98,88               | 98,89             | 98,88          |
| 2. | 2                     | 80                      | 82         | 4,49         | 97,56               | 98,80             | 98,18          |
| 3. | 3                     | 87                      | 90         | 2,43         | 96,67               | 98,90             | 97,79          |
| 4. | 4                     | 85                      | 89         | 3,33         | 95,51               | 97,80             | 96,67          |
| 5. | 5                     | 89                      | 90         | 1,11         | 98,89               | 97,83             | 98,35          |
| 6. | 6                     | 111                     | 113        | 1,77         | 98,23               | 99,12             | 98,68          |
| 7. | 7                     | 99                      | 102        | 2,94         | 97,06               | 98,08             | 97,57          |



|                      |    |     |     |           |       |       |       |
|----------------------|----|-----|-----|-----------|-------|-------|-------|
| 8.                   | 8  | 101 | 102 | 0,98      | 99,02 | 99,03 | 99,02 |
| 9.                   | 9  | 102 | 105 | 2,86      | 97,14 | 98,13 | 97,64 |
| 10.                  | 10 | 109 | 112 | 2,94      | 97,32 | 98,25 | 97,79 |
| <b>Rata-rata (%)</b> |    |     |     | 2,39<br>7 | 97,63 | 98,48 | 98,06 |

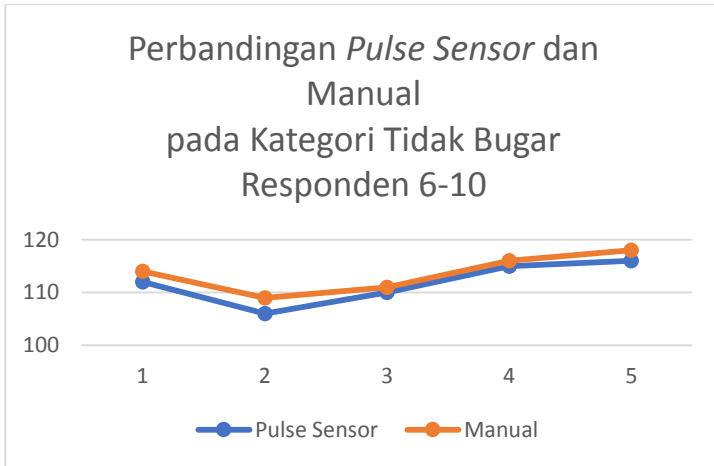
Dari Tabel 7.3 dilakukan pengambilan data dengan mengambil 10 responden dengan menggunakan dua kategori yaitu rajin berolahraga dan tidak. Rata-rata nilai *error* dari perbandingan tersebut sebesar 2,397%, rata-rata nilai sensitivitas adalah 97,63%, rata-rata nilai spesifitas adalah 98,48%, dan rata-rata nilai akurasi adalah 98,06%.

Perbandingan dari sistem yang dibuat dengan perhitungan secara manual dilakukan untuk menentukan tingkat akurasi dari sistem yang sudah dibuat. Perbandingan dari sistem dan manual dibagi menjadi 2 kategori yaitu kondisi bugar dan tidak bugar. Grafik perbandingan *Pulse Sensor* dan manual dengan kategori bugar dapat dilihat pada Gambar 7.19.



**Gambar 7.19** Grafik Perbandingan *Pulse Sensor* dan Manual

Pada Gambar 7.19 didapatkan nilai BPM diatas rata-rata dengan kondisi bugar, hal tersebut terjadi karena responden yang dideteksi detak jantungnya rajin berolahraga sehingga diperoleh kondisi tersebut. Grafik perbandingan *Pulse Sensor* dan manual dengan kategori tidak bugar dapat dilihat pada Gambar 7.20.



**Gambar 7.20** Grafik Perbandingan *Pulse Sensor* dan Manual

#### 7.4.2.5 Deteksi Detak Jantung dengan Algoritma FFT

Pada proses pengambilan data, ada beberapa responden yang di deteksi untuk melakukan pengambilan data tingkat kebugaran yang dibagi menjadi dua yaitu laki-laki dan perempuan dengan kondisi rajin berolahraga dan tidak.

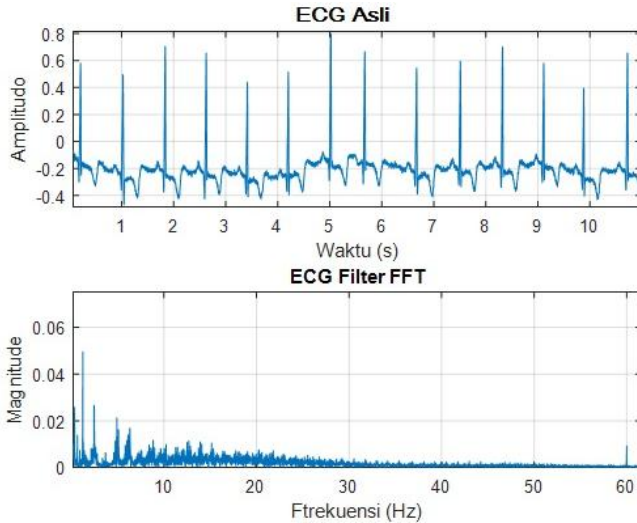
##### - Responden 1

Responden 1 merupakan seorang mahasiswa dari Program Studi Teknik Elektro. Data diri dari responden 1 dapat dilihat pada Tabel 7.4.

**Tabel 7.4** Data Diri Responden 1

|               |                            |
|---------------|----------------------------|
| Nama          | Fahmi Abdul Aziz           |
| Umur          | 21 Tahun                   |
| Jenis Kelamin | Laki-laki                  |
| Alamat        | Jl. Warungboto 3, UH<br>IV |

Data dari responden 1 diambil dengan kondisi berolahraga. Hasil rekaman detak jantung pada responden 1 dapat dilihat pada Gambar 7.21



**Gambar 7.21** Grafik ECG Responden 1

Pada Gambar 7.21 adalah pola dari bentuk gelombang jantung yang asli dan telah diekstraksi dengan algoritma FFT. Hasil dari ekstraksi menggambarkan suatu pola yang lebih sederhana dibandingkan dengan bentuk gelombang aslinya. Algoritma FFT digunakan untuk memisahkan isyarat dari gelombang EKG menjadi komponen penyusun frekuensi. Sedangkan data lain akan ditampilkan sebagai spektrum frekuensi yang nilai magnitudenya sangat kecil atau mendekati nol.

Hasil rekaman detak jantung normal terlihat dari pola-pola gelombang EKG yang dihasilkan karena responden 1 rajin berolahraga sehingga BPM yang diperoleh yaitu 88 BPM sedangkan perhitungan secara manual sebesar 89 BPM. Hasil tersebut diperoleh melalui perhitungan oleh sistem yang telah dibuat dan secara manual, sehingga diperoleh tingkat error sebesar 1,12%, sensitivitas sebesar 98,88%, spesifikasi sebesar 98,89%, dan akurasi sebesar 98,88%.

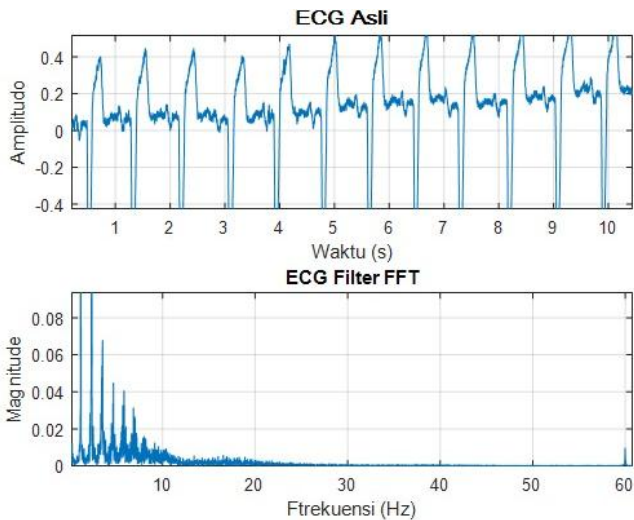
#### - Responden 2

Responden 2 merupakan seorang mahasiswa dari Program Studi Teknik Elektro. Data diri dari responden 2 dapat dilihat pada Tabel 7.5.

**Tabel 7.5** Data Diri Responden 2

|               |                 |
|---------------|-----------------|
| Nama          | M Fazrul Rahman |
| Umur          | 21 Tahun        |
| Jenis Kelamin | Laki-laki       |
| Alamat        | Dompu, NTB      |

Data dari responden 2 diambil dengan kondisi berolahraga. Hasil rekaman detak jantung pada responden 2 dapat dilihat pada Gambar 7.22



**Gambar 7.22** Grafik ECG Responden 2

Pada Gambar 7.22 adalah pola dari bentuk gelombang jantung yang asli dan telah diekstraksi dengan algoritma FFT. Hasil dari ekstraksi menggambarkan suatu pola yang lebih sederhana dibandingkan dengan bentuk gelombang aslinya. Algoritma FFT digunakan untuk memisahkan isyarat dari gelombang EKG menjadi komponen penyusun frekuensi. Sedangkan data lain akan ditampilkan sebagai spektrum frekuensi yang nilai magnitudenya sangat kecil atau mendekati nol.

Hasil rekaman detak jantung normal terlihat dari pola-pola gelombang EKG yang dihasilkan karena responden 2 rajin berolahraga sehingga BPM yang diperoleh yaitu 80 BPM sedangkan perhitungan secara manual sebesar 82 BPM. Hasil tersebut diperoleh melalui perhitungan oleh sistem yang telah dibuat dan secara manual, sehingga diperoleh tingkat error sebesar 4,49%, sensitivitas sebesar 97,56%, spesifikasi sebesar 98,80%, dan akurasi sebesar 98,18%.

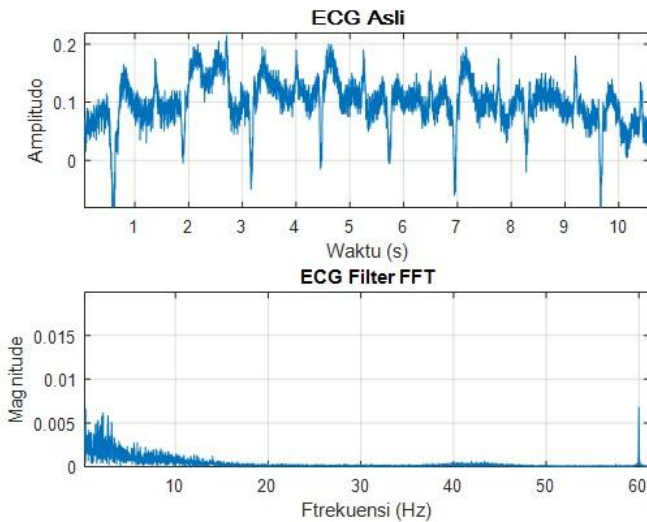
#### - Responden 3

Responden 3 merupakan seorang mahasiswa dari Program Studi Teknik Elektro. Data diri dari responden 3 dapat dilihat pada Tabel 7.6.

**Tabel 7.6** Data Diri Responden 3

|               |                 |
|---------------|-----------------|
| Nama          | Berza Handika S |
| Umur          | 22 Tahun        |
| Jenis Kelamin | Laki-laki       |
| Alamat        | Lampung         |

Data dari responden 3 diambil dengan kondisi berolahraga. Hasil rekaman detak jantung pada responden 3 dapat dilihat pada Gambar 7.23



**Gambar 7.23** Grafik ECG Responden 3



Pada Gambar 7.23 adalah pola dari bentuk gelombang jantung yang asli dan telah diekstraksi dengan algoritma FFT. Hasil dari ekstraksi menggambarkan suatu pola yang lebih sederhana dibandingkan dengan bentuk gelombang aslinya. Algoritma FFT digunakan untuk memisahkan isyarat dari gelombang EKG menjadi komponen penyusun frekuensi. Sedangkan data lain akan ditampilkan sebagai spektrum frekuensi yang nilai magnitudenya sangat kecil atau mendekati nol.

Hasil rekaman detak jantung normal terlihat dari pola-pola gelombang EKG yang dihasilkan karena responden 3 rajin berolahraga sehingga BPM yang diperoleh yaitu 87 BPM sedangkan perhitungan secara manual sebesar 90 BPM. Hasil tersebut diperoleh melalui perhitungan oleh sistem yang telah dibuat dan secara manual, sehingga diperoleh tingkat error sebesar 2,43%, sensitivitas sebesar 96,67%, spesifikasi sebesar 98,90%, dan akurasi sebesar 97,79%.

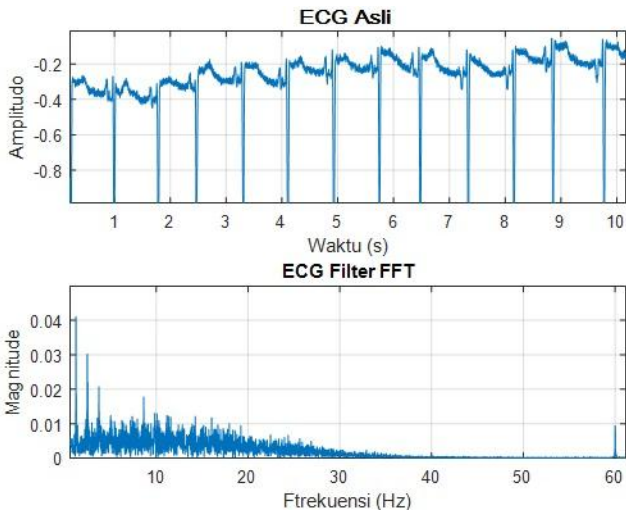
#### - Responden 4

Responden 4 merupakan seorang mahasiswa dari Program Studi Teknik Elektro. Data diri dari responden 4 dapat dilihat pada Tabel 7.7.

**Tabel 7.7** Data Diri Responden 4

|               |           |
|---------------|-----------|
| Nama          | Ikhsan    |
| Umur          | 19 Tahun  |
| Jenis Kelamin | Laki-laki |
| Alamat        | Bangka    |

Data dari responden 4 diambil dengan kondisi berolahraga. Hasil rekaman detak jantung pada responden 4 dapat dilihat pada Gambar 7.24



**Gambar 7.24** Grafik ECG Responden 4

Pada Gambar 7.24 adalah pola dari bentuk gelombang jantung yang asli dan telah diekstraksi dengan algoritma FFT. Hasil dari ekstraksi menggambarkan suatu pola yang lebih sederhana dibandingkan dengan bentuk gelombang aslinya. Algoritma FFT digunakan untuk memisahkan isyarat dari gelombang EKG menjadi komponen penyusun frekuensi. Sedangkan data lain akan ditampilkan sebagai spektrum frekuensi yang nilai magnitudenya sangat kecil atau mendekati nol.

Hasil rekaman detak jantung normal terlihat dari pola-pola gelombang EKG yang dihasilkan karena responden 4 rajin berolahraga sehingga BPM yang diperoleh yaitu 85 BPM sedangkan perhitungan secara manual sebesar 89 BPM. Hasil tersebut diperoleh melalui perhitungan oleh sistem yang telah dibuat dan secara manual, sehingga diperoleh tingkat error sebesar 3,33%, sensitivitas sebesar 95,51%, spesifikasi sebesar 97,80%, dan akurasi sebesar 96,67%.

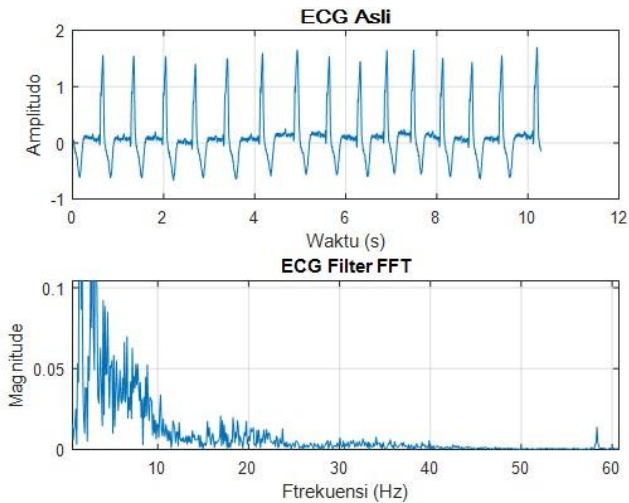
#### - Responden 5

Responden 5 merupakan seorang mahasiswa dari Program Studi Teknik Industri. Data diri dari responden 5 dapat dilihat pada Tabel 7.8.

**Tabel 7.8** Data Diri Responden 5

|               |           |
|---------------|-----------|
| Nama          | Tama      |
| Umur          | 19 Tahun  |
| Jenis Kelamin | Laki-laki |
| Alamat        | Medan     |

Data dari responden 5 diambil dengan kondisi berolahraga. Hasil rekaman detak jantung pada responden 5 dapat dilihat pada Gambar 7.25



**Gambar 7.25** Grafik ECG Responden 5

Pada Gambar 7.25 adalah pola dari bentuk gelombang jantung yang asli dan telah diekstraksi dengan algoritma FFT. Hasil dari

ekstraksi menggambarkan suatu pola yang lebih sederhana dibandingkan dengan bentuk gelombang aslinya. Algoritma FFT digunakan untuk memisahkan isyarat dari gelombang EKG menjadi komponen penyusun frekuensi. Sedangkan data lain akan ditampilkan sebagai spektrum frekuensi yang nilai magnitudenya sangat kecil atau mendekati nol.

Hasil rekaman detak jantung normal terlihat dari pola-pola gelombang EKG yang dihasilkan karena responden 5 rajin berolahraga sehingga BPM yang diperoleh yaitu 89 BPM sedangkan perhitungan secara manual sebesar 90 BPM. Hasil tersebut diperoleh melalui perhitungan oleh sistem yang telah dibuat dan secara manual, sehingga diperoleh tingkat error sebesar 1,11%, sensitivitas sebesar 98,89%, spesifikasi sebesar 97,83%, dan akurasi sebesar 98,35%.

- Responden 6

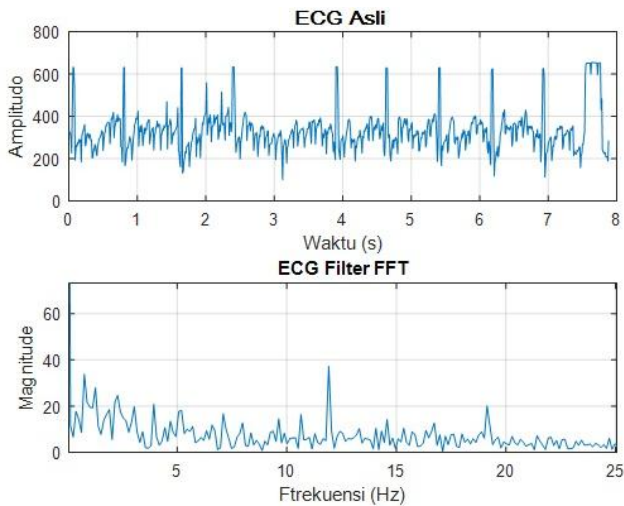
Responden 6 merupakan seorang mahasiswi dari Program Studi Farmasi. Data diri dari responden 6 dapat dilihat pada Tabel 7.9.

**Tabel 7.9** Data Diri Responden 6

|      |                |
|------|----------------|
| Nama | Yeni Rahmawati |
| Umur | 20 Tahun       |

|               |           |
|---------------|-----------|
| Jenis Kelamin | Perempuan |
| Alamat        | Dompu     |

Data dari responden 6 diambil dengan kondisi tidak berolahraga. Hasil rekaman detak jantung pada responden 6 dapat dilihat pada Gambar 7.26.



**Gambar 7.26** Grafik ECG Responden 6

Pada Gambar 7.26 adalah pola dari bentuk gelombang jantung yang asli dan telah diekstraksi dengan algoritma FFT. Hasil dari ekstraksi menggambarkan suatu pola yang lebih sederhana dibandingkan dengan bentuk gelombang aslinya. Algoritma FFT digunakan untuk memisahkan isyarat dari gelombang EKG menjadi

komponen penyusun frekuensi. Sedangkan data lain akan ditampilkan sebagai spektrum frekuensi yang nilai magnitudenya sangat kecil atau mendekati nol.

Hasil rekaman detak jantung tidak normal terlihat dari pola-pola gelombang EKG yang dihasilkan karena responden 6 jarang berolahraga sehingga BPM yang diperoleh yaitu 111 BPM sedangkan perhitungan secara manual sebesar 113 BPM. Hasil tersebut diperoleh melalui perhitungan oleh sistem yang telah dibuat dan secara manual, sehingga diperoleh tingkat error sebesar 1,77%, sensitivitas sebesar 98,23%, spesifikasi sebesar 99,12%, dan akurasi sebesar 98,68%.

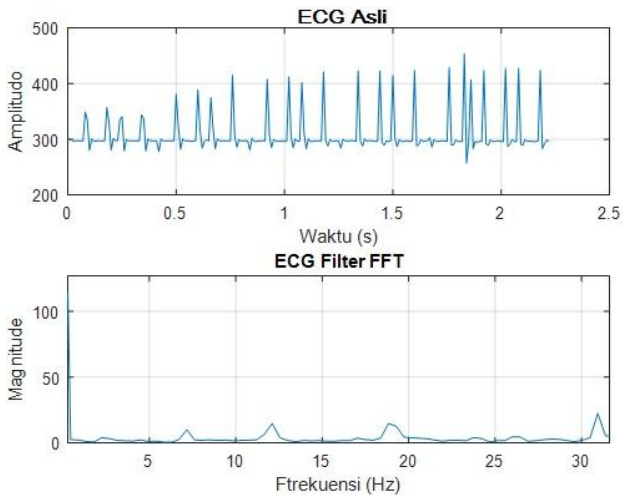
- Responden 7

Responden 7 merupakan seorang mahasiswi dari Program Studi Farmasi. Data diri dari responden 7 dapat dilihat pada Tabel 7.10.

**Tabel 7.10** Data Diri Responden 7

|               |                 |
|---------------|-----------------|
| Nama          | Rooidatun Nahda |
| Umur          | 20 Tahun        |
| Jenis Kelamin | Perempuan       |
| Alamat        | Bengkulu        |

Data dari responden 7 diambil dengan kondisi tidak berolahraga. Hasil rekaman detak jantung pada responden 7 dapat dilihat pada Gambar 7.27.



**Gambar 7.27** Grafik ECG Responden 7

Pada Gambar 7.27 adalah pola dari bentuk gelombang jantung yang asli dan telah diekstraksi dengan algoritma FFT. Hasil dari ekstraksi menggambarkan suatu pola yang lebih sederhana dibandingkan dengan bentuk gelombang aslinya. Algoritma FFT digunakan untuk memisahkan isyarat dari gelombang EKG menjadi komponen penyusun frekuensi. Sedangkan data lain akan



ditampilkan sebagai spektrum frekuensi yang nilai magnitudenya sangat kecil atau mendekati nol.

Hasil rekaman detak jantung tidak normal terlihat dari pola-pola gelombang EKG yang dihasilkan karena responden 7 jarang berolahraga sehingga BPM yang diperoleh yaitu 99 BPM sedangkan perhitungan secara manual sebesar 102 BPM. Hasil tersebut diperoleh melalui perhitungan oleh sistem yang telah dibuat dan secara manual, sehingga diperoleh tingkat error sebesar 2,94%, sensitivitas sebesar 97,06%, spesifikasi sebesar 98,08%, dan akurasi sebesar 97,57%.

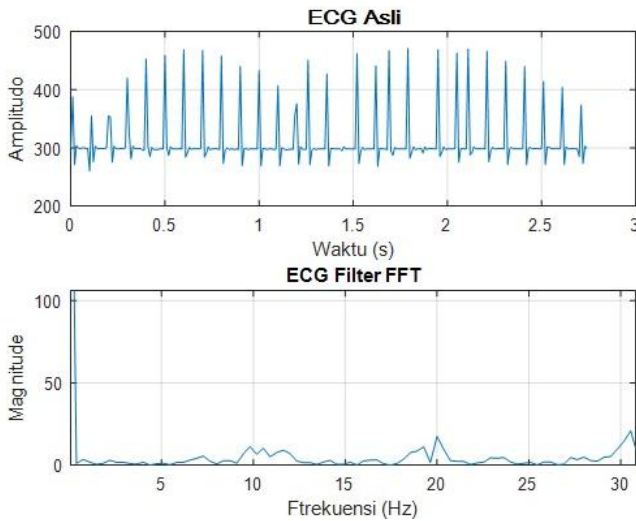
- Responden 8

Responden 8 merupakan seorang mahasiswi dari Program Studi Ilmu Kesehatan Masyarakat. Data diri dari responden 8 dapat dilihat pada Tabel 7.11.

**Tabel 7.11** Data Diri Responden 8

|               |                         |
|---------------|-------------------------|
| Nama          | Teti Sunia A.S          |
| Umur          | 20 Tahun                |
| Jenis Kelamin | Perempuan               |
| Alamat        | Jl. Nitikan, Umbulharjo |

Data dari responden 8 diambil dengan kondisi tidak berolahraga. Hasil rekaman detak jantung pada responden 8 dapat dilihat pada Gambar 7.28



**Gambar 7.28** Grafik ECG Responden 8

Pada Gambar 7.28 adalah pola dari bentuk gelombang jantung yang asli dan telah diekstraksi dengan algoritma FFT. Hasil dari ekstraksi menggambarkan suatu pola yang lebih sederhana dibandingkan dengan bentuk gelombang aslinya. Algoritma FFT digunakan untuk memisahkan isyarat dari gelombang EKG menjadi komponen penyusun frekuensi. Sedangkan data lain akan

ditampilkan sebagai spektrum frekuensi yang nilai magnitudenya sangat kecil atau mendekati nol.

Hasil rekaman detak jantung tidak normal terlihat dari pola-pola gelombang EKG yang dihasilkan karena responden 8 jarang berolahraga sehingga BPM yang diperoleh yaitu 101 BPM sedangkan perhitungan secara manual sebesar 102 BPM. Hasil tersebut diperoleh melalui perhitungan oleh sistem yang telah dibuat dan secara manual, sehingga diperoleh tingkat error sebesar 0,98%, sensitivitas sebesar 99,02%, spesifikasi sebesar 98,13%, dan akurasi sebesar 97,64%.

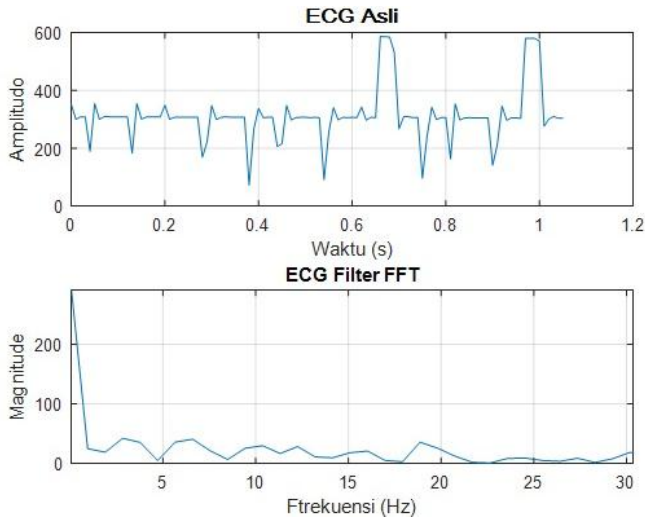
- Responden 9

Responden 9 merupakan seorang mahasiswi dari Program Studi Ekonomi Pembangunan. Data diri dari responden 9 dapat dilihat pada Tabel 7.12.

**Tabel 7.12** Data Diri Responden 9

|               |                 |
|---------------|-----------------|
| Nama          | Ipa Aryanti     |
| Umur          | 19 Tahun        |
| Jenis Kelamin | Perempuan       |
| Alamat        | Jl. Pacar No. 1 |

Data dari responden 9 diambil dengan kondisi tidak berolahraga. Hasil rekaman detak jantung pada responden 9 dapat dilihat pada Gambar 7.29



**Gambar 7.29** Grafik ECG Responden 9

Pada Gambar 7.29 adalah pola dari bentuk gelombang jantung yang asli dan telah diekstraksi dengan algoritma FFT. Hasil dari ekstraksi menggambarkan suatu pola yang lebih sederhana dibandingkan dengan bentuk gelombang aslinya. Algoritma FFT digunakan untuk memisahkan isyarat dari gelombang EKG menjadi komponen penyusun frekuensi. Sedangkan data lain akan

ditampilkan sebagai spektrum frekuensi yang nilai magnitudenya sangat kecil atau mendekati nol.

Hasil rekaman detak jantung tidak normal terlihat dari pola-pola gelombang EKG yang dihasilkan karena responden 9 jarang berolahraga sehingga BPM yang diperoleh yaitu 102 BPM sedangkan perhitungan secara manual sebesar 105 BPM. Hasil tersebut diperoleh melalui perhitungan oleh sistem yang telah dibuat dan secara manual, sehingga diperoleh tingkat error sebesar 2,86%, sensitivitas sebesar 97,14%, spesifikasi sebesar 98,13%, dan akurasi sebesar 97,64%.

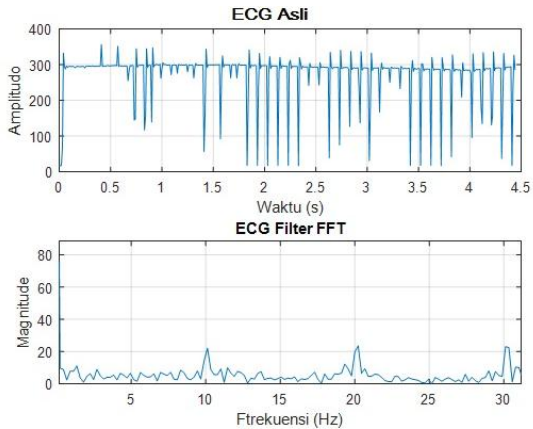
- Responden 10

Responden 10 merupakan seorang mahasiswa dari Program Studi Perbankan Syariah. Data diri dari responden 10 dapat dilihat pada Tabel 7.13.

**Tabel 7.13** Data Diri Responden 10

|               |            |
|---------------|------------|
| Nama          | Tomy       |
| Umur          | 19 Tahun   |
| Jenis Kelamin | Laki-laki  |
| Alamat        | Majalengka |

Data dari responden 10 diambil dengan kondisi tidak berolahraga. Hasil rekaman detak jantung pada responden 10 dapat dilihat pada Gambar 7.30



**Gambar 7.30** Grafik ECG Responden 10

Pada Gambar 7.30 adalah pola dari bentuk gelombang jantung yang asli dan telah diekstraksi dengan algoritma FFT. Hasil dari ekstraksi menggambarkan suatu pola yang lebih sederhana dibandingkan dengan bentuk gelombang aslinya. Algoritma FFT digunakan untuk memisahkan isyarat dari gelombang EKG menjadi komponen penyusun frekuensi. Sedangkan data lain akan ditampilkan sebagai spektrum frekuensi yang nilai magnitudenya sangat kecil atau mendekati nol.

Hasil rekaman detak jantung tidak normal terlihat dari pola-pola gelombang EKG yang dihasilkan karena responden 10 jarang berolahraga sehingga BPM yang diperoleh yaitu 109 BPM sedangkan perhitungan secara manual sebesar 112 BPM. Hasil tersebut diperoleh melalui perhitungan oleh sistem yang telah dibuat dan secara manual, sehingga diperoleh tingkat error sebesar 2,94%, sensitivitas sebesar 97,32%, spesifikasi sebesar 98,25%, dan akurasi sebesar 97,7

# BAB 8: Rancang Bangun Sistem Pemantauan Infus Berbasis Android



## **BAB 8: Rancang Bangun Sistem Pemantauan Infus Berbasis Android**

### **8.1 Bahan Penelitian**

Bahan penelitian yang akan digunakan mencakup komponen–komponen elektronika yang akan dirangkai menjadi suatu alat. Alat yang dirancang untuk mampu menghitung tetesan cairan infus, memantau sisa cairan infus. Bahan–bahan itu antara lain:

- a. Arduino uno R3
- b. Botol Infus
- c. Infus set
- d. Power Supply
- e. *Smartphone* Android
- f. Modul Bluetooth HC-06

### **8.2 Alat Penelitian**

Alat penelitian meliputi perangkat untuk perancangan dan pengujian. Berikut perangkat yang digunakan dalam penelitian, yaitu:

#### **8.2.1 Perangkat keras**

1. **Multimeter:** Alat ini digunakan untuk mengukur tegangan pada alat.

2. **Obeng set:** Obeng berfungsi untuk memasang sekrup supaya sistem dapat di pasang pada alat.
3. **Laptop/PC/computer** dengan system operasi Microsoft Windows 7 Komputer melalui Arduino IDE digunakan untuk memprogram arduino serta mengaplikasikan MIT App Inven-tor untuk membuat aplikasi android
4. **Solder dan tenol:** Alat ini digunakan untuk menghu-bungkan dan melepas komponen

### 8.2.2 Perangkat lunak

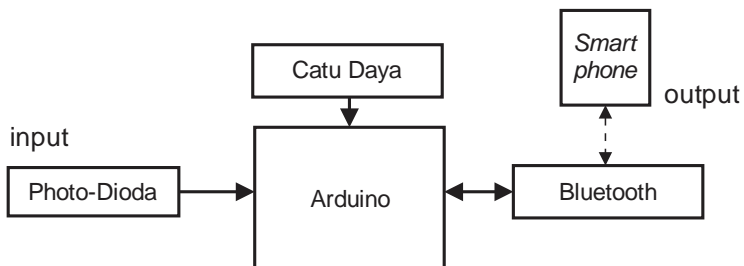
1. **IDE Arduino:** Software ini berperan dalam pembuatan program yang akan dimasukkan ke arduino supaya dapat bekerja sebagaimana mestinya.
2. **MIT App Inventor:** Software ini berfungsi untuk membuat dan mendesain aplikasi pada android.

### 8.3 Perancangan Sistem

Dalam tahap ini, penulis akan melakukan perancangan sistem yang akan dibuat. Dalam merancang sistem ini, penulis merancang system dengan dua tahap perancangan yaitu perancangan perangkat keras dan perancangan perangkat lunak. Untuk menghasilkan sistem yang sesuai, perancangan sistem dilakukan dengan mengacu pada teori – teori dan penelitian sebelumnya.

#### 8.3.1 Perancangan Perangkat Keras

Perancangan *hardware* dan program pada monitoring tetesan infus berbasis Arduino Uno ini dibuat dalam bentuk diagram blok. Diagram blok inilah yang akan membantu pembuatan perancangan sistem secara keseluruhan untuk mempermudah penulis dalam melakukan perancangan. Ditunjukkan pada Gambar 8.1.



**Gambar 8.1** diagram blok rancangan perangkat keras

### **Keterangan diagram blok:**

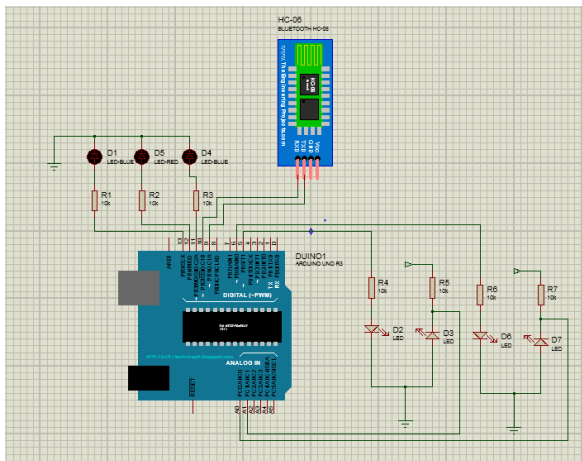
1. catu daya. Sebagai masukan daya untuk arduino uno
2. Mikrokontroler sebagai pengolah data, menggunakan arduino uno R3 Atmel 328.
3. *Bluetooth* sebagai pengirim berupa informasi.
4. Android sebagai penerima atau *receiver* berupa informasi.

Perakitan perangkat keras merupakan proses dalam menghubungkan semua perangkat sehingga dapat membaca nilai sensor sampai memicu kinerja perangkat *output*.

Diagram blok pada sistem ini memiliki 3 bagian utama yaitu, *input*, mikrokontroler, dan *output*. Pada bagian input sistem ini terdiri dari sebuah rangkaian sensor *Photodiode*. Sensor pada sistem ini terletak pada tabung tetes infus yang berfungsi untuk mendeteksi ada tidaknya tetesan. Sensor menerima data dari pancaran cahaya inframerah. Data yang diterima berupa perubahan nilai analog cahaya saat adanya cahaya dan tidak adanya. Selanjutnya data analog tersebut dikonversi menjadi data digital menggunakan fitur ADC pada arduino uno. Data digital tersebut dikirim melalui Bluetooth HC-06 ke *handphone* android dan akan diolah dan digolongkan menjadi data ada tetesan dan tidak ada tetesan. Setelah data adanya tetesan telah didapatkan, selanjutnya data

tersebut ditampilkan pada android dan akan ada peringatan ketika cairan infus akan habis. Perakitan perangkat keras merupakan proses dalam menghubungkan semua perangkat sehingga dapat membaca nilai sensor sampai memicu kinerja perangkat *output*.

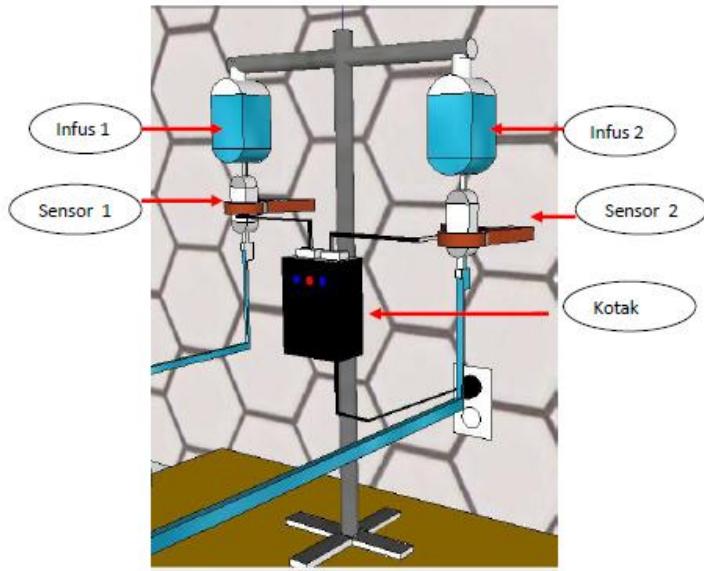
Adapun Rangkaian Sistem Mikrokontroler dapat dilihat pada Gambar 8.2



**Gambar 8.2** Rangkaian Sistem mikrokontroler Arduino Uno

Pada rangkaian di atas sensor photodiode dihubungkan dengan pin analog A0 dan pin analog A1, serta led inframerah dihubungkan dengan pin 5 dan pin 6. Sedangkan untuk led indikator dihubungkan dengan pin 11, pin 12, pin 13 dan modul Bluetooth HC-06 pin Rx dengan pin 10 dan pin Tx dengan pin 9.

Rangkaian system mikrokontroller maka dibuat desain 3D rancang bangun system pemantauan infus. Desain 3D dapat dilihat pada Gambar 8.3.



**Gambar 8.3** Desain 3D Alat Pemantauan Infus

Pada desain 3D tersebut kotak terdiri dari arduino uno, modul bluetooth HC-06 dan 3 led indikator, yang dihubungkan dengan sensor yang terdiri dari photodiode dan led inframerah. Kotak tersebut dipasang pada tiang infus. Sedangkan, Sensor I dan sensor II dipasang pada chamber pada infus I dan infus II.

### 8.3.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada penelitian ini dibangun menggunakan perangkat lunak yang mendukung pemrograman Arduino Uno yaitu Arduino IDE, Setelah itu membuat aplikasi android menggunakan *software* MIT App inventor yang kemudian dihubungkan ke *software* arduino UNO.

#### - Pemrograman Arduino

Pembuatan kode program *software* dilakukan dengan menggunakan aplikasi ide arduino versi 1.8.2 dengan basis bahasa C yang di sesuaikan agar dapat menunjang kinerja alat secara keseluruhan.

#### **a. Kode program**

Pada kode program yang diupload ke mikrokontroler arduino merupakan program untuk menerima data pembacaan sensor photodiode dan mengirim data tersebut ke android

#### **1) Kode program pembacaan sensor**

Pada listing program pembacaan sensor merupakan barisan kode untuk membaca nilai analog cahaya dari sensor. Kode program pembacaan sensor dapat dilihat pada Listing 8.1.

```
void loop() {
digitalWrite(5, HIGH);
digitalWrite(6, HIGH);
digitalWrite(12, HIGH);
sensor=analogRead(A1);
sensor2=analogRead(A0);
if(sensor>=500)
digitalWrite(11, HIGH); // LED 13 dinyalakan
else
digitalWrite(11, LOW); // LED 13 dipadamkan
if(sensor2>=500)
digitalWrite(13, HIGH); // LED 13 dinyalakan
else
digitalWrite(13, LOW); // LED 13
```

**Listing 8.1** Kode program pembacaan sensor

Pada Listing di atas merupakan kode program untuk menginisialisai beberapa pin digital menjadi *output* dan memberikan keluaran nilai logika 1 (High) pada pin arduino uno serta sebagai fungsi utama untuk membaca perubahan nilai analog cahaya pada sensor I dan sensor II. Dan kode program untuk memprogram led indikator pada *hardware*, jika nilai sensor lebih dari 500 maka led indikator akan bernilai 1 (*high*) dan jika kurang dari 500 maka led indikator akan bernilai 0 (*low*).



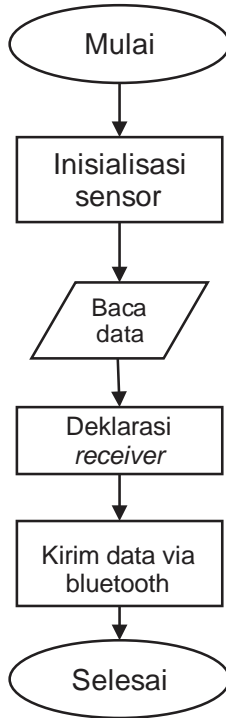
## 2) Kode program komunikasi android

Kode program komunikasi android merupakan kode program yang berfungsi sebagai transmisi data dari alat pemantauan infus ke android. Kode program dapat dilihat pada Listing 8.2.

```
SoftwareSerial bt(10,9);//RX,TX
unsigned long millis();
if(millis()-previousMillis>interval){
previousMillis=millis();
bt.print("");
bt.print(sensor);
bt.print(",");
bt.println(sensor2);
```

**Listing 8.2** Kode program komunikasi android

Pada Listing program SoftwareSerial bt berfungsi untuk menjadikan pin 10 (RX) sebagai penerima dan pin 9 (TX) sebagai pengirim. Listing program pada bt.begin (9600) digunakan sebagai komunikasi serial untuk terhubung dengan *bluetooth* HC-06 dan sebagai kecepatan aliran data menggunakan *baudrate* 9600 bps. Listing program *millis* berfungsi untuk mengeksekusi program dalam waktu *millisecond* untuk melakukan suatu proses selama waktu tertentu. Sedangkan program bt.print berfungsi untuk mengirimkan data pembacaan sensor ke android melalui *bluetooth* HC-06.



**Gambar 8.4** Flowchart program arduino

Pemrograman arduino dilakukan untuk mengeksekusi perintah yang diinginkan pada rangkaian Arduino Uno. Saat *hardware* di masukkan tegangan dalam hal ini dihubungkan dengan catu daya, mikrokontroler akan memulai proses inisialisasi *input* maupun *output* serta variabel yang dibutuhkan. Data yang masuk ke dalam arduino uno selanjutnya di olah dan dikirimkan

melalui *Bluetooth* HC-06. Adapun bentuk *flowchart* dapat dilihat pada Gambar 8.4.

Ketika alat menerima catudaya dari power supply maka program akan memulai program dan sensor akan berinisialisasi. Inisialisasi adalah proses persiapan untuk menentukan nilai awal untuk digunakan. Setelah inisialisasi kemudian data input sensor akan diterima oleh arduino. Pada waktu yang sama maka selanjutnya data akan dikirim ke android setelah pada *receiver* memberikan perintah. Ketika alat telah selesai digunakan maka program akan selesai.

#### - Pemrograman MIT App Inventor

Pada pemrograman *MIT App Inventor* merupakan pemrograman untuk merancang sebuah *software* aplikasi yang berfungsi sebagai *receiver* atau penerima data pada *smartphone* android. Perancangan *MIT App Inventor* menggunakan *mode online* yang berarti setiap pengguna harus terkoneksi dengan jaringan internet yang stabil agar tidak mengganggu proses pembuatan aplikasi dan memiliki akun gmail. *MIT App Inventor* adalah tool pemrograman berbasis blok yang memungkinkan semua orang untuk membuat aplikasi untuk perangkat Android. Menariknya dari *tool* ini berbasis *visual block programming*, jadi dalam penyusunan

pembuatan aplikasi ini menggunakan atau menyusun *drag-drops* suatu *block* seperti menyusun sebuah *puzzle* tanpa menuliskan kode program (*coding*).

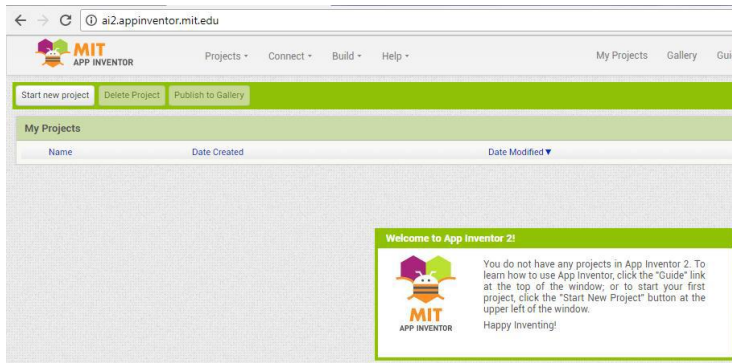
## 1) Mengakses MIT App Inventor

Untuk mengakses MIT App Inventor kita dapat mengakses laman webnya yaitu di <http://appinventor.mit.edu>. Tampilan awal dari software MIT Inventor dapat dilihat pada Gambar 8.5.

Gambar 8.5 merupakan Tampilan awal pada *software* MIT Inventor diwajibkan *login* menggunakan akun *gmail* pada *create apps*. Setelah berhasil masuk menggunakan akun *gmail* maka akan muncul tampilan baru pada MIT Inventor. Tampilan baru pada MIT Inventor dapat dilihat pada Gambar 8.6.



**Gambar 8.5** Tampilan awal pada software MIT Inventor



**Gambar 8.6** Tampilan baru pada MIT Inventor

Gambar 8.6 merupakan tampilan awal pembuatan aplikasi android menggunakan MIT Inventor dan ada beberapa menu yang perlu diketahui fungsinya, antara lain:

- a. **start new project:** digunakan untuk membuat proyek baru pada *MIT App Inventor*
- b. **Delete Project:** digunakan untuk menghapus proyek pada *MIT App Inventor*
- c. **Publish to gallery:** digunakan untuk menmpublish proyek ke galeri pada *MIT App Inventor*
- d. **Project:** adalah menu awal berisi antara lain *start new project*, *save project* dan *delete project*.
- e. **Connect:** adalah menu untuk menghubungkan *project* yang telah dibuat dengan menggunakan media perantara lain. *MIT Inventor Companion* dapat diunduh pada *Playstore*.

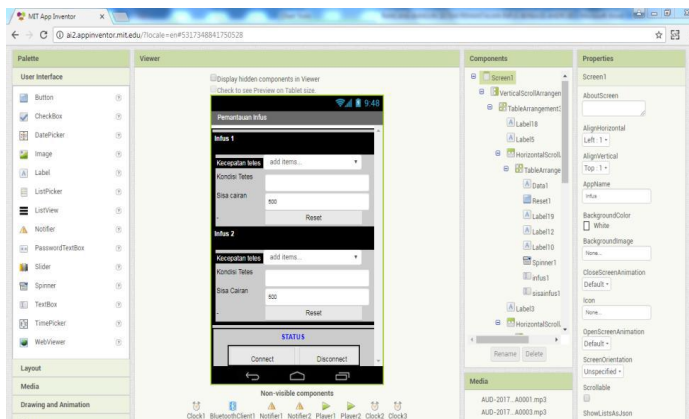
f. **Build:** adalah menu sebagai *download* aplikasi yang telah dibuat. Simpan proyek ke komputer atau melalui *scan barcode* yang akan menuju ke *link download* atau bisa langsung di *download* pada *smartphone*.

## 2) Membuat Aplikasi Android

Dalam membuat aplikasi android pemantauan infus dengan menggunakan MIT App Inventor

### a) Membuat tampilan designer app android

Dapat dilihat tampilan aplikasi android rancang bangun pemantauan infus berbasis android dapat dilihat pada gambar 8.7.



**Gambar 8.7** Tampilan design aplikasi

Gambar 8.7 merupakan desain tampilan aplikasi android menggunakan pemrograman *MIT App Inventor*. Berikut adalah komponen-komponen yang digunakan dalam perancangan aplikasi ini:

### **1. *Listpicker***

Sebuah tombol yang saat diklik menampilkan daftar teks yang bisa dipilih pengguna di antaranya. *Listpicker* pada aplikasi ini digunakan sebagai tombol untuk koneksi dengan *bluetooth*

### **2. *Label***

*Label* adalah komponen yang digunakan untuk menampilkan teks. Label menampilkan teks yang ditentukan oleh properti teks.

### **3. *Button***

*Button* adalah sebuah tombol yang digunakan untuk memberikan sebuah perintah.

### **4. *BluetoothClient***

Digunakan sebagai modul komunikasi dengan menggunakan *bluetooth*

### **5. *Clock***

Merupakan Komponen yang tidak terlihat yang menyediakan waktu bersamaan dengan menggunakan jam internal pada handphone.

### **6. *Notifier***

Komponen *Notifier* berfungsi untuk menampilkan dialog peringatan, pesan, peringatan sementara, dan membuat entri log Android.

### **7. Player**

Komponen Player berfungsi untuk memutar suara dalam bentuk mp3.

### **8. Spinner**

Komponen untuk membuat item pada daftar pilihan.

### **9. Textbox**

Komponen ini berfungsi sebagai penampil teks dalam kotak teks

### **10. Vertical Scroll Arrangement**

Komponen ini adalah elemen pemformatan *layout* untuk menempatkan komponen yang harus ditampilkan dari atas ke bawah

### **11. Tabel Arrangement**

Komponen ini adalah elemen format untuk menempatkan komponen yang harus ditampilkan dalam bentuk table

### **12. Horizontal Scroll Arrangement**

Komponen ini adalah elemen pemformatan *layout* untuk menempatkan komponen yang harus ditampilkan dari kiri ke kanan

## **b) Membuat kode program pada MIT App Inventor**



Setelah pembuatan *designer* aplikasi selesai, kemudian memasukan kode program. Pada MIT App Inventor kode program nya disusun dengan menggunakan *block* seperti *puzzle*. Berikut adalah tampilan kode program dapat dilihat pada gambar 8.8.

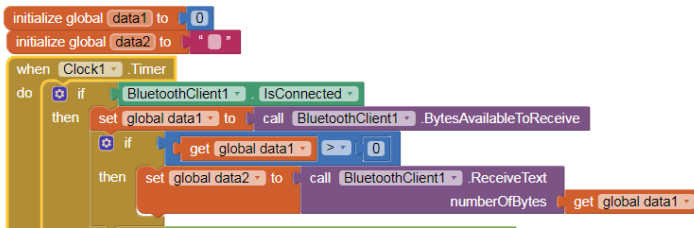
```
when ListPicker1 .BeforePicking
do set ListPicker1 .Elements to BluetoothClient1 .AddressesAndNames

when ListPicker1 .AfterPicking
do set ListPicker1 .Selection to call BluetoothClient1 .Connect
address ListPicker1 .Selection
set Label1 .Text to "Connect"

when Disconnect .Click
do call BluetoothClient1 .Disconnect
set Label1 .Text to "Disconnect"
set Clock1 .TimerEnabled to true
```

**Gambar 8.8** Barisan awal kode blok

Gambar 8.8 merupakan kode program digunakan ketika *bluetooth on* maka ketika diklik akan muncul daftar nama alamat perangkat *Bluetooth* lain sebagai tujuan kirim data dan kemudian dapat memilih *mode* sambungkan maka *bluetooth* akan tersambung. Ketika *mode* tersambung *on* maka tulisan status akan berubah menjadi *connect*. Pada saat ingin mengakhiri koneksi maka klik tombol *disconnect* maka tulisan akan berubah menjadi *disconnect*. Ketika *Bluetooth* sudah terhubung maka data yang diterima akan dibagi dalam variabel, Program dapat dilihat pada Gambar 8.9.



**Gambar 8.9** Kode program penerimaan data Bluetooth

Kode program pada Gambar 8.9 terdapat initialize global data 1 dan data 2, perintah ini berfungsi untuk membuat variabel data 1 dan variabel data 2. Dan program pada clock 1 memperlihatkan kejadian yang ditangani oleh clock 1 dengan perintah jika *Bluetooth* terhubung maka variabel data 1 akan menerima byte yang dikirim oleh modul *Bluetooth* bersamaan dengan itu maka variabel data 2 juga akan menampilkan data yang diterima oleh variabel data 1. Program ini digunakan ketika aplikasi tersebut sudah terhubung dengan hardware melalui komunikasi *bluetooth* maka android akan menerima data dari *hardware*.

```

if (length(trim(get global data2)) == 7)
then
  set Data1 . Text to segment text (get global data2)
  start 1
  length 3
  if (Data1 . Text ≤ 500)
  then
    set infus1 . Text to "Tidak menetes"
  else
    set infus1 . Text to "menetes"
  if (length(trim(get global data2)) == 7)
  then
    set Data2 . Text to segment text (get global data2)
    start 5
    length 3
    if (Data2 . Text ≤ 500)
    then
      set infus2 . Text to "Tidak menetes"
    else
      set infus2 . Text to "menetes"
  
```

Gambar 8.10 Kode blok pemrosesan

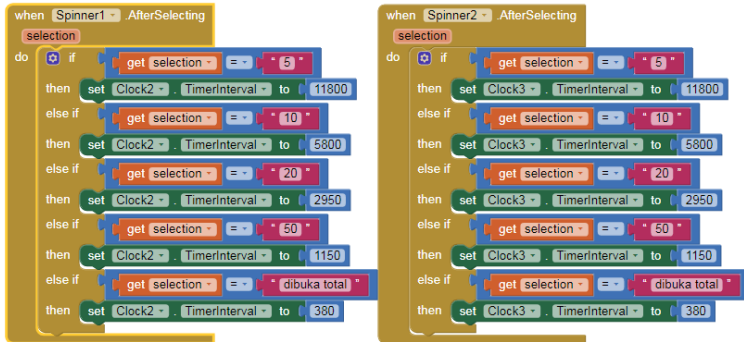
```

when Clock2 . Timer
do
  if (infus1 . Text == "menetes")
  then
    set sisainfus1 . Text to (sisainfus1 . Text - 0.067)
    if (sisainfus1 . Text == 50.095)
    then
      call Player1 . Start
      call Notifier1 . ShowMessageDialog
      message "Infus 1 Akan Segera habis. segera lakukan pengga..."
      title "Peringatan Infus 1"
      buttonText "Keluar"
  when Clock3 . Timer
do
  if (infus2 . Text == "menetes")
  then
    set sisainfus2 . Text to (sisainfus2 . Text - 0.067)
    if (sisainfus2 . Text == 50.095)
    then
      call Player2 . Start
      call Notifier1 . ShowMessageDialog
      message "Infus 2 Akan Segera habis. segera lakukan pengga..."
      title "Peringatan Infus 2"
      buttonText "Keluar"
  
```

Gambar 8.11 Kode blog sisa cairan infus

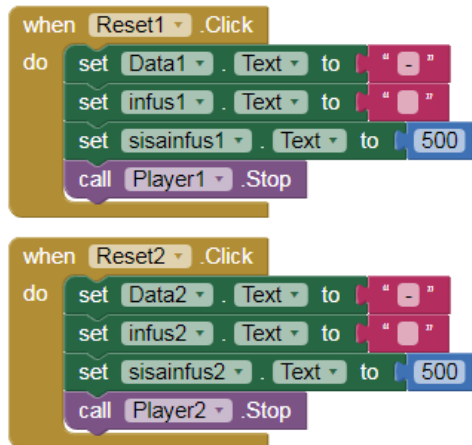
Blok program pada Gambar 8.10 berfungsi untuk mengambil data yang tersimpan pada variabel global data 2 yang diterima dari modul *Bluetooth*. Data yang diterima berupa string dengan format Data=xxx,xxx maka textbox data 1 akan menerima data mulai dari 1 sampai dengan 3, dengan panjang data 3. Sedangkan textbox data 2 akan menerima data mulai dari 5 sampai dengan 7, dengan panjang data 3 dan data tersebut akan dikelompokkan menjadi data menetes dan tidak menetes, dengan perintah jika data yang diterima nilainya kurang dari 500 maka textbox akan menampilkan text tidak menetes dan jika data yang diterima lebih dari 500 maka textbox akan menampilkan text 'menetes'.

Pada Gambar 8.11. kode program pada clock 2 memperlihatkan kejadian yang ditangani oleh clock 2, clock 2 disini berfungsi sebagai clock pada sensor I. Sedangkan, kode program pada clock 3 memperlihatkan kejadian yang ditangani oleh clock 3, clock 3 disini berfungsi sebagai clock pada sensor II. Pada clock 2 dan clock 3 terdapat perintah yang sama yaitu jika textbox menampilkan text menetes maka sisa cairan infus akan berkurang sebesar 0.067 mL. Dan jika sisa cairan infus = 50.095 maka akan muncul peringatan berupa suara yang berbunyi "Infus 1 akan habis" dan akan muncul textbox yang menampilkan text berupa "Infus 1 Akan Segera habis. segera lakukan penggantian pada infus 1"



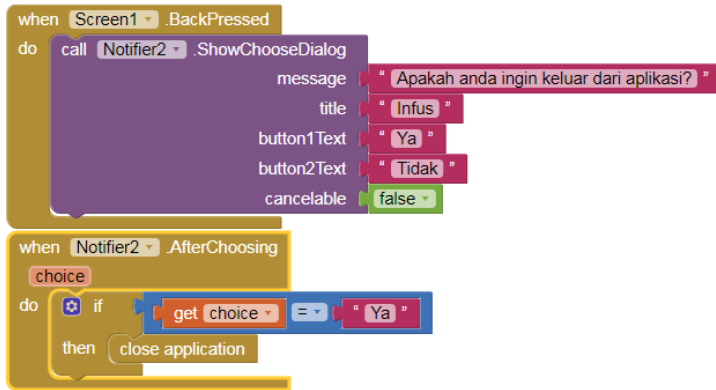
**Gambar 8.12** Kode blok pengaturan kecepatan infus

Pada Gambar 8.12. kode program pada Spinner 1 berfungsi sebagai daftar pilih kecepatan pada sensor I. Sedangkan kode program pada Spinner 2 berfungsi sebagai daftar pilih kecepatan pada sensor II. Pada Spinner 1 dan Spinner 2 terdapat daftar pilih kecepatan tetes yang sama yaitu 5 tetes/menit, 10 tetes/menit, 20 tetes/menit, 50 tetes/menit, dan ketika chamber pada infus dibuka secara total. Jika kecepatan tetesan pada daftar pilih dipilih maka waktu interval pada hasil pembacaan sensor akan berubah sesuai kecepatan tetesan yang dipilih.



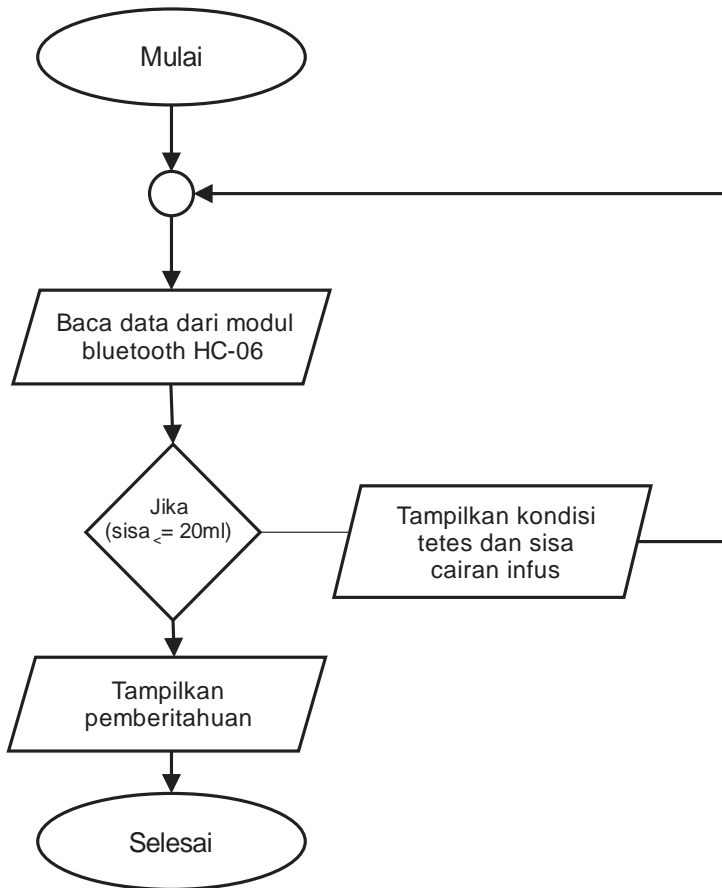
**Gambar 8.13** Kode blok tombol reset

Pada Gambar 8.13 kode program pada Reset1 berfungsi sebagai tombol reset pada pada sensor I. Sedangkan kode program pada Reset2 berfungsi sebagai tombol reset pada pada sensor II. Pada tombol reset 1 dan tombol reset 2 terdapat program yang sama, yaitu apabila tombol reset diklik maka textbox akan kembali kosong dan pada sisa cairan infuse akan kembali menjadi 500mL dan juga berfungsi sebagai tombol stop untuk peringatan suara ketika infus akan habis.



**Gambar 8.14** Kode untuk menutup aplikasi

Pada Gambar 8.14. merupakan kode program sebagai perintah ketika ingin keluar dari aplikasi pemantauan infus. Ketika tombol kembali ditekan maka akan muncul dialog pemberitahuan yang bertuliskan “Apakah ingin keluar dari aplikasi” dan ada pilihan ya atau tidak. Jika iya maka aplikasi akan keluar. Bentuk *flowchart* dapat dilihat pada Gambar 8.15.



**Gambar 8.15** Flow chart program pada MIT App Inventor

Pada flowchart program MIT App Inventor ketika dijalankan maka akan memulai program. Ketika *Bluetooth* disambungkan maka data dari *Bluetooth* HC-06 diterima pada *smartphone* android. Jika sisa cairan infus sama dengan 50,069 mL maka akan ada peringatan



pada *smartphone* android. Jika tidak maka *interface* hanya menampilkan kondisi tetesan dan sisa cairan saat ini.

## **8.4 Pengujian Sistem**

### **8.4.1 Pengujian Perangkat Keras**

Pengujian Perangkat Keras ini dimaksudkan untuk mengetahui hardware yang digunakan bekerja dengan baik atau tidak. Untuk mengetahuinya, maka perlu dilakukan serangkaian pengujian pada masing-masing hardware yang akan digunakan di dalam sistem tersebut.

### **8.4.2 Pengujian perangkat lunak**

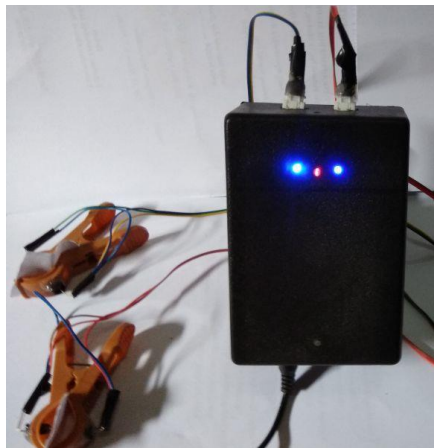
Pengujian ini dimaksudkan untuk meneliti dan menyusun listing pada software agar ketika didownload ke Arduino uno dapat bekerja dengan baik. Dan melakukan perancangan pada aplikasi android

### **8.4.3 Pengujian Keseluruhan alat**

Pengujian alat pemantauan infus berbasis android ini merupakan hasil yang akan diterapkan pada keseluruhan alat sehingga alat mampu bekerja seperti yang disesuaikan secara optimal.

## 8.5 Implementasi Sistem

Sistem pemantauan infus berbasis android dibuat untuk membantu pekerjaan perawat dalam memantau infus pada pasien. Sistem infus pada pasien ini, dirancang menggunakan sensor *photodiode*. Sensor pada sistem ini terletak ditengah – tengah tabung tetes infus yang berfungsi untuk mendeteksi ada tidaknya tetesan. Sistem ini menggunakan Arduino Uno untuk proses pengolahan *input* yang berasal dari *photodiode*. *Prototype* pemantauan infus berbasis android dapat dilihat pada Gambar 8.16



**Gambar 8.16** Perangkat keras alat pemantauan infus

Cara kerja dari sistem ini adalah sensor menerima data dari pancaran sinar inframerah. Data yang diterima berupa perubahan nilai analog cahaya saat ada cahaya dan tidak ada cahaya. Selanjutnya data analog tersebut dikirim melalui Bluetooth HC-06 ke

*handphone* android dan akan diolah dan digolongkan menjadi data ada tetesan atau tidak ada tetesan dan dari data itu kemudian diolah untuk menghitung sisa cairan infus. Setelah data adanya tetesan dan sisa cairan telah didapatkan, selanjutnya data tersebut ditampilkan pada android.

Apabila sisa cairan sama dengan batas yang ditentukan, maka akan ada pemberitahuan pada *smartphone* android perawat. Dan sebaliknya, apabila sisa cairan belum mencapai batas yang ditentukan, maka *smartphone* android akan terus menerima data dan aliran infus tetap mengalir pada selang infus.

Setelah melakukan proses perancangan *software* dan *hardware* selesai, maka tahap selanjutnya adalah melakukan pengujian kinerja pada sistem yang telah dirancang. Pengujian dilakukan agar bisa mendapatkan data dari sistem tersebut, sehingga dengan diperolehnya data dapat diketahui untuk kinerja dari sistem yang telah dirancang tersebut.

### **8.5.1 Pengujian Perangkat Keras**

Pengujian perangkat keras merupakan proses pengekseskuan sistem perangkat keras untuk menentukan apakah sistem tersebut telah bekerja dengan baik. Pengujian dilakukan dengan melakukan percobaan untuk melihat kemungkinan kesalahan yang terjadi dari setiap proses. Pengambilan data pada Alat Pemantauan Infus

Berbasis Android dilakukan dengan pengamatan pada sistem kerja alat dilakukan dengan pengukuran pada komponen-komponen yang digunakan. Dalam tahapan-tahapan melakukan pengujian yang dilakukan pertama kali adalah melakukan pengujian terhadap perangkat-perangkat inputan yaitu pengujian terhadap sensor yang digunakan yaitu sensor photodiode pada sensor 1 dan sensor 2. Kemudian melakukan pengujian secara keseluruhan sistem.



**Gambar 8.17** Pengukuran  $V_{out}$  pada sensor *photodiode*

### 8.5.1.1 Pengujian Sensor Photodiode

Pengujian sensor *photodiode* dilakukan untuk mengetahui respon yang diberikan oleh sensor. Pengujian sensor *photodiode* dilakukan dengan menghubungkan sensor *photodiode* dan led inframerah dengan arduino dengan memberikan tegangan 5 VDC dari modul mikrokontroller Arduino Uno. Saat memancarkan cahaya inframerah pada *photodiode* kemudian *output* tegangan di cek pada pin *Vout* yang dihubungkan dengan *probe* positif dan GND yang dihubungkan dengan *probe* negatif. Maka didapat hasil seperti pada Tabel 8.17

**Tabel 8.1** Keterangan *Vin* dan *Vout* pada sensor *photodiode*

| Sensor    | Tegangan <i>Input</i> | <i>Output Tegangan</i> | Keterangan     |
|-----------|-----------------------|------------------------|----------------|
| Sensor I  | 5 V                   | 1.42 DCV               | Terkena cahaya |
| 4.88 DCV  |                       | Tanpa terkena cahaya   |                |
| Sensor II | 5 V                   | 1.66 DCV               | Terkena cahaya |
| 4.88 DCV  |                       | Tanpa terkena cahaya   |                |

Dari pengujian tersebut dapat diketahui *output* sensor *photodiode* akan bernilai low jika terkena cahaya , sedangkan sensor akan bernilai *high* jika tidak terkena cahaya.

### 8.5.1.2 Pengujian Modul Bluetooth

Pada pengujian koneksi *bluetooth* ke android merupakan komunikasi antara *hardware* dengan *smartphone android* dengan menghubungkan modul *Bluetooth* HC-06 dengan arduino uno dan memberikan terganggan masukan sebesar 5 VDC dengan tujuan untuk mengetahui jarak transmisi antara *bluetooth* yang terdapat pada *smartphone*. Pengujian koneksi *bluetooth* ditunjukkan pada Tabel 8.2

**Tabel 8.2** Pengujian koneksi *bluetooth*

| No | Jarak antara sistem <i>Bluetooth</i> | Keterangan               |
|----|--------------------------------------|--------------------------|
| 1  | 1 Meter                              | Lancar menerima perintah |
| 2  | 5Meter                               | Lancar menerima perintah |
| 3  | 6 Meter                              | Lancar menerima perintah |
| 4  | 7 Meter                              | Lancar menerima perintah |
| 5  | 8 Meter                              | Lancar menerima perintah |
| 6  | 9 Meter                              | Lancar menerima perintah |
| 7  | 10Meter                              | Lancar menerima perintah |
| 8  | 11 Meter                             | Tidak lancar menerima    |
| 9  | 12 Meter                             | Tidak Menerima           |
| 10 | 13 Meter                             | Tidak Menerima           |
| 11 | 14 Meter                             | Tidak Menerima           |
| 12 | 15 Meter                             | Tidak Menerima           |

Pengujian sistem koneksi antara *bluetooth* HC-06 dengan *bluetooth* yang ada di dalam sistem *smartphone* android merupakan komunikasi *nirkabel* (tanpa kabel) yang sangat mudah digunakan pada aplikasi ini. Jarak yang sangat efektif adalah 1 meter sampai 10 meter untuk menstabilkan hasil ukur pada sensor yang dikoneksikan pada android.

## 8.5.2 Pengujian *Software*

### 8.5.2.1 Pengujian Aplikasi pada *Smartphone Android*

Pengujian sistem ini dilakukan dengan cara meng instalasi aplikasi yang telah dibuat dengan menggunakan MIT App Inventor. Aplikasi diberi nama “Infus”. Tampilan untuk membuka aplikasi di tunjukan pada Gambar 8.18



**Gambar 8.18** Tampilan aplikasi pada *smartphone android*

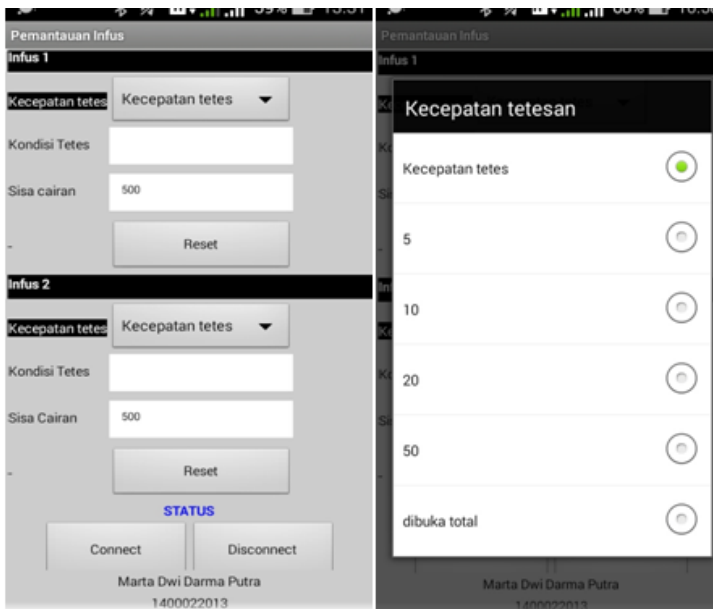
Pada Gambar 8.18 Tampilan aplikasi *smartphone* android pada umumnya sama dengan aplikasi yang lainnya. Pengujian aplikasi ini dilakukan untuk mengetahui respon yang diberikan oleh aplikasi pemantauan infus berbasis android terhadap hardware yang terdiri dari arduino uno dan modul *bluetooth* hc-06. Setelah aplikasi sudah terinstal pada android maka langkah selanjutnya adalah mengaktifkan *bluetooth* yang ada pada *smartphone* android sebagai *wireless* pada *smartphone* seperti pada Gambar 8.19



**Gambar 8.19** Mengaktifkan *bluetooth* pada *smartphone*

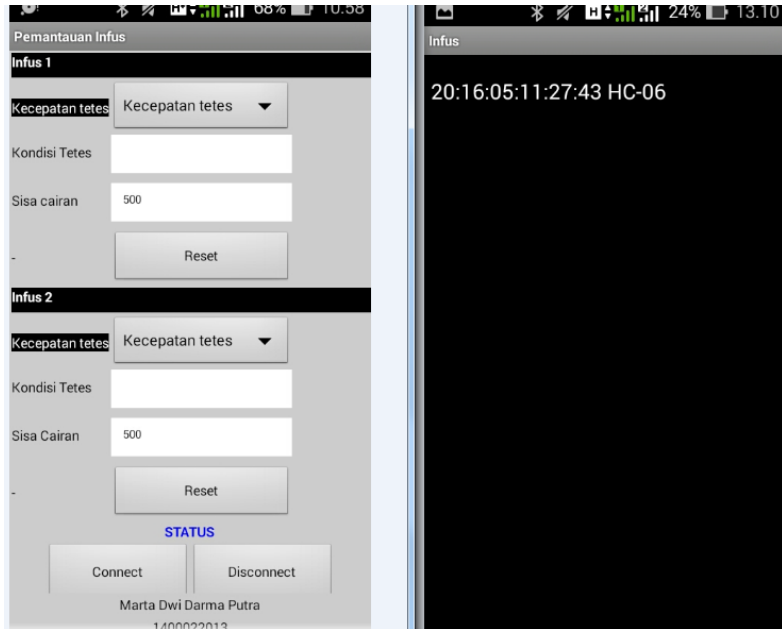


Setelah dapat dipastikan bahwa *bluetooth* pada android sudah pada mode aktif maka langkah selanjutnya adalah membuka aplikasi pemantauan infus yang sudah terinstal pada android. Setelah aplikasi pemantauan infus pada android terbuka maka selanjutnya pilih kecepatan tetesan yang diinginkan sesuai dengan pengaturan kecepatan pada chamber infus. Seperti pada Gambar 8.20



**Gambar 8.20** Memilih kecepatan tetesan

Selanjutnya langkah untuk menghubungkan ke *hardware* adalah dengan meng-klik tombol *bluetooth* yang ada pada aplikasi, selanjutnya menghubungkan dengan *bluetooth* HC-06 seperti Gambar 8.21



**Gambar 8.21** Aplikasi pemantauan pada android

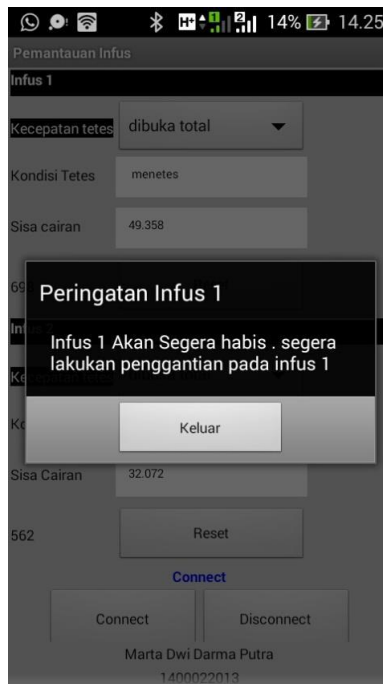
Pada saat menghubungkan perangkat *bluetooth* antara android dengan *hardware* posisi keduanya harus tidak terlalu jauh saat menghubungkan. Karena dapat mengganggu respon yang diberikan pada aplikasi dan menyebabkan *error*. Setelah dapat terkoneksi dengan *bluetooth* maka hasil pemantauan akan tertampil nilai ukur pada android seperti pada Gambar 8.22



**Gambar 8.22** Tampilan data pada aplikasi pemantauan infus

Selanjutnya pada hasil penelitian dan pengujian diperoleh bukti saat alat mulai dihidupkan (disambungkan ke catu daya) dan kedua sensor di pasang diinfus maka akan menampilkan hasil pemantauan kondisi tetes pada infus I dan infus II. Kemudian *bluetooth* akan mengirimkan data digital dari *hardware* ke android berupa hasil pembacaan sensor photodiode dan kemudian data tersebut akan dikonversi menjadi kondisi tetesan infus. Apabila, nilai pembacaan sensor =  $<500$  maka kondisi tetesan pada infus akan menjadi “tidak

menetes”. Dan apabila nilai pembacaan sensor = >500 maka kondisi tetes akan menjadi “menetes” dan pada sisa cairan infus akan berkurang sebesar 0,067 yang artinya alat pemantauan sudah dapat berjalan sesuai dengan yang diharapkan yaitu menampilkan kondisi tetes dan sisa cairan infus. Selanjutnya adalah apabila sisa cairan infus = 50 mL maka akan muncul notifikasi seperti pada Gambar 8.23



**Gambar 8.23** Peringatan ketika infus akan habis

Pada gambar 8.23. pemberitahuan peringatan ketika infus akan habis berbentuk dialogbox yang berisi text “Infus 1 akan segera

habis. Segera lakukan penggantian pada infus 1”. Disertai juga dengan notifikasi berupa nada suara yang berbunyi”Infus 1 akan habis”. Untuk menghentikan nada suara maka klik tombol reset baik itu pada infus I maupun pada infus II.

Selanjut nya ketika sudah selesai menggunakan aplikasi pada android maka hal berikutnya adalah *logout* pada aplikasi dengan menekan tombol kembali pada android kemudian tekan “Ya”. Tampilan ketika *logout* dapat dilihat pada Gambar 8.24



**Gambar 8.24** Tampilan untuk melakukan *logout* aplikasi

## 8.6 Hasil dan Pengujian Data

Pada pengujian data pada alat pemantauan infus berbasis android dilakukan dengan cara menguji perangkat keras (*hardware*) dan perangkat lunak (*software*). Untuk pengujian alat diaplikasikan secara langsung menggunakan infus set. Pada pengujian ini dilakukan oleh penulis dengan bantuan ibu Pelita Al Islamiah, SKM.S.ST yang memang mehamai tentang infus. Pengujian pada infus dapat dilihat pada Gambar 8.25



**Gambar 8.25** Gambaran secara keseluruhan

### 8.6.1 Pengujian sensor pendeteksi tetesan

Pada alat pemantauan infus berbasis android untuk mendeteksi adanya tetesan maka pasang sensor pada chamber infus. Kemudian buka perlahan regulator pengatur tetesan pada infus. Sehingga terlihat ada cairan infus yang menetes. Setiap ada cairan yang menetes maka LED indikator pada *hardware* akan menyala dan pada *smartphone* android kondisi tetesan akan menjadi menetes dan ketika tidak ada tetesan maka kondisi tetesan akan menjadi tidak menetes.

Pada alat ini terdapat 2 (dua) sensor pendeteksi tetesan yaitu pendeteksi tetesan pada infus I dan pendeteksi tetesan pada infus II. Maka dari itu dilakukan pengujian pada infus I dan infus II.

#### 1. Pengujian sensor pendeteksi pada infus I

Dalam pengujian sensor pendeteksi tetesan pada infus 1. Hasil pengujian dapat dilihat dilihat pada Tabel 8.3

**Tabel 8.3** Hasil Pengujian Sensor Pendeteksi Tetesan Infus I

| No | Kondisi Tetesan | Kondisi LED Indikator | Vout sensor |
|----|-----------------|-----------------------|-------------|
| 1  | Menetes         | Kedip                 | 1.42 DCV    |
| 2  | Tidak Menetes   | Padam                 | 4.88 DCV    |

## 2. Pendeteksi tetesan pada infus II

Dari pengujian sensor pendeteksi tetesan pada infus II. Hasil pengujian dapat dilihat dilihat pada Tabel 8.4

**Tabel 8.4** Hasil Pengujian Sensor Pendeteksi Tetesan Infus II

| No | Kondisi Tetesan | Kondisi LED Indikator | Vout sensor |
|----|-----------------|-----------------------|-------------|
| 1  | Menetes         | Kedip                 | 1.66 DCV    |
| 2  | Tidak Menetes   | Padam                 | 4.88 DCV    |

### 8.6.2 Pengujian Pengamatan Sisa Cairan Infus

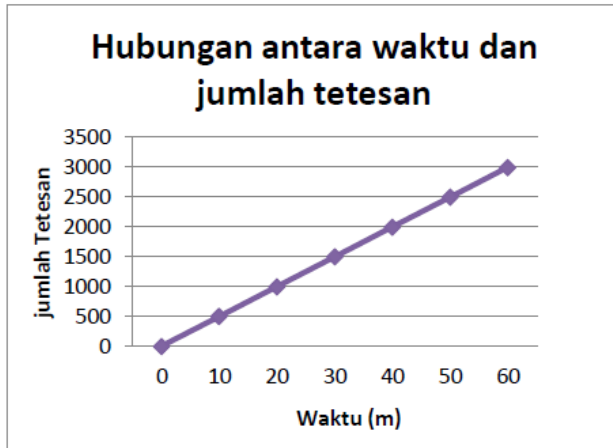
Pemantauan infus berbasis android menampilkan sisa cairan infus. Tampilan sisa cairan infus akan berkurang setiap tetesnya. Jika 1mL = 15 tetes makro maka 1 tetes = 0.067 mL, jadi setiap tetesnya akan berkurang 0.067 mL. Sebelum dilakukan pengujian pengamatan sisa cairan infus, dilakukan pengujian hubungan jumlah tetesan terhadap waktu. Berikut ini adalah data yang menunjukkan hubungan jumlah tetesan terhadap waktu. Hasil yang didapat dapat dilihat pada tabel 8.5.



**Tabel 8.5** Hubungan jumlah tetesan terhadap waktu

| <b>No</b> | <b>Waktu (m)</b> | <b>Jumlah Tetesan<br/>pada infus</b> |
|-----------|------------------|--------------------------------------|
| 1         | 0                | 0                                    |
| 2         | 10               | 500                                  |
| 3         | 20               | 1000                                 |
| 4         | 30               | 1500                                 |
| 5         | 40               | 2000                                 |
| 6         | 50               | 2500                                 |
| 7         | 60               | 3000                                 |

Pengambilan data dilakukan pada waktu pengujian hubungan jumlah tetesan terhadap waktu dilakukan dengan mengatur kecepatan pada infus sebesar 50 tetes/menit dan dilakukan selama 60 menit. Dari data yang telah diperoleh menunjukkan bahwa hubungan jumlah tetesan terhadap waktu adalah bersifat linear. Linieritas data yang telah diambil ditunjukkan pada Gambar 8.26 dibawah ini.



**Gambar 8.26** Grafik hubungan antara waktu dan jumlah tetesan

# BAB 9: Aplikasi Pemantau Terapi Jari Pasca Stroke Menggunakan Sarung Tangan Pintar Secara Real Time

## **BAB 9: Aplikasi Pemantau Terapi Jari Pasca Stroke Menggunakan Sarung Tangan Pintar Secara Real Time**

### **9.1. Subyek Penelitian**

Perancangan aplikasi pemantau terapi jari pasca *stroke* menggunakan sarung tangan pintar secara *real time* merupakan subyek dalam proyek ini. Informasi diperoleh dari berbagai sumber diantaranya seorang fisioterapis, buku asuhan keperawatan, dan internet. Beberapa sensor *flex* yang digunakan untuk mendeteksi gerakan-gerakan dari setiap jari. Semua sensor melakukan pembacaan diolah mikrokontroler untuk dimasukkan datanya ke *microsoft visual basic* yang kemudian diproses menjadi sebuah grafik *real time*. Hasil dari pengolahan data kemudian dikeluarkan ke tampilan pada layar laptop/PC sesuai dengan gerakan yang telah ditentukan.

### **9.2. Bahan Penelitian**

Bahan-bahan penelitian yang digunakan dalam perancangan aplikasi dan sistem ini adalah:

1. **Papan mikrokontroler *arduiono nano*** digunakan sebagai prosesor untuk membaxa setiap perubahan data dari sensor.
2. **Sarung tangan pintar** yang dilengkapi sensor *flex* sebagai pendeteksi gerakan dan tekukan jari-jari tangan.

3. **Modul *Bluetooth HC-05*** digunakan sebagai konektor antar perangkat keras dan perangkat lunak.
4. ***Microsoft Visual Basic 2010*** sebagai perangkat lunak yang digunakan untuk menampilkan garfik secara *real time*.

### 9.3. Alat Penelitian

Proyek ini membutuhkan beberapa peralatan untuk membuat rancangan aplikasi pemantau terapi pasca *stroke* menggunakan sarung tangan pintar secara *real time*, peralatan yang dibutuhkan antara lain:

1. Perangkat keras mikrokontroler *arduino nano*: digunakan sebagai prosesor pada sistem utama.
2. Perangkat lunak *Arduino IDE*: digunakan untuk menulis program ke dalam papan mikrokontroler *arduino nano*.
3. Perangkat lunak *Visual Basic 2010*: digunakan untuk membangun aplikasi yang digunakan untuk menampilkan grafik secara *real time*.
4. Laptop atau PC: digunakan sebagai media untuk menjalankan aplikasi yang telah dibangun pada proyek ini.
5. *Multimeter digital*: digunakan untuk mengukur tegangan dan arus pada rangkaian, menguji jalur-jalur rangkaian sebelum dinyalakan menggunakan catu daya baterai.

6. Tang potong: digunakan untuk memotong kabel dan kaki-kaki komponen saat perakitan sistem pada perangkat keras.
7. Solder dan timah: digunakan untuk pemasangan komponen-komponen saat dirakit.

#### 9.4. Perancangan Sistem

Perancangan sistem dilakukan dalam dua tahap perancangan yaitu perancangan perangkat keras dan perancangan perangkat lunak. Untuk menghasilkan sistem yang ideal, maka perancangan sistem dilakukan dengan mengacu pada teori-teori yang telah ada dan pemanfaatan *datasheet* sensor dan mikrokontroler yang digunakan.

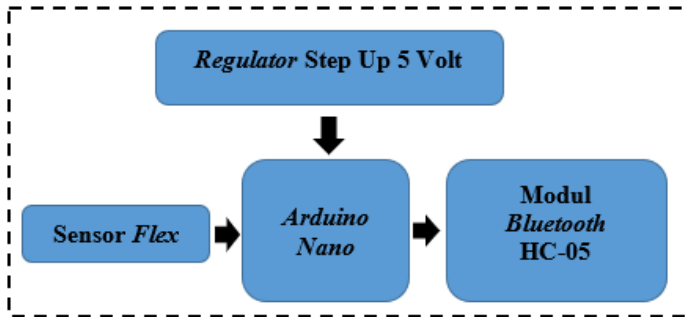
Tahap pertama dalam perancangan sistem ini yaitu membuat blok diagram perangkat keras, kemudian dilanjutkan dengan membangun sebuah sarung tangan perangkat pintar (*smart glove*). Tahap kedua yaitu merancang perangkat lunak yang digunakan untuk mengoperasikan sarung tangan pintar ini dan sekaligus sebagai penampil grafik secara *real time*.

##### 9.4.1 Perancangan Perangkat Keras

Perancangan perangkat keras terdiri dari sistem kontrol yaitu *arduino nano* yang bertugas sebagai pengendali dan pembaca data ADC yang dikeluarkan oleh sensor *flex* serta melakukan pengolahan

data. Perancangan rangkaian-rangkaian sensor serta baterai dan *converter DC step-up* untuk mencatu tegangan pada semua perangkat keras yang digunakan dalam proyek ini.

Diagram blok sarung tangan pintar yang dirancang dalam penelitian ini dapat dilihat pada Gambar 9.1.



**Gambar 9.1** Diagram Blok Sarung Tangan Pintar

Dari diagram blok di atas dapat diketahui bahwa pada bagian catu daya tegangan menggunakan regulator 5 volt DC yang difungsikan sebagai penaik tegangan dari sebuah baterai yang memiliki tegangan 3,7 volt. Kemudian pada bagian *input* atau masukan menggunakan sensor *flex* yang berjumlah 5 buah yang dipasang pada tiap-tiap jari tangan. Pada bagian prosesor digunakan sebuah mikrokontroler *arduino nano* yang bertugas untuk melakukan berbagai macam tugas baik pembacaan data dari sensor serta mentransfer data ke perangkat lunak menggunakan bantuan modul

*bluetooth* HC-05. Dengan bantuan modul *bluetooth* HC-05 maka diharapkan sistem akan menjadi lebih efisien dan ringkas.

#### 9.4.2 Perancangan Perangkat Lunak

Pada bagian sebelumnya telah diuraikan tentang perancangan perangkat keras, untuk itu agar perangkat keras tersebut dapat bekerja maka dibutuhkan juga perancangan perangkat lunak. Perangkat lunak untuk program mikrokontroler *arduino uno* menggunakan *software arduino* IDE dan *software visual basic* 2010.

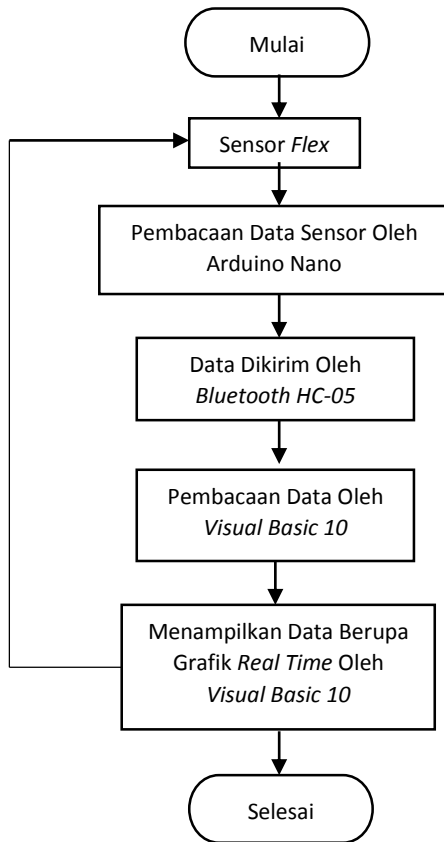
##### - Pemrograman Arduino

Pemrograman mikrokontroler *arduino nano* dilakukan agar dapat melakukan suatu proses pada rangkaian tersebut. Dalam penelitian ini, untuk memprogram mikrokontroler digunakan *software Arduino* yang menggunakan bahasa C. sebelum merancang sebuah program, terlebih dahulu dibuat diagram alir (*flowchart*).

Diagram alir memiliki peranan penting dalam pembuatan sebuah program agar program tersebut bekerja sesuai dengan yang diharapkan. Saat perangkat keras (sarung tangan pintar) diberi catu daya, mikrokontroler *arduino nano* memulai proses inisialisasi jalur masuk, jalur keluar, dan beberapa variabel yang dibutuhkan. Selanjutnya data yang masuk akan dikonversikan dan diolah oleh



mikrokontroler. Hasil pengolahan tersebut menjadi acuan untuk grafik *real time* yang akan dikeluarkan. Diagram alir sarung tangan pintar dapat dilihat pada Gambar 9.2.



**Gambar 9.2** Diagram Alir Aplikasi dan Sarung Tangan Pintar

Dari diagram alir di atas dapat disimpulkan bahwa pada bagian masukan (*input*) terdapat sensor *flex*. Dimana saat sistem dimulai maka sistem akan melakukan pembacaan data dari sensor *flex*. Kemudian setelah data dibaca dan diproses oleh *arduino nano*, selanjutnya data akan dikirim melalui koneksi nirkabel yaitu menggunakan modul *bluetooth HC-05*. Setelah data dikirim, maka dilakukan proses pembacaan data oleh aplikasi yang telah dibangun menggunakan *microsoft visual basic 2010*.

Setelah *visual basic* membaca dan memproses data masukan dari sensor *flex* yang telah diproses oleh *arduino nano*, kemudian data tersebut akan ditampilkan menjadi sebuah grafik *real time* yang akan berubah-ubah sesuai gerakan jari-jari tangan. Pada proses pembacaan data dan penampilan data menjadi grafik tersebut, maka proses akan berulang-ulang secara terus menerus sampai sistem dimatikan.

#### - Pemrograman Visual Basic 2010

Pemrograman *visual basic* dilakukan agar dapat melakukan suatu proses pada aplikasi yang akan dibangun. Dalam penelitian ini, untuk memprogram *visual basic* digunakan *software microsoft visual basic 2010* yang menggunakan bahasa C. Sebelum merancang sebuah aplikasi, terlebih dahulu disusun komponen apa saja yang dibutuhkan

untuk membuat tampilan GUI (*Graphic User Interface*). GUI memiliki peranan penting dalam pembuatan sebuah aplikasi agar aplikasi tersebut mudah untuk dipahami serta mudah untuk digunakan dan dapat bekerja sesuai dengan yang diharapkan.

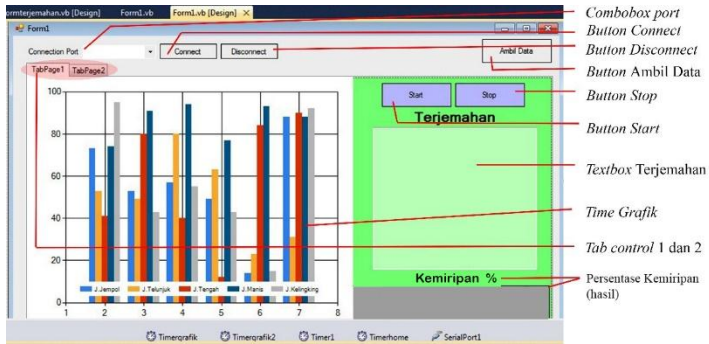
Untuk membangun aplikasi ini dibutuhkan komponen-komponen penyusun GUI serta fungsinya masing-masing adalah sebagai berikut:

1. **From1:** sebagai wadah untuk menempatkan berbagai komponen yang akan dibutuhkan untuk aplikasi ini.
2. **Label:** sebagai komponen penyusun yang digunakan untuk memberikan inisial nama pada tiap-tiap fungsi.
3. **Combobox port:** sebagai komponen untuk memilih *port* masukan dari sarung tangan.
4. **Buttonconnect:** sebagai tombol untuk melakukan eksekusi penghubung antar sarung tangan dengan aplikasi.
5. **Buttondisconnect:** sebagai pemutus hubungan setelah tersambung dengan perangkat sarung tangan.
6. **Button ambil data:** digunakan untuk eksekusi pengambilan data awal gerakan jari.
7. **Tabcontrol 1:** digunakan sebagai wadah untuk menempatkan grafik atau *chart* yang digunakan.

8. **TabControl 2:** digunakan sebagai wadah untuk menempatkan grafik atau *chart* saat pengambilan data awal.
9. **GroupBox:** digunakan sebagai wadah untuk menempatkan inisialisasi setiap jari. *GroupBox* terdiri dari beberapa label penamaan masing-masing jari.
10. **Button:** beberapa *button* pilihan seperti terjemahan baru, ubah terjemahan dan hapus terjemahan tersedia di halaman pengambilan data.
11. **DataGridView:** digunakan untuk penempatan data setelah pengambilan data dilakukan.
12. **Panel:** digunakan sebagai wadah untuk menempatkan button terjemahan baru, ubah terjemahan dan hapus terjemahan.
13. **Textbox:** digunakan untuk menampilkan terjemahan hasil dari gerakan jari-jari tangan.
14. **Trackbar:** digunakan untuk memilih tingkat toleransi data yang akan dibaca. Mulai dari 5% hingga 25%.
15. **Timegrafik:** digunakan untuk menampilkan data grafik secara terus menerus sehingga dapat menampilkan hasil data secara *real time*.
16. **Timer:** digunakan untuk pendukung *time* grafik.
17. **Serialport:** digunakan untuk menginisialisai jalur masuk dari data serial *port* yang digunakan.

18. **Buttonstart dan stop:** digunakan untuk eksekusi saat akan menampilkan grafik gerakan tangan secara *real time*.
19. **Form terjemahan:** digunakan sebagai menu *popup* saat melakukan eksekusi penambahan terjemahan yang mana didalamnya berisi *numericupdown* sebagai pemilih alamat yang akan disimpan, serta *button* simpan dan *button cancel* yang mana berfungsi untuk menyimpan data dan membatalkan saat pengambilan data dilakukan. Serta dilengkapi dengan *textbox* untuk mengisi *form* terjemahan data.

Saat perangkat keras (sarung tangan pintar) disambungkan dengan aplikasi yang telah dibangun, *microsoft visual basic* memulai proses inisialiasai perangkat, kemudian setelah koneksi terhubung maka tahap selanjutnya adalah proses inisialiasai perubahan nilai pada masing-masing sensor *flex* pada tiap-tiap jari. Selanjutnya data yang masuk akan dikonversikan dan diolah oleh *microsoft visual basic* sehingga dapat menampilkan data keluaran berupa grafik *real time*. Hasil pengolahan tersebut menjadi acuan untuk pantauan akan keadaan pasien yang diteliti. Aplikasi yang dibangun menggunakan *software microsoft visual basic 2010* dapat dilihat pada Gambar 9.3.



**Gambar 9.3** Hasil Rancangan Aplikasi yang dibangun

Pada perancangan grafik *real time* seperti yang ditunjukkan pada Gambar 9.3, peneliti menggunakan angka 0 sampai dengan 100 sebagai instrumen ukur untuk menunjukkan besarnya perubahan nilai data adc dari sensor *flex*. Pada dasarnya perubahan nilai data adc dengan menggunakan mikrokontroler *arduino nano 10 bit* adalah 0 sampai 1024. Namun nilai tersebut dianggap terlalu besar untuk perubahan nilai dari sensor *flex* itu sendiri. Sehingga peneliti menetapkan angka 0 sampai 1024 tersebut menjadi nilai minimal adalah 0 dan untuk nilai maksimalnya adalah 100.

Penjelasan program untuk merubah angka 0 sampai 1024 pada tampilan aplikasi yang peneliti bangun akan dijelaskan lebih lanjut dan detail pada bagian lampiran 2, yaitu lampiran penjelasan pemrograman *microsoft visual basic*. Adapun untuk listing program pada keseluruhan sistem dan aplikasi juga dapat dilihat pada

lampiran 1 dan lampiran 2 yang disajikan pada halaman akhir dari penelitian ini.

## 9.5. Pengujian Sistem

Pengujian sistem dapat dilakukan setelah semua proses perancangan selesai. Proses pengujian ini untuk melihat apakah sistemnya dapat bekerja sesuai dengan fungsi dan tugasnya masing-masing. Sistem yang pertama diuji adalah pada bagian perangkat keras, yaitu sensor *flex* dan modul *bluetooth*. Kemudian sistem yang diuji pada perangkat lunak adalah aplikasi yang peneliti bangun. Berikut dibawah ini adalah beberapa hasil pengujian sistem dari masing-masing bagian yang diuji serta pengujian sistem secara keseluruhan.

Ada beberapa pengujian yang dilakukan, antara lain pengujian sensor *flex*, pengujian koneksi modul *bluetooth*, pengujian aplikasi yang dibangun serta pengujian sistem secara keseluruhan. Pada pengujian sistem secara keseluruhan dilakukan pengambilan data hasil gerakan jari-jari tangan orang normal, untuk mendapatkan data acuan saat pengujian terhadap penderita *stroke*.

### 9.5.1 Pengujian Sensor Flex

Sensor *flex* adalah sensor yang memiliki output berupa resistansi. Prinsip kerjanya sama dengan potensiometer yang

berubah resistansinya ketika terkena lekukan. Keluaran sensor berupa tegangan analog yang dapat diukur menggunakan voltmeter. Untuk mendapatkan nilai ADC 10 bit digunakan rumus:

$$\text{Data ADC} = \frac{V_{in} \times 1024}{V_{ref}} \quad (1)$$

$V_{ref}$  (tegangan referensi) yang digunakan adalah 5 volt. Ada 3 posisi yang diuji untuk sensor *flex*, yaitu posisi jari lurus semua, posisi jari menekuk 90 derajat, dan posisi jari menggenggam penuh.

- Posisi Jari Lurus Semua

$V_{out}$  yang terukur = 3,40 volt

$$\text{Data ADC} = \frac{V_{in} \times 1024}{V_{ref}} = \frac{3,40 \times 1024}{5} \quad (2)$$

Data ADC = 696,32

- Posisi Jari Menekuk 90°

$V_{out}$  yang terukur = 3 volt

$$\text{Data ADC} = \frac{V_{in} \times 1024}{V_{ref}} = \frac{3 \times 1024}{5} \quad (3)$$

Data ADC = 614,4



- Posisi Menggenggam Penuh

Vout yang terukur = 2,7 volt

$$\text{Data ADC} = \frac{V_{in} \times 1024}{V_{ref}} = \frac{2,7 \times 1024}{5} \quad (4)$$

Data ADC = 552,96

Hasil dari pengujian dan persamaan di atas menunjukkan bahwa terdapat perbedaan nilai antara posisi jari lurus semua, menekuk 90° dan menggenggam penuh. Hal ini menunjukkan bahwa sensor *flex* sudah berfungsi dengan sebagaimana mestinya dan layak untuk digunakan pada penelitian ini.

### 9.5.2 Pengujian Sistem Secara Keseluruhan

Pada pengujian sistem secara keseluruhan, dibuatlah data acuan dari setiap gerakan jari-jari tangan yang berbeda satu sama lainnya. Hal ini bertujuan agar dapat membedakan grafik hasil dari setiap gerakan yang diberikan sarung tangan pintar pada penelitian ini. Data-data grafik yang telah disimpan inilah yang dijadikan acuan untuk masing-masing gerakan. Berikut di bawah ini merupakan beberapa contoh data acuan dari tiap-tiap gerakan terapi jari yang sudah disimpan.

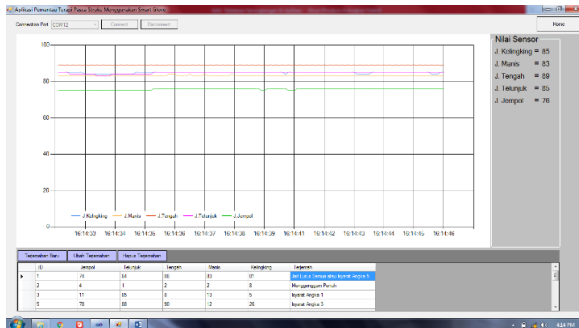
- Data Grafik Posisi Jari Lurus Semua atau Isyarat Angka 5

Data dari grafik pada posisi jari diluruskan semua atau sama dengan isyarat angka 5 adalah seperti ditunjukkan pada Gambar 9.4



**Gambar 9.4** Grafik Semua Jari Lurus atau Isyarat Angka 5

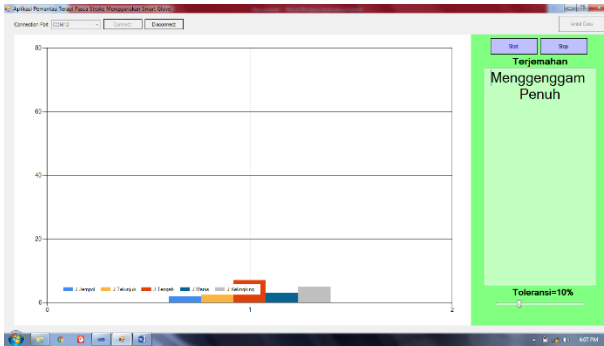
Adapun proses saat pengambilan data dapat dilihat pada Gambar 9.5.



**Gambar 9.5** Proses Pengambilan Data Grafik Semua Jari Lurus atau Isyarat Angka 5

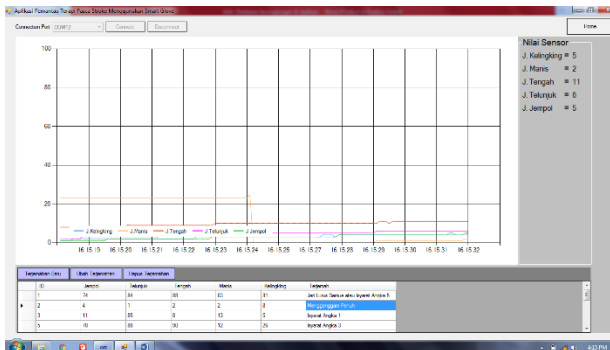
- Data Grafik Posisi Menggenggam Penuh

Data dari grafik pada posisi jari menggenggam penuh adalah seperti ditunjukkan pada 9.6.



**Gambar 9.6** Grafik Posisi Menggenggam Penuh

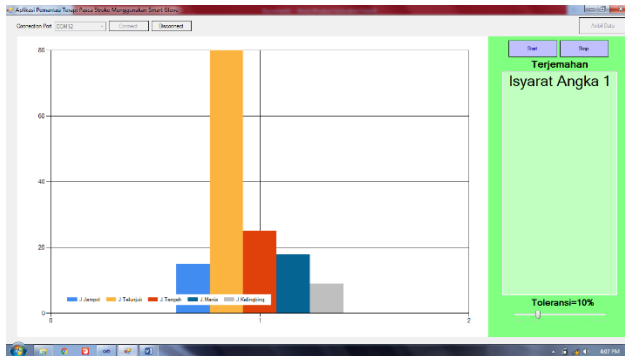
Adapun proses saat pengambilan data dapat dilihat pada Gambar 9.7



**Gambar 9.7** Proses Pengambilan Data Grafik Posisi Menggenggam Penuh

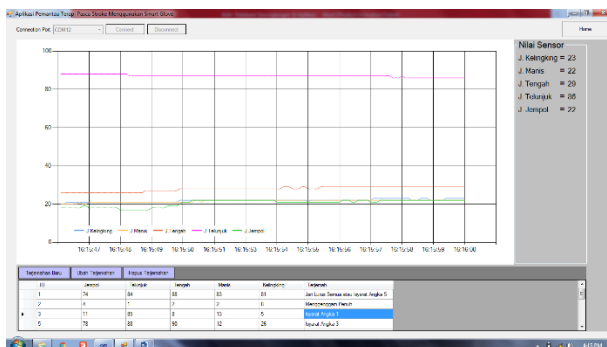
- Data Grafik Posisi Jari Menunjukkan Isyarat Angka 1

Data dari grafik pada posisi jari menunjukkan isyarat angka 1 adalah seperti ditunjukkan pada Gambar 9.8.



Gambar 9.8 Grafik Posisi Jari Menunjukkan Isyarat Angka 1

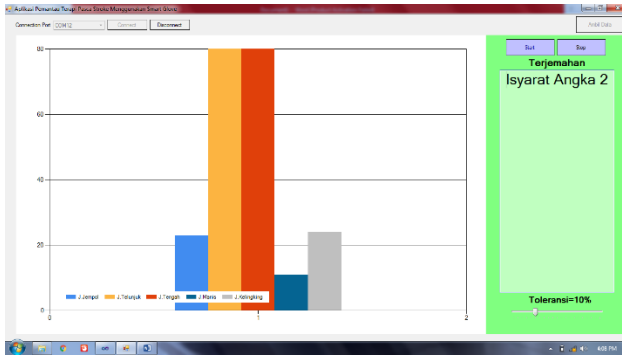
Adapun proses saat pengambilan data dapat dilihat pada Gambar 9.9.



Gambar 9.9 Proses Pengambilan Data Grafik Posisi Jari Menunjukkan Isyarat Angka 1

- Data Grafik Posisi Jari Menunjukkan Isyarat Angka 2

Data dari grafik pada posisi jari menunjukkan isyarat angka 2 adalah seperti ditunjukkan pada Gambar 9.10.



**Gambar 9.10** Grafik Posisi Jari Menunjukkan Isyarat Angka 2

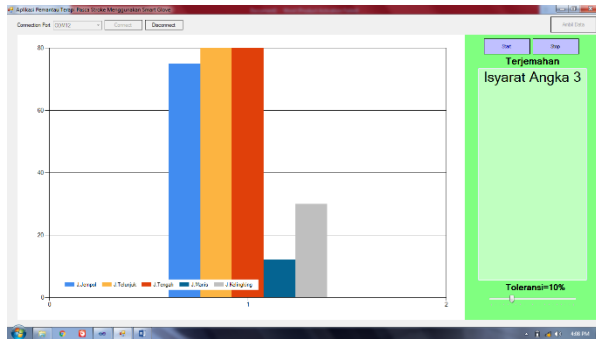
Adapun proses saat pengambilan data dapat dilihat pada Gambar 9.11



**Gambar 9.11** Proses Pengambilan Data Grafik Posisi Jari Menunjukkan Isyarat Angka 2

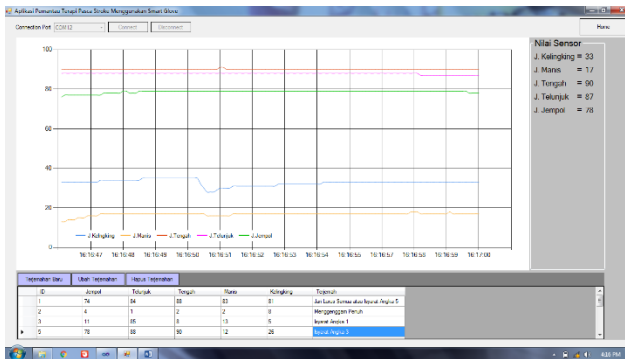
- Data Grafik Posisi Jari Menunjukkan Isyarat Angka 3

Data dari grafik pada posisi jari menunjukkan isyarat angka 3 adalah seperti ditunjukkan pada Gambar 9.12



**Gambar 9.12** Grafik Posisi Jari Menunjukkan Isyarat Angka 3

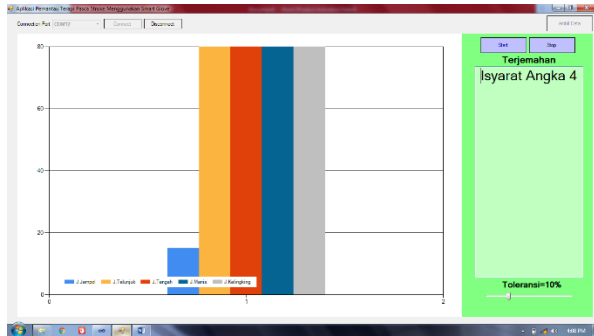
Adapun proses saat pengambilan data dapat dilihat pada Gambar 9.13



**Gambar 9.13** Proses Pengambilan Data Grafik Posisi Jari Menunjukkan Isyarat Angka 3

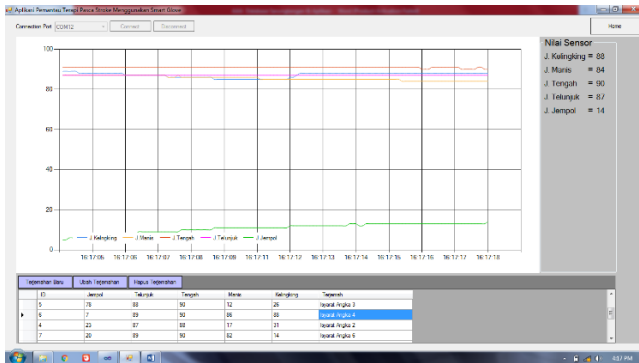
- Data Grafik Posisi Jari Menunjukkan Isyarat Angka 4

Data dari grafik pada posisi jari menunjukkan isyarat angka 4 adalah seperti ditunjukkan pada Gambar 9.14



**Gambar 9.14** Grafik Posisi Jari Menunjukkan Isyarat Angka 4

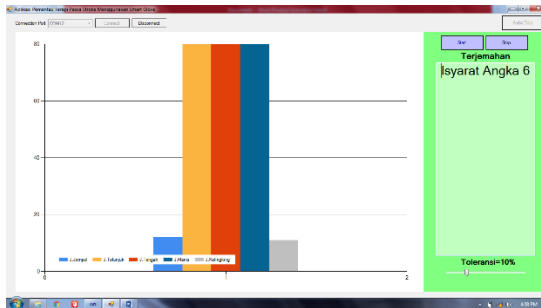
Adapun proses saat pengambilan data dapat dilihat pada Gambar 9.15



**Gambar 9.15** Proses Pengambilan Data Grafik Posisi Jari Menunjukkan Isyarat Angka 4

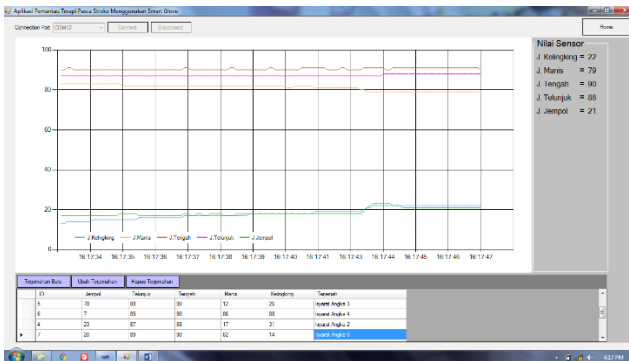
- Data Grafik Posisi Jari Menunjukkan Isyarat Angka 6

Data dari grafik pada posisi jari menunjukkan isyarat angka 6 adalah seperti ditunjukkan pada Gambar 9.16



**Gambar 9.16** Grafik Posisi Jari Menunjukkan Isyarat Angka 6

Adapun proses saat pengambilan data dapat dilihat pada Gambar 9.17

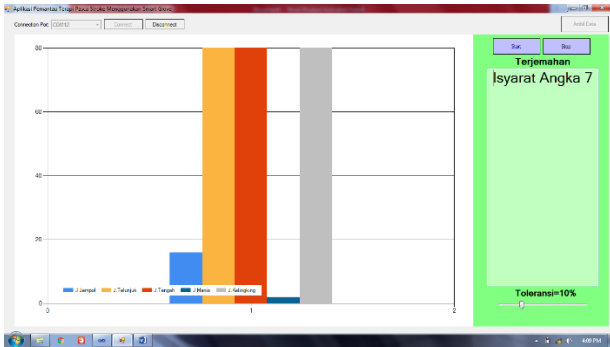


**Gambar 9.17** Proses Pengambilan Data Grafik Posisi Jari Menunjukkan Isyarat Angka 6



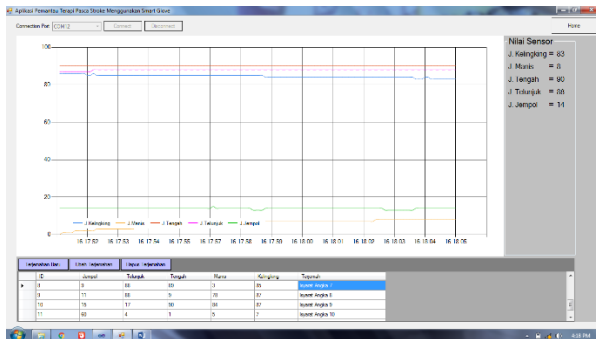
- Data Grafik Posisi Jari Menunjukkan Isyarat Angka 7

Data dari grafik pada posisi jari menunjukkan isyarat angka 7 adalah seperti ditunjukkan pada Gambar 9.18



**Gambar 9.18** Grafik Posisi Jari Menunjukkan Isyarat Angka 7

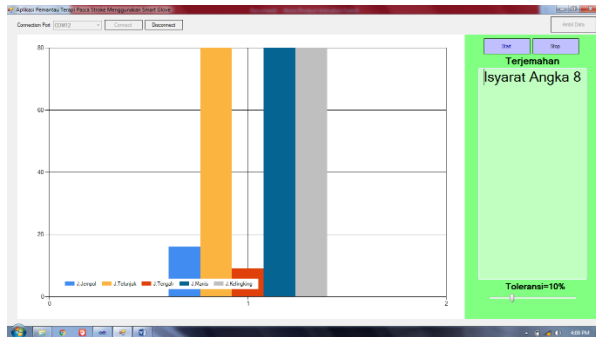
Adapun proses saat pengambilan data dapat dilihat pada Gambar 9.19



**Gambar 9.19** Proses Pengambilan Data Grafik Posisi Jari Menunjukkan Isyarat Angka 7

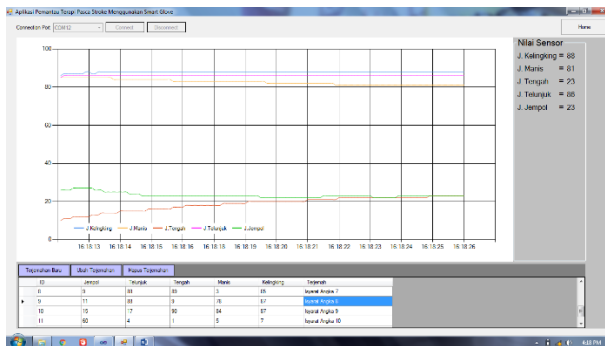
- Data Grafik Posisi Jari Menunjukkan Isyarat Angka 8

Data dari grafik pada posisi jari menunjukkan isyarat angka 8 adalah seperti ditunjukkan pada Gambar 9.20



**Gambar 9.20** Grafik Posisi Jari Menunjukkan Isyarat Angka 8

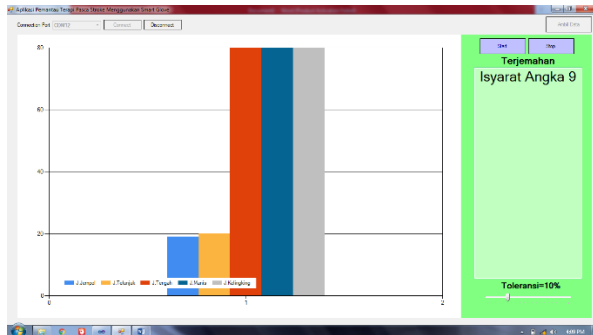
Adapun proses saat pengambilan data dapat dilihat pada Gambar Gambar 9.21



**Gambar 9.21** Proses Pengambilan Data Grafik Posisi Jari Menunjukkan Isyarat Angka 8

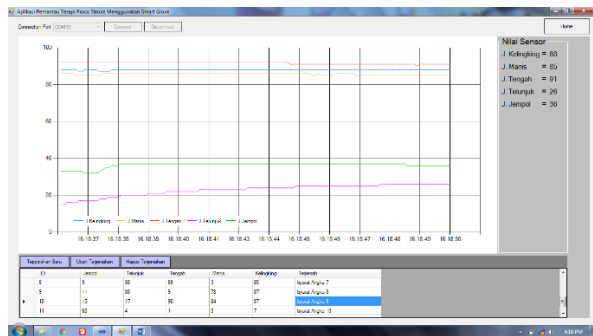
- Data Grafik Posisi Jari Menunjukkan Isyarat Angka 9

Data dari grafik pada posisi jari menunjukkan isyarat angka 9 adalah seperti ditunjukkan pada Gambar 9.22



**Gambar 9.22** Grafik Posisi Jari Menunjukkan Isyarat Angka 9

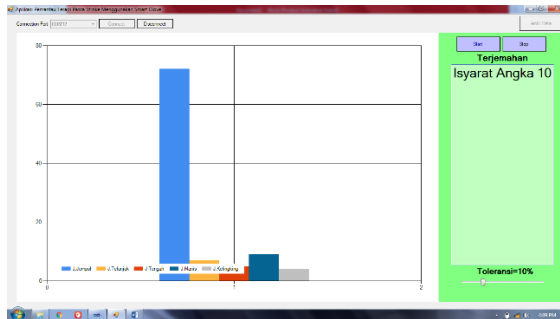
Adapun proses saat pengambilan data dapat dilihat pada Gambar 9.23



**Gambar 9.23** Proses Pengambilan Data Grafik Posisi Jari Menunjukkan Isyarat Angka 9

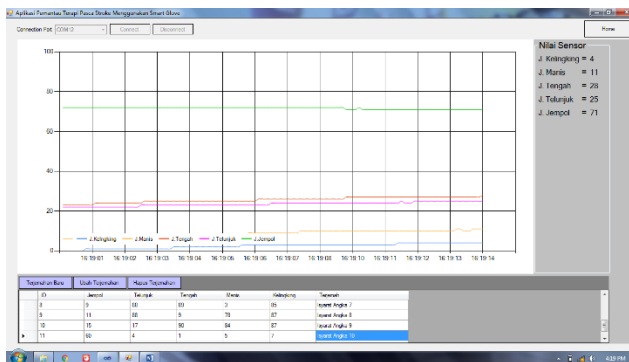
- Data Grafik Posisi Jari Menunjukkan Isyarat Angka 10

Data dari grafik pada posisi jari menunjukkan isyarat angka 10 adalah seperti ditunjukkan pada Gambar 9.24



**Gambar 9.24** Grafik Posisi Jari Menunjukkan Isyarat Angka 10

Adapun proses saat pengambilan data dapat dilihat pada Gambar 9.25

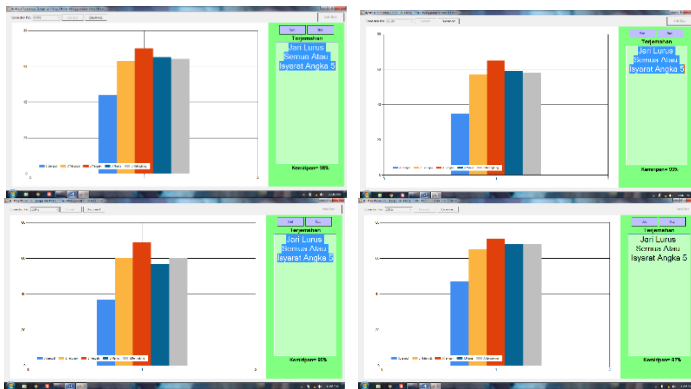


**Gambar 9.25** Proses Pengambilan Data Grafik Posisi Jari Menunjukkan Isyarat Angka 10

Dari data acuan diatas, selanjutnya dilakukan pengambilan data pada jari tangan dari beberapa orang yang berbeda-beda. Adapun hasil data dan grafik yang dapat dilihat pada beberapa gambar pada subbab selanjutnya.

- Data Hasil Pengujian Posisi Jari Lurus Semua atau Isyarat Angka 5.

Berikut adalah 4 buah grafik hasil pengambilan data dari jari-jari tangan orang yang berbeda untuk posisi jari lurus semua atau isyarat angka 5 dengan tingkat kemiripan yang berbeda dapat dilihat pada Gambar 9.26



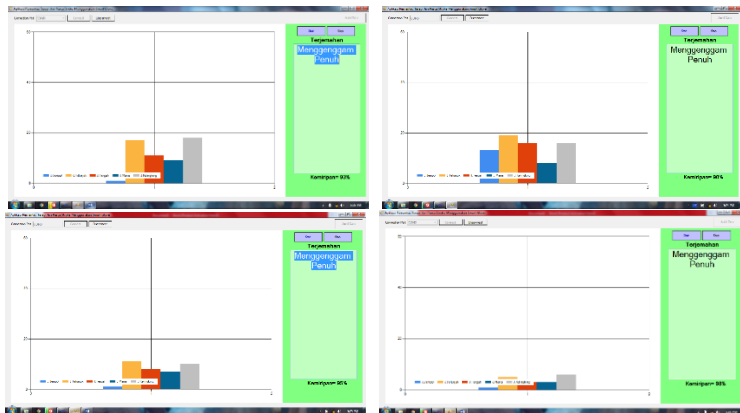
**Gambar 9.26** Grafik Data Hasil Posisi Jari Lurus Semua atau Isyarat Angka 5

Untuk terapi pada posisi jari lurus semua atau isyarat angka 5 adalah bertujuan untuk melatih otot pada setiap jari-jari tangan mulai dari jempol sampai kelingking agar dapat diluruskan pada

posisi lurus sempurna serta sebisa mungkin harus mengikuti lurus selayaknya jari-jari orang normal.

- Data Hasil Pengujian Posisi Jari Menggenggam Penuh.

Berikut adalah empat buah grafik hasil pengambilan data dari jari-jari tangan orang yang berbeda untuk posisi jari menggenggam penuh dengan tingkat kemiripan yang berbeda dapat dilihat pada Gambar 9.27



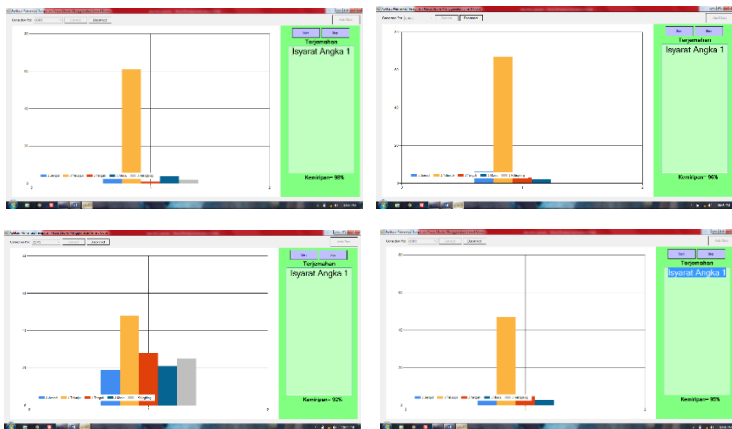
**Gambar 9.27** Grafik data hasil posisi jari menggenggam penuh

Untuk terapi pada posisi jari menggenggam penuh adalah bertujuan untuk melatih otot pada setiap jari-jari tangan mulai dari jempol sampai kelingking agar dapat ditekukkan secara penuh layaknya posisi menggenggam tangan pada orang normal. Dari data di atas dapat dilihat bahwa pada setiap percobaan terdapat perbedaan tingkat kemiripan dari masing-masing tangan. Adapun

data yang didapat adalah kemiripan berkisar 90% hingga 98%. Dengan demikian dapat disimpulkan bahwa semakin besar tingkat kemiripan gerakan jari seorang pasien, maka semakin baik pula hasil dari terapi jari yang dilakukan.

- **Data Hasil Pengujian Posisi Jari Isyarat Angka 1.**

Berikut adalah empat buah grafik hasil pengambilan data dari jari-jari tangan orang yang berbeda untuk posisi jari dengan isyarat angka 1 dengan tingkat kemiripan yang berbeda dapat dilihat pada Gambar 9.28



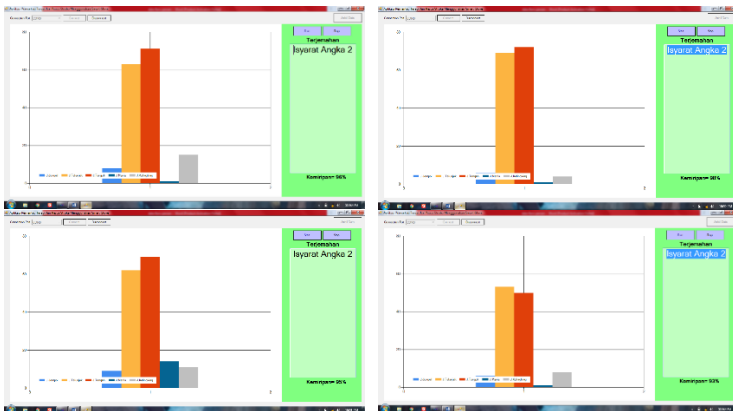
**Gambar 9.28** Grafik Data Hasil Posisi Jari Isyarat Angka 1

Untuk terapi pada posisi jari dengan isyarat angka 1 adalah bertujuan untuk melatih otot pada jari telunjuk yang harus diluruskan secara penuh serta jari jempol, jari tengah, jari manis dan jari

kelingking yang harus ditekuk secara penuh. Dari data di atas dapat dilihat bahwa pada setiap percobaan terdapat perbedaan tingkat kemiripan dari masing-masing tangan. Adapun data yang didapat adalah kemiripan berkisar 92% hingga 98%. Dengan demikian dapat disimpulkan bahwa semakin besar tingkat kemiripan gerakan jari seorang pasien, maka semakin baik pula hasil dari terapi jari yang dilakukan.

- **Data Hasil Pengujian Posisi Jari Isyarat Angka 2.**

Berikut adalah empat buah grafik hasil pengambilan data dari jari-jari tangan orang yang berbeda untuk posisi jari dengan isyarat angka 2 dengan tingkat kemiripan yang berbeda dapat dilihat pada Gambar 9.29



**Gambar 9.29** Grafik Data Hasil Posisi Jari Isyarat Angka 2



Untuk terapi pada posisi jari dengan isyarat angka 2 adalah bertujuan untuk melatih otot pada jari telunjuk dan jari tengah yang harus diluruskan secara penuh serta jari jempol, jari manis dan jari kelingking yang harus ditekuk secara penuh. Dari data di atas dapat dilihat bahwa pada setiap percobaan terdapat perbedaan tingkat kemiripan dari masing-masing tangan. Adapun data yang didapat adalah kemiripan berkisar 93% hingga 98%. Dengan demikian dapat disimpulkan bahwa semakin besar tingkat kemiripan gerakan jari seorang pasien, maka semakin baik pula hasil dari terapi jari yang dilakukan.

- **Data Hasil Pengujian Posisi Jari Isyarat Angka 3.**

Berikut adalah empat buah grafik hasil pengambilan data dari jari-jari tangan orang yang berbeda untuk posisi jari dengan isyarat angka 3 dengan tingkat kemiripan yang berbeda dapat dilihat pada Gambar 9.30

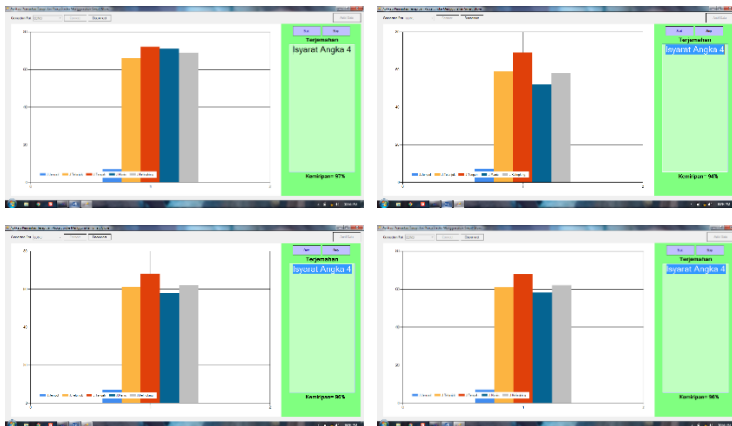


**Gambar 9.30** Grafik Data Hasil Posisi Jari Isyarat Angka 3

Untuk terapi pada posisi jari dengan isyarat angka 3 adalah bertujuan untuk melatih otot pada jari telunjuk, jari tengah dan jari jempol yang harus diluruskan secara penuh serta jari manis dan jari kelingking yang harus ditebuk secara penuh. Dari data di atas dapat dilihat bahwa pada setiap percobaan terdapat perbedaan tingkat kemiripan dari masing-masing tangan. Adapun data yang didapat adalah kemiripan berkisar 92% hingga 99%. Dengan demikian dapat disimpulkan bahwa semakin besar tingkat kemiripan gerakan jari seorang pasien, maka semakin baik pula hasil dari terapi jari yang dilakukan.

- Data Hasil Pengujian Posisi Jari Isyarat Angka 4.

Berikut adalah empat buah grafik hasil pengambilan data dari jari-jari tangan orang yang berbeda untuk posisi jari dengan isyarat angka 4 dengan tingkat kemiripan yang berbeda dapat dilihat pada Gambar 9.31



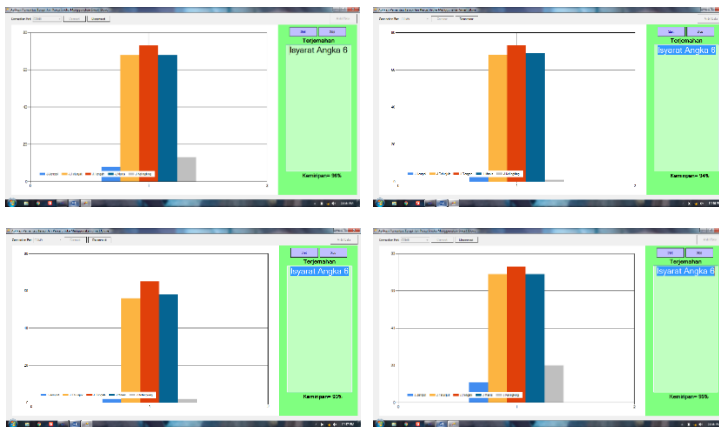
**Gambar 9.31** Grafik Data Hasil Posisi Jari Isyarat Angka 4

Untuk terapi pada posisi jari dengan isyarat angka 4 adalah bertujuan untuk melatih otot pada jari telunjuk, jari tengah, jari manis dan jari kelingking yang harus diluruskan secara penuh serta jari jempol yang harus ditekuk secara penuh. Dari data di atas dapat dilihat bahwa pada setiap percobaan terdapat perbedaan tingkat kemiripan dari masing-masing tangan. Adapun data yang didapat adalah kemiripan berkisar 94% hingga 97%. Dengan demikian dapat

disimpulkan bahwa semakin besar tingkat kemiripan gerakan jari seorang pasien, maka semakin baik pula hasil dari terapi jari yang dilakukan.

- Data Hasil Pengujian Posisi Jari Isyarat Angka 6.

Berikut adalah empat buah grafik hasil pengambilan data dari jari-jari tangan orang yang berbeda untuk posisi jari dengan isyarat angka 6 dengan tingkat kemiripan yang berbeda dapat dilihat pada Gambar 9.32



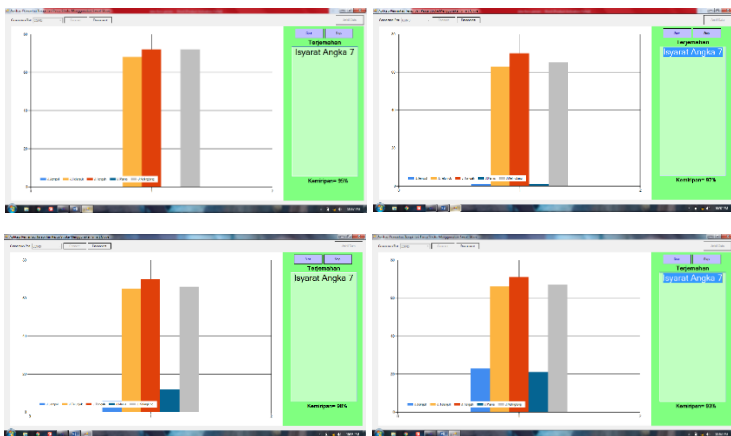
**Gambar 9.32** Grafik Data Hasil Posisi Jari Isyarat Angka 6

Untuk terapi pada posisi jari dengan isyarat angka 6 adalah bertujuan untuk melatih otot pada jari telunjuk, jari tengah dan jari manis yang harus diluruskan secara penuh serta jari jempol dan jari kelingking yang harus ditekuk secara penuh. Dari data di atas dapat

dilihat bahwa pada setiap percobaan terdapat perbedaan tingkat kemiripan dari masing-masing tangan. Adapun data yang didapat adalah kemiripan berkisar 93% hingga 96%. Dengan demikian dapat disimpulkan bahwa semakin besar tingkat kemiripan gerakan jari seorang pasien, maka semakin baik pula hasil dari terapi jari yang dilakukan.

- **Data Hasil Pengujian Posisi Jari Isyarat Angka 7.**

Berikut adalah empat buah grafik hasil pengambilan data dari jari-jari tangan orang yang berbeda untuk posisi jari dengan isyarat angka 7 dengan tingkat kemiripan yang berbeda dapat dilihat pada Gambar 9.33

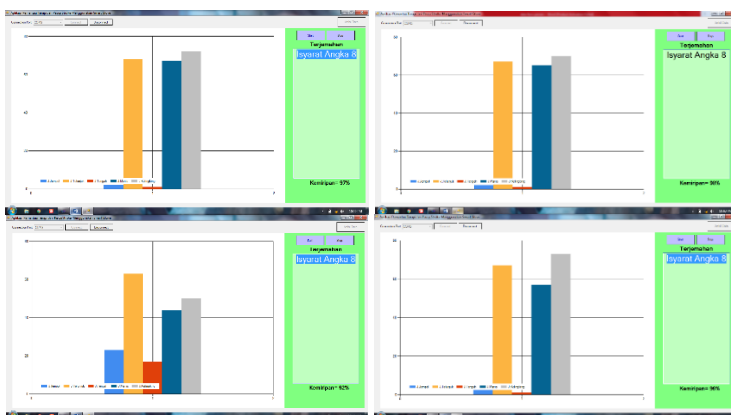


**Gambar 9.33** Grafik Data Hasil Posisi Jari Isyarat Angka 7

Untuk terapi pada posisi jari dengan isyarat angka 7 adalah bertujuan untuk melatih otot pada jari telunjuk, jari tengah dan jari kelingking yang harus diluruskan secara penuh serta jari jempol dan jari manis yang harus ditekuk secara penuh. Dari data di atas dapat dilihat bahwa pada setiap percobaan terdapat perbedaan tingkat kemiripan dari masing-masing tangan. Adapun data yang didapat adalah kemiripan berkisar 93% hingga 98%. Dengan demikian dapat disimpulkan bahwa semakin besar tingkat kemiripan gerakan jari seorang pasien, maka semakin baik pula hasil dari terapi jari yang dilakukan.

- **Data Hasil Pengujian Posisi Jari Isyarat Angka 8.**

Berikut adalah empat buah grafik hasil pengambilan data dari jari-jari tangan orang yang berbeda untuk posisi jari dengan isyarat angka 8 dengan tingkat kemiripan yang berbeda dapat dilihat pada Gambar 9.34

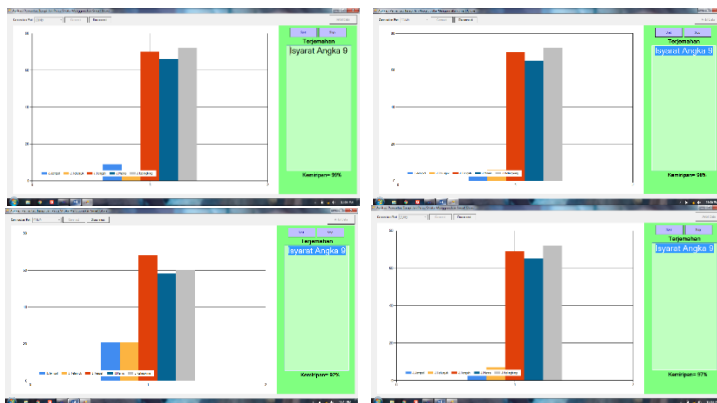


**Gambar 9.34** Grafik Data Hasil Posisi Jari Isyarat Angka 8

Untuk terapi pada posisi jari dengan isyarat angka 8 adalah bertujuan untuk melatih otot pada jari telunjuk, jari manis dan jari kelingking yang harus diluruskan secara penuh serta jari jempol dan jari tengah yang harus ditekuk secara penuh. Dari data di atas dapat dilihat bahwa pada setiap percobaan terdapat perbedaan tingkat kemiripan dari masing-masing tangan. Adapun data yang didapat adalah kemiripan berkisar 92% hingga 98%. Dengan demikian dapat disimpulkan bahwa semakin besar tingkat kemiripan gerakan jari seorang pasien, maka semakin baik pula hasil dari terapi jari yang dilakukan.

- Data Hasil Pengujian Posisi Jari Isyarat Angka 9.

Berikut adalah empat buah grafik hasil pengambilan data dari jari-jari tangan orang yang berbeda untuk posisi jari dengan isyarat angka 9 dengan tingkat kemiripan yang berbeda dapat dilihat pada Gambar 9.35



**Gambar 9.35** Grafik Data Hasil Posisi Jari Isyarat Angka 9

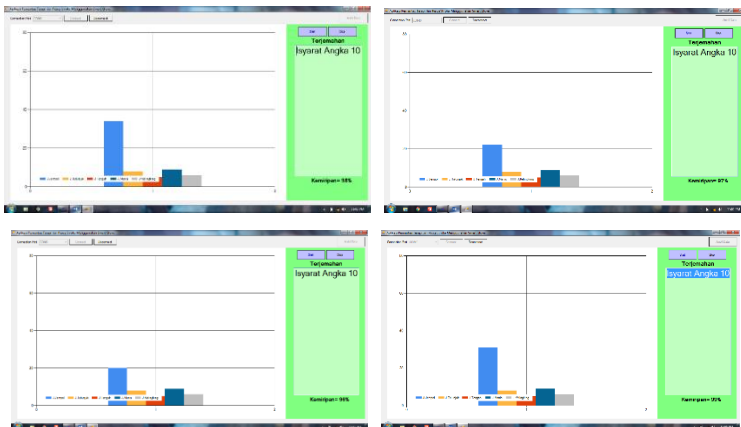
Untuk terapi pada posisi jari dengan isyarat angka 9 adalah bertujuan untuk melatih otot pada jari tengah, jari manis dan jari kelingking yang harus diluruskan secara penuh serta jari jempol dan jari telunjuk yang harus ditekuk secara penuh. Dari data di atas dapat dilihat bahwa pada setiap percobaan terdapat perbedaan tingkat kemiripan dari masing-masing tangan. Adapun data yang didapat adalah kemiripan berkisar 92% hingga 99%. Dengan demikian dapat disimpulkan bahwa semakin besar tingkat kemiripan gerakan jari



seorang pasien, maka semakin baik pula hasil dari terapi jari yang dilakukan.

- Data Hasil Pengujian Posisi Jari Isyarat Angka 10.

Berikut adalah empat buah grafik hasil pengambilan data dari jari-jari tangan orang yang berbeda untuk posisi jari dengan isyarat angka 10 dengan tingkat kemiripan yang berbeda dapat dilihat pada Gambar 9.36



**Gambar 9.36** Grafik Data Hasil Posisi Jari Isyarat Angka 10

Untuk terapi pada posisi jari dengan isyarat angka 10 adalah bertujuan untuk melatih otot pada jari jempol yang harus diluruskan secara penuh sedangkan jari telunjuk jari tengah, jari manis dan jari kelingking yang harus ditekuk secara penuh. Dari data di atas dapat dilihat bahwa pada setiap percobaan terdapat perbedaan tingkat

kemiripan dari masing-masing tangan. Adapun data yang didapat adalah kemiripan berkisar 96% hingga 99%. Dengan demikian dapat disimpulkan bahwa semakin besar tingkat kemiripan gerakan jari seorang pasien, maka semakin baik pula hasil dari terapi jari yang dilakukan.

Gerakan terapi jari yang dapat dideteksi oleh sistem adalah gerakan isyarat angka 1,2,3,4,5,6,7,8,9 dan 10 serta isyarat menggenggam penuh dan jari lurus semua. Sementara untuk terapi jari yang tidak berhasil adalah dikarenakan terdapatnya error saat pembacaan semua jari diluruskan, hal ini disebabkan oleh isyarat yang diterjemah merupakan posisi yang sama nilainya dengan isyarat angka 5. Maka dari itu untuk isyarat angka 5 kemudian digabungkan menjadi dua makna yaitu sama dengan jari lurus semua.

Penyebab dari terbatasnya gerakan terapi jari adalah adanya beberapa gerakan terapi jari yang serupa atau yang nilai keluaran dari tiap-tiap sensornya tidak mempunyai perbedaan yang signifikan, sehingga data yang terbaca oleh sensor memiliki nilai yang hampir sama dan sulit dibedakan.

Dari 20 kali percobaan, didapat data hasil 80% keberhasilan atau kemiripan data gerakan jari-jari tangan dengan nilai kemiripan diatas 90%. Dengan demikian dapat disimpulkan bahwa tingkat

keberhasilan dari penelitian ini adalah ketika pengguna sarung tangan pintar / pasien terapi stroke ini berhasil atau mampu mencapai kemiripan gerakan terapi jari dengan nilai kemiripan setinggi mungkin (maksimal 100%).

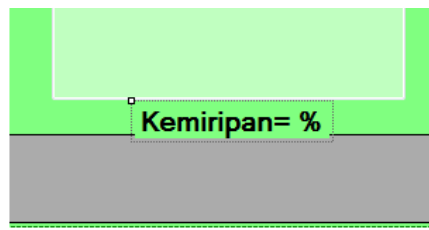
Semakin rendah nilai kemiripan yang dapat dicapai, maka terapi harus terus diberikan kepada pasien. Begitu pula dengan sebaliknya, semakin tinggi nilai kemiripan yang dapat dicapai oleh pasien maka tingkat keberhasilan dari terapi ini akan semakin baik.

### 9.5.3 Pengujian Software Secara Keseluruhan

Pengujian *software* atau perangkat lunak berupa aplikasi yang dibangun menggunakan *microsoft visual basic* 2010 telah berjalan sebagaimana mestinya. Hal ini dapat dibuktikan dengan berhasilnya aplikasi yang telah dibangun untuk menampilkan grafik secara real time pada tampilan komputer/PC. Dengan demikian maka dapat disimpulkan bahwa sistem telah berjalan sebagaimana seperti yang peneliti harapkan.

Untuk tingkat keberhasilan pada penelitian ini dapat dilihat pada angka tingkat kemiripan yang digunakan. Tingkat kemiripan yang dimaksud dapat dilihat pada aplikasi yang telah dibangun. Dengan nilai tingkat persentase kemiripan yang dapat dilihat dari nilai minimum (0%) sampai dengan nilai maksimum (100%). Untuk

nilai tingkat kemiripan itu sendiri, dapat naik atau turun secara bertahap sebesar 1%. Dimana dapat disimpulkan bahwa semakin besar nilai persentase kemiripan yang dapat dicapai oleh pasien terapi, maka hasil terapi akan semakin baik. Artinya pengguna sarung tangan ini mampu meniru gerakan jari-jari tangan dengan sempurna sesuai dengan data masukan yaitu data dari jari-jari tangan orang normal. Nilai tingkat kemiripan yang dapat dibaca oleh sistem pada aplikasi ini dapat dilihat pada Gambar 9.37.



**Gambar 9.37** Label Penunjuk Persentase Kemiripan

## Referensi

Barrett, S. F. (2010). *Arduino Microcontroller Processing for Everyone!*. Morgan and Claypool Publishers.

Djuandi, F. (2011). Pengenalan arduino. *E-book*. *www.tobuku*, 1-24.

Faludi, R. (2010). *Building wireless sensor networks: with ZigBee, XBee, arduino, and processing*. " O'Reilly Media, Inc."

<https://www.arduino.cc/>

<https://processing.org/>

Kadir, A. (2015). Buku Pintar Pemrograman Arduino. *Penerbit Mediacom, Yogyakarta*.

McRoberts, M. (2013). *Beginning Arduino*. Apress.

Noble, J. (2009). *Programming interactivity: a designer's guide to Processing, Arduino, and OpenFrameworks*. " O'Reilly Media, Inc."

Pokress, S. C., & Veiga, J. J. D. (2013). MIT App Inventor: Enabling personal mobile computing. *arXiv preprint arXiv:1310.2830*.

Yudhana, A., & Putra, M. D. D. (2018). Rancang Bangun Sistem Pemantauan Infus Berbasis Android. *Transmisi*, 20(2), 91-95.

Yudhana, A., Purnama, H. S., Ramadhani, M., & Subrata, A. C. (2018). Otomasi dan Instrumentasi untuk Smart Farming dan Smart Glove.

## Profil Penulis



### **Anton Yudhana, S.T., M.T., Ph.D.**

Lahir Purworejo pada tanggal 8 Agustus 1976. Beliau menyelesaikan studi S1 Teknik Elektro di Institut Teknologi Sepuluh November (ITS) Surabaya (2001) konsentrasi Telekomunikasi Multimedia, S2 Teknik Elektro Universitas Gadjah Mada (2005) konsentrasi Pengolahan Sinyal, dan S3 untuk Jurusan Teknik Elektro konsentrasi *Radio Communication* dan *Signal Processing* di Universiti Teknologi Malaysia. Sejak 2001 beliau aktif mengajar di Program Studi Teknik Elektro Universitas Ahmad Dahlan, Yogyakarta. Saat ini beliau juga aktif mengajar di S2 Teknik Informatika di Universitas Ahmad Dahlan untuk Mata Kuliah Komunikasi Data dan Jaringan Komputer. Penelitian-penelitian yang dilakukan selama lima tahun terakhir berfokus kepada aplikasi-aplikasi otomasi dan komunikasi di bidang pertanian dan kesehatan. Penelitian tersebut telah mengantarkannya dalam program *research collaboration* dengan universitas di luar negeri diantaranya *Scheme for Academic Mobility Exchange (SAME)* yang diselenggarakan Dikti dan Kemdikbud pada tahun 2013 di Massey University, New Zealand untuk topik penelitian *Wireless Sensor Network*. Penelitian yang dilakukan pada tahun 2017 bekerja sama dengan Iwate Prefectural University, Japan di bidang smart sensor untuk difabel dan penderita stroke. Penelitian terbaru (2019) tentang *Heart Beat Sensor* dan *Sinyal EEG* bekerja sama dengan Karabuk University, Turkey dan Macquarie University Sydney, Australia telah menjadi bahasan dalam buku ini.

Penulis dapat dihubungi melalui email [eyudhana@ee.uad.ac.id](mailto:eyudhana@ee.uad.ac.id)



**Hendril Satrian Purnama, S.T.**

Lahir di Lombok Timur, pada tanggal 30 Desember 1994. Telah menyelesaikan studi S1 Teknik Elektro di Universitas Ahmad Dahlan Yogyakarta, pada tahun 2017 dengan konsentrasi Otomasi Industri, Sejak tahun 2017 ia telah menjadi asisten peneliti dibawah program studi Teknik Elektro Universitas Ahmad Dahlan, Yogyakarta. Selain itu ia juga aktif menjadi asistant editor pada beberapa Jurnal Internasional dalam bidang teknik elektro, komputer dan informatika. Minat penelitiannya meliputi bidang elektronika daya, pengembangan energi terbarukan dan robotika.

Penulis dapat dihubungi melalui email: [lfriyan220@gmail.com](mailto:lfriyan220@gmail.com)