

# **RANCANG BANGUN ALAT PEREKAM FREKUENSI LISTRIK JANTUNG PORTABLE MENGGUNAKAN GROVE CHEST STRAP HEART RATE SENSOR DENGAN ARDUINO UNO**

**Nenny Anggraini**

nenny\_a@yahoo.com

Dosen Jurusan Teknik Informatika, UIN Syarif Hidayatullah Jakarta

## **ABSTRAK**

Jantung merupakan pemompa berotot yang bekerja secara terus menerus dan tanpa berhenti untuk memompa darah manusia yang dialirkan ke seluruh tubuh, jantung berkontraksi atau berdenyut secara berirama akibat potensial aksi yang ditimbulkannya sendiri, potensial aksi ini dicetuskan oleh nodus-nodus *pacemaker* yang terdapat di jantung dan dipengaruhi oleh beberapa jenis elektrolit seperti  $K^+$ ,  $Na^+$ , dan  $Ca^{++}$ . EKG adalah rekaman mengenai aktivitas listrik di cairan tubuh yang dirangsang oleh aliran listrik jantung yang mencapai permukaan tubuh, bagi sebagian orang yang menderita penyakit jantung pemeriksaan EKG sangat dibutuhkan, pemeriksaan secara berkala tentunya membutuhkan biaya serta memakan waktu karena harus bolak balik ke dokter atau rumah sakit, Untuk mengatasi permasalahan tersebut maka dibutuhkan alat *portable* yang memiliki kemudahan dalam pengoperasiannya dan dapat digunakan di rumah. Untuk itu Penulis mempunyai ide untuk merancang dan membuat sebuah Alat perekam listrik jantung *portable* berbasis Microcontroller menggunakan *Sensor Grove Chest Strap Heart Rate* yang mampu menangkap sinyal listrik jantung. Dengan menggunakan pengantarmukaan paralel pada komunikasinya, akan mengubah sinyal analog menjadi sinyal digital yang akan diteruskan ke komputer melalui perangkat Arduino UNO. Komunikasi antara sensor dengan Arduino UNO menggunakan gelombang radio AM (nirkabel) dengan menggunakan Bahasa pemrograman C yang digunakan untuk menampilkan data perekam frekuensi listrik jantung dalam bentuk *stream* pada *serial port*, sedangkan *software processing* dengan bahasa pemrograman Java untuk mengubah data serial tersebut menjadi grafik dari frekuensi listrik jantung yang nantinya dapat dihitung tingginya frekuensi dari listrik jantung tersebut dalam satuan bpm (*beats per minute*).

**Kata Kunci:** *frekuensi listrik jantung, sensor grove chest strap heart rate, arduinouno*

## **I. PENDAHULUAN**

Jantung adalah salah satu organ manusia yang letaknya di dalam rongga dada sebelah kiri yang terdiri dari organ berupa otot, berbentuk kerucut, berongga dan dengan basisnya di atas dan puncaknya di bawah. Jantung merupakan pemompa berotot yang bekerja secara terus menerus dan tanpa berhenti untuk memompa darah manusia yang dialirkan ke seluruh tubuh. Sistem listrik jantung adalah Jantung yang berkontraksi atau berdenyut secara berirama akibat potensial aksi yang ditimbulkannya sendiri, hal ini disebabkan karena jantung memiliki mekanisme aliran listrik yang dicetuskannya sendiri guna berkontraksi atau memompa dan berelaksasi. Potensial aksi ini dicetuskan

oleh nodus-nodus *pacemaker* yang terdapat di jantung dan dipengaruhi oleh beberapa jenis elektrolit seperti  $K^+$ ,  $Na^+$ , dan  $Ca^{++}$ . Gangguan terhadap kadar elektrolit tersebut di dalam tubuh dapat mengganggu mekanisme pembentukan dan penghantaran listrik jantung. Sumber listrik Jantung adalah *SA Node (Nodus Sinoatrial)*. Arus listrik yang dihasilkan oleh otot jantung menyebar ke jaringan di sekitar jantung dan dihantarkan melalui cairan-cairan tubuh. Sebagian kecil aktivitas listrik ini mencapai permukaan tubuh dan dapat dideteksi menggunakan alat khusus. Rekaman aliran listrik jantung disebut dengan elektrokardiogram atau EKG. EKG adalah rekaman mengenai aktivitas listrik di cairan tubuh yang dirangsang oleh aliran listrik

jantung yang mencapai permukaan tubuh. Jadi EKG bukanlah rekaman langsung aktivitas listrik jantung yang sebenarnya. Siklus jantung bekerja secara berselang seling dalam hal kontraksi (memompa). Kerja jantung dalam berkontraksi untuk mengosongkan isi jantung dan juga berelaksasi dalam rangka mengisi darah kembali. Siklus jantung terdiri atas periode

Tes EKG itu dilakukan untuk beberapa keperluan antara lain:

- Memeriksa aktivitas listrik jantung
- Menemukan penyebab nyeri dada, yang dapat disebabkan serangan jantung, inflamasi kantung sekitar jantung (perikarditis).
- Menemukan penyebab gejala penyakit jantung, seperti detak jantung lebih cepat atau tidak beraturan (Aritmia).
- Mengetahui apakah dinding ruang-ruang jantung terlalu tebal.
- Memeriksa seberapa baik kerja suatu obat dan apakah obat tersebut memiliki efek samping terhadap jantung.
- Memeriksa apakah suatu alat mekanis yang dicangkok dalam jantung, misalnya *pacemaker*, bekerja dengan baik untuk mengendalikan denyut jantung.
- Memeriksa kesehatan jantung pada penderita penyakit atau kondisi tertentu, seperti hipertensi, kolesterol tinggi, diabetes, atau penyakit lainnya.

Bagi sebagian orang pemeriksaan EKG secara berkala tentunya membutuhkan biaya serta memakan waktu karena harus bolak balik ke Dokter atau rumah sakit, Untuk mengatasi permasalahan tersebut maka dibutuhkan alat *portable* yang memiliki kemudahan dalam pengoperasiannya dan dapat digunakan di rumah. Untuk itu Penulis mempunyai ide untuk merancang dan membuat sebuah alat perekam listrik jantung *portable* berbasis *Microcontroller* menggunakan *Grove Chest Strap Heart Rate Sensor* yang mampu menangkap sinyal listrik jantung, Dengan menggunakan pengantarmukaan paralel pada komunikasinya, akan mengubah sinyal analog menjadi sinyal digital yang akan diteruskan ke komputer melalui perangkat Arduino UNO. Komunikasi antara sensor dengan Arduino UNO menggunakan gelombang radio AM (nirkabel) dengan menggunakan Bahasa

sistol (kontraksi dan pengosongan isi) dan juga periode diastol (relaksasi dan pengisian jantung). Atrium dan ventrikel mengalami siklus sistol dan diastol terpisah. Kontraksi terjadi akibat penyebaran eksitasi (mekanisme listrik jantung) ke seluruh jantung. Sedangkan relaksasi timbul setelah repolarisasi atau tahapan relaksasi dari otot jantung.

programan C yang digunakan untuk menampilkan data perekam frekuensi listrik jantung dalam bentuk *stream* pada *serial port*, sedangkan *software processing* dengan bahasa pemrograman Java untuk mengubah data serial tersebut menjadi grafik dari frekuensi listrik jantung yang nantinya dapat dihitung tingginya frekuensi dari listrik jantung tersebut dalam satuan bpm (*beats per minute*).

Permasalahan penelitian yang penulis angkat adalah Bagaimana merancang dan membuat sebuah alat *portable* yang mampu merekam Frekuensi sinyal listrik jantung dengan menggunakan *Grove Chest Strap Heart Rate Sensor* berbasis *Microcontroller* Arduino Uno.

Merancang dan membuat alat perekam frekuensi listrik jantung *portable* dengan hasil tampilan dalam bentuk angka dan grafik di komputer atau laptop.

## II. METODOLOGI PENELITIAN

### 2.1 Metode Pengumpulan Data

#### 2.1.1 Observasi

Pada metode ini, peneliti mengumpulkan data dan informasi yaitu dengan cara meninjau dan melakukan pengamatan secara langsung ke lapangan terhadap suatu kegiatan yang sedang dilakukan atau berjalan, untuk memperoleh informasi mengenai kebutuhan sistem (*system requirements*).

Pada penelitian ini peneliti melakukan observasi pada alat-alat kedokteran yang mempunyai fungsi untuk menghitung frekuensi sinyal jantung seperti EKG, Penulis juga membaca literatur tentang cara kerja dari EKG, sampai akhirnya penulis mempunyai ide untuk membuat suatu alat yang bisa menghitung frekuensi sinyal jantung yang *portable* sehingga pengguna bisa menggunakan alat tersebut sendiri.

### 2.1.2 Studi Pustaka dan Literatur Sejenis

Peneliti mempelajari tiap konsep yang berkaitan dengan topik bahasan penelitian dengan cara membaca buku-buku referensi, *e-book*, artikel, jurnal dan *website*. Adapun sumber-sumber tersebut dapat dilihat selengkapnya pada daftar pustaka.

Penelitian studi literatur yang dilakukan pada hasil penulisan karya ilmiah, yaitu menekankan pada kelebihan dan kekurangan yang dilihat dari sisi sistem yang telah dirancang. Sebagai sumber referensi dan bahan acuan terhadap sistem yang akan dibuat. Studi literatur hasil dari penelitian atau hasil penulisan karya ilmiah di antaranya:

Tabel 1. Perbandingan Studi Literatur

No	Nama peneliti	Judul Penelitian	Kelebihan	Kekurangan
1	Wahab, Wahidin, Budiono, Mintardja Indra  Perpustakaan Universitas Indonesia UI - Laporan Penelitian	Perancangan alat detektor denyut jantung berbasis mikrokontroler 8031	+ Dapat menghitung denyut nadi per menit	- Masih menggunakan Mikrokontroler 8031 - Bahasa pemrograman yg digunakan hanya terbatas utk ruang lingkup mesin 8031 - Menggunakan sensor Fotodioda utk mendeteksi cahaya lewat ujung jari - Alat tidak <i>portable</i> dan belum ada tampilan grafik frekuensinya
2.	SERTU EDY SUSANTO ELKA  <a href="http://lemjiantek.mil.id/article-110-rancang-bangun-alat-penghitung-detak-nadi.html">http://lemjiantek.mil.id/ article-110-rancang- bangun-alat-penghitung- detak-nadi.html</a>	Rancang Bangun Penghitung Detak Nadi Menggunakan Sadapan Ujung Jari dengan Tampilan LCD untuk Seleksi Penerimaan Calon Prajurit	+ Dapat menghitung denyut nadi kurang dari satu menit  +Alat ini dimanfaatkan oleh satuan Kesehatan militer dan dapat diaplikasikan untuk seleksi penerimaan calon prajurit TNI AD.	- Masih menggunakan Mikrokontroler AT8951 - Bahasa pemrograman yg digunakan hanya terbatas utk ruang lingkup mesin AT8951 - Menggunakan <i>LED superbright</i> dan sensor Fotodioda utk mendeteksi cahaya lewat ujung jari - Alat tidak <i>portable</i> dan belum ada tampilan grafik frekuensinya
3.	Agung, I Gusti Agung Putu Raka, Institusi Fakultas Teknik Universitas UdayanaTahun Terbit2008	Perencanaan Dan Realisasi Alat Penghitung Jumlah Detak Jantung Berbasis Mikrokontroler AT89S52	+ Dapat menghitung frekuensi detak jantung per menit  +Menggunakan sensor berupa elektrode, penguat dan filter serta dilengkapi dengan RTC Ds12887 sebagai sumber waktu konstan dan bersifat <i>real time</i> .	- Masih menggunakan Mikrokontroler AT89S52 - Bahasa pemrograman yg digunakan hanya terbatas utk ruang lingkup mesin AT89S52 - Alat tidak <i>portable</i> dan belum ada tampilan grafik frekuensinya

### 2.2 Metode Pengembangan Sistem

Pengembangan sistem yang dilakukan menggunakan tiga tahap siklus pengembangan model RAD (*Rapid Application Development*), yaitu fase

perencanaan syarat-syarat, fase perancangan sistem (*Workshop Design*), fase implementasi. Model RAD digunakan karena melihat dari aplikasi yang dikembangkan merupakan aplikasi yang sederhana dan tidak membutuhkan waktu

yang lama, metode RAD adalah metode yang diperuntukkan untuk jangka waktu yang pendek sesuai dengan aplikasi yang akan dibuat.

Adapun dalam tahap pengembangan sistem RAD (*Rapid Application Development*) ini terdiri dari beberapa aktivitas yang tentunya sesuai dengan tahapan yang telah dijabarkan pada alur proses pengembangan sistem. Tahapan tersebut yaitu (Kendall & Kendall, 2010):

### 2.2.1 Fase Perencanaan Syarat-syarat

Dalam fase ini peneliti melakukan beberapa kegiatan di antaranya:

- 1) Identifikasi masalah  
Bertujuan untuk menjelaskan masalah yang ada, sehingga penelitian ini dibuat.
- 2) Usulan Penyelesaian Masalah  
Merupakan usulan penyelesaian dari masalah-masalah yang telah dijelaskan pada poin 1 (Identifikasi masalah).
- 3) Analisis Rancang Bangun Alat Perekam Frekuensi Listrik Jantung *portable* menggunakan *grove chest strap heart rate sensor* dengan Arduino uno  
Bertujuan untuk melakukan perancangan terhadap alat yang akan dibuat

### 2.2.2 Perancangan Sistem (*Workshop design*)

Pada tahap ini peneliti mendesain perancangan Rancang Bangun Alat Perekam Frekuensi Listrik Jantung *portable* menggunakan *grove chest strap heart rate sensor* dengan Arduino uno yang diusulkan agar dapat berjalan dengan lebih baik. Tahapan yang dilakukan antara lain:

- 1) Perancangan Layout Rangkaian Lengkap Alat perekam frekuensi jantung *portable*
- 2) Perancangan tampilan antarmuka (*interface*), Pada tahap ini penulis mendesign tampilan dari Alat perekam frekuensi jantung *portable*, yang meliputi tampilan alat secara *hardware* dan tampilan *interface* pada hasil output pada program yang dihasilkan.
- 3) Pengkodean  
Pada tahap ini dilakukan pembuatan program dengan menggunakan pengantarmukaan paralel pada

komunikasinya, yang akan mengubah sinyal analog menjadi sinyal digital yang selanjutnya diteruskan ke komputer melalui perangkat Arduino UNO. Komunikasi antara sensor dengan Arduino UNO menggunakan gelombang radio AM (nirkabel) dengan menggunakan Bahasa pemrograman C yang digunakan untuk menampilkan data perekam frekuensi listrik jantung dalam bentuk *stream* pada *serial port*, sedangkan *software processing* dengan bahasa pemrograman Java untuk mengubah data serial tersebut menjadi grafik dari frekuensi listrik jantung yang nantinya dapat dihitung tingginya frekuensi dari listrik jantung tersebut dalam satuan bpm (*beats per minute*).

### 2.2.3 Fase Implementasi

Pada tahap ini peneliti melakukan beberapa tahap implementasi di antaranya:

- 1) Spesifikasi *Software* dan *Hardware* Perangkat keras (*hardware*) dan perangkat lunak (*software*) yang dibutuhkan untuk menjalankan system perekam frekuensi jantung *portable* ini.
- 2) Instalasi ke *Software* Arduino  
Pada tahap ini, peneliti melakukan instalasi *Software* Arduino, hal ini dimaksudkan supaya computer mengenali Alat perkan frekuensi jantung.
- 3) Implementasi Rancang Bangun Alat perekam frekuensi Jantung *Portable*  
Tahap ini merupakan implelementasi Rancang Bangun Alat perekam frekuensi Jantung *Portable* yang akan diujikan dengan memakaikan sensor ke dada pengguna yang akan di rekam frekuensi jantungnya dan selanjutnya akan terlihat di layar LCD berapa frekuensi jantung yang dihasilkan dan akan terlihat grafik frekuensi dari pemeriksaan.
- 4) Implementasi antarmuka  
Tahap ini merupakan desain antarmuka (*user interface*) yang digunakan dalam rancang bangun alat perekam frekuensi jantung, desain antarmuka ini meliputi desain secara perangkat keras dan perangkat lunak

- 5) Pengujian  
Pada tahapan ini dilakukan pengujian terhadap Rancang bangun alat perekam frekuensi jantung yang sudah di desain pada tahap sebelumnya.

### 2.3 Alasan Menggunakan RAD

Dari lima macam model-model perangkat lunak yang dikemukakan oleh Roger S.Pressman, dipilih model RAD sebagai metode pengembangan system dengan alasan sebagai berikut:

1. Aplikasi yang dirancang dan dikembangkan merupakan aplikasi yang sederhana dan tidak memerlukan waktu yang lama. Hal ini sesuai dengan tujuan dari model RAD yang dikemukakan oleh Kenneth E. Kendall dan Julie E. Kendall yaitu RAD digunakan untuk

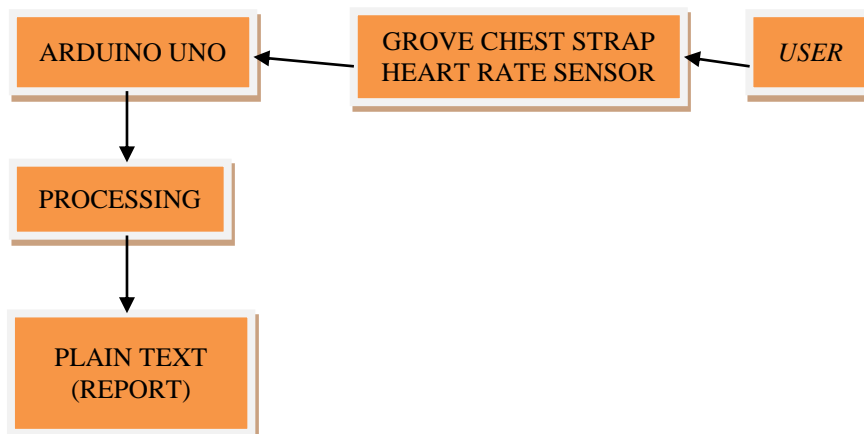
mempersingkat waktu antara perancangan dan penerapan sistem informasi (Kendall dan Kendall, 2010).

2. Dengan menggunakan metode RAD akan dicapai suatu system fungsional yang utuh dalam periode waktu yang sangat pendek jika kebutuhan dapat dipahami dengan baik (Pressman, 2002)
3. Kekurangan dalam pengembangan metode formal antara lain banyak memakan waktu dan mahal (Pressman, 2002). Aplikasi ini sederhana dan tidak memakan waktu dan biaya.

## III. HASIL DAN PEMBAHASAN

### 3.1 Rancang Bangun Rangkaian Perekam Frekuensi Jantung *Portable*

Rancang bangun yang akan dikembangkan oleh peneliti, digambarkan pada gambar di bawah ini:

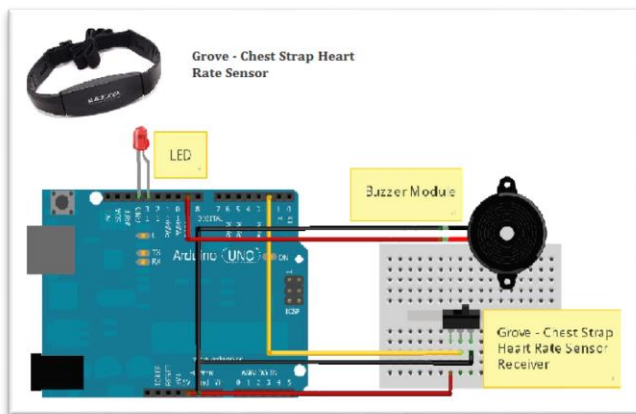


Gambar 1. Rancang Bangun

Pada blok diagram rancang bangun di atas, proses perekaman frekuensi jantung dimulai dengan *user* yang akan diperiksa akan memakai *Grove Chest Strap Heart Rate Sensor* yang akan dipakai pada dada *user* yang akan diperiksa, selanjutnya alat perekam frekuensi jantung yang dijematani oleh mesin Arduino akan menginisialisasi supaya frekuensi jantung *user* bisa dideteksi dari sensor ke mesin arduino, kalo sudah terdeteksi maka hasil dari proses perekaman frekuensi jantung akan terlihat pada tampilan berupa grafik dan *report* yang menunjukkan rekaman frekuensi jantung *user* tersebut, secara teknis Alat perekam frekuensi listrik jantung *portable* berbasis *Microcontroller* menggunakan *Grove Chest Strap Heart Rate Sensor* akan menangkap sinyal listrik jantung, dengan menggunakan

pengantarmukaan paralel pada komunikasinya, akan mengubah sinyal analog menjadi sinyal digital yang akan diteruskan ke komputer melalui perangkat Arduino UNO. Komunikasi antara sensor dengan Arduino UNO menggunakan gelombang radio AM (nirkabel) dengan menggunakan Bahasa pemrograman C yang digunakan untuk menampilkan data perekam frekuensi listrik jantung dalam bentuk *stream* pada *serial port*, sedangkan *software processing* dengan bahasa pemrograman Java untuk mengubah data serial tersebut menjadi grafik dari frekuensi listrik jantung yang nantinya dapat dihitung tingginya frekuensi dari listrik jantung tersebut dalam satuan bpm (*beats per minute*).

### 3.2 Perancangan Skematik Alat Pererekam Listrik Jantung



Gambar 2. Skema Alat Pererekam Listrik Jantung

### 3.3 Implementasi sistem

Pada tahap ini penulis melakukan beberapa tahap implementasi di antaranya: Implementasi Rangkaian ini berdasarkan pada skematik rangkaian yang sudah dilakukan sebelumnya, peralatan yang dibutuhkan adalah:

- Arduino UNO
- Grove - Chest Strap Heart Rate Sensor
- Kabel
- Kabel USB ( menghubungkan Arduino UNO dengan PC)
- Buzzer Module
- LED

Adapun proses pembuatannya adalah melalui tahap-tahap sebagai berikut:

- Pasang kabel hitam pada Radio Receiver AM dengan kabel jumper

ke *ground* analog pin pada arduino uno.

- Pasang kabel merah pada Radio Receiver AM dengan kabel jumper ke pin 5v pada arduino uno.
- Pasang kabel kuning pada Radio Receiver AM dengan kabel jumper ke pin 2 digital pada arduino uno.
- Hubungkan 2 kabel pada buzzer module ke pin *ground* analog pin dan pin 9 digital pada arduino uno.
- Pasang 1 buah LED dengan posisi kaki LED positif ke *ground* digital pin dan kaki LED negatif ke 13 digital pin pada arduino uno.

### 3.4 Pengkodean

Penulis melakukan pengkodean dengan menggunakan Bahasa pemrograman C yang digunakan untuk menampilkan data perekam frekuensi listrik jantung dalam bentuk *stream* pada *serial port*, sedangkan *software processing* dengan bahasa pemrograman Java untuk mengubah data serial tersebut menjadi grafik dari frekuensi listrik jantung yang nantinya dapat dihitung tingginya frekuensi dari listrik jantung tersebut dalam satuan bpm (*beats per minute*). Setelah melewati proses implementasi rangkaian, langkah selanjutnya adalah melakukan *coding* sebagai berikut:

Buat *sketch* pada program Arduino dengan *source code*:

<pre> unsigned int speakerPin = 9; unsigned char LEDpin = 13; unsigned char counter=0; unsigned int heart_rate=0; unsigned long temp[21]; unsigned long sub=0; volatile unsigned char state = LOW; bool data_effect=true; const int max_heartpluse_duty=5000;//you can change it follow your system's request.2000 meams 2 seconds. System return error if the duty overtrip 5 second.  void setup() {   Serial.begin(9600);   //Serial.println("Please ready your chest belt.");   //delay(1000); </pre>	<pre> //Serial.println(temp[counter]); switch(counter) {   case(0):     sub=temp[counter]-temp[20];     Serial.println(sub);     break;    //Serial.println(counter,DEC);   default:     sub=temp[counter]-temp[counter-1];     Serial.println(sub);     beupon();     delay(10);     beepoff();     break; } </pre>
---	--

<pre> array_init(); Serial.println("Heart rate test begin."); attachInterrupt(0, interrupt, RISING);//set interrupt 0,digital port 2 }  void loop() { }  void beepon(){ analogWrite(speakerPin, 150); digitalWrite(LEDpin, HIGH); delay(10000); } void beepoff(){ analogWrite(speakerPin, 0); digitalWrite(LEDpin, LOW); }  void sum();//calculate the heart rate { if(data_effect) { heart_rate=1200000/(temp[20]- temp[0]);/60*20*1000/20_total_time Serial.print("Your heart rate is: "); Serial.print(heart_rate); Serial.println(" bpm"); } data_effect=1;//sign bit } void interrupt() { temp[counter]=millis(); state = !state; </pre>	<pre> if(sub&gt;max_heartpluse_duty)//set 5 seconds as max heart pluse duty { data_effect=0;//sign bit counter=0; //Serial.println("Heart rate measure error,test will restart! or your dead" ); beepon(); array_init(); } if (counter==20&amp;&amp;data_effect) { counter=0; sum(); } else if(counter!=20&amp;&amp;data_effect) counter++; else { counter=0; data_effect=1; } }  void array_init() { for(unsigned char i=0;i!=20;++i) { temp[i]=0; } temp[20]=millis(); } </pre>
---	---

Buat *sketch Java* pada *program Processing* dengan *Source Code*:

<pre> import processing.serial.*;  // Step 2: Declare PFont variable PFont f; Serial myPort; // The serial port float[] vals; PrintWriter output; String heartrate = "Calculating your heart rate";  void setup() { size(400, 300); f = createFont("Arial", 18, true); frameRate(5); // List all the available serial ports println(Serial.list()); // Open whatever port is the one you're using. myPort = new Serial(this, Serial.list()[12], 9600); // don't generate a serialEvent() unless you get a newline character: myPort.bufferUntil('\n'); // An array of initiate values vals = new float[width]; for (int i = 0; i &lt; vals.length; i++) { </pre>	<pre> if (inputString != null) { // trim off any whitespace: String inString = trim(inputString);  // Slide everything down in the array for (int i = 0; i &lt; vals.length-1; i++) { vals[i] = vals[i+1]; } // get the ASCII string: String inputString = myPort.readStringUntil('\n');  output.println(inString); output.flush(); // convert to an int and map to the screen height: int inByte = int(inString); // Add a new value vals[vals.length-1] = 250 - (inByte/5); println(inByte); textFont(f, 18); // Step 4: Specify font to be used </pre>
--	---

<pre> vals[i] = 250; } int d = day(); // Values from 1 - 31 int m = month(); // Values from 1 - 12 int y = year(); // 2003, 2004, 2005, etc. int se = second(); // Values from 0 - 59 int mi = minute(); // Values from 0 - 59 int h = hour(); // Values from 0 - 23 String s = String.valueOf(d) + String.valueOf(m) + String.valueOf(y) + String.valueOf(h) + String.valueOf(mi) + String.valueOf(se); output = createWriter("data" + s + ".txt"); } void draw() { background(0); // Draw lines connecting all points for (int i = 0; i &lt; vals.length-1; i++) { stroke(255, 123, 0); strokeWeight(1.5); line(i, vals[i], i+1, vals[i+1]); } </pre>	<pre> fill(255); // Step 5: Specify font color // Step 6: Display Text text (inByte, 5, 290); } else { vals[vals.length-1] = 250; } if (inputString != null) { if (inputString.charAt(0) == 'Y') { heartrate = inputString; } } }  textFont(f, 18); // Step 4: Specify font to be used fill(255); // Step 5: Specify font color // Step 6: Display Text text (heartrate, 180, 290); } </pre>
---	--



### 3.5 Pengujian Alat

- Dekatkan *Radio Receiver AM* dengan *Grove - Chest Strap Heart Rate Sensor* dengan jarak maksimum 30cm.
- *Save* dan *Upload source code* pada *sketch* program arduino. Lalu klik pojok kanan atas pada Arduino untuk membuka *Serial Monitor*. Jika berhasil maka akan tampil seperti gambar berikut:

```
11339
382
20
12511
1172
Heart_rate_is: 104
```

Gambar 3. Hasil Pengujian

- Ubah *serial port list* yang digunakan pada *source code* sesuai dengan *port* dan *baut-rate* yang digunakan oleh Arduino, pada percobaan kali ini digunakan *serial list* nomor 12 pada *COM port* 27 dan *baut-rate* 9600 (`Serial.list()[12], 9600`).

### 3.6 Pengujian Program

#### 1. Sketch Program Arduino 1

Source Code	Penjelasan
<pre>unsigned char counter=0; unsigned int heart_rate=0; unsigned long temp[21]; unsigned long sub=0; volatile unsigned char state = LOW; bool data_effect=true; const int max_heartpluse_duty=2000;</pre>	<p>Menginisialisasi variabel <code>counter</code>, <code>heart_rate</code>, <code>temp</code>, <code>sub</code>, <code>state</code>, <code>data_effect</code> dan <code>max_heartpluse_duty</code>. <code>HeartPulse 2000</code> berarti 2 detik, jika terjadi overtrip selama 2 detik, sistem menampilkan error.</p>
<pre>void setup() {   pinMode(pin, OUTPUT);   Serial.begin(9600);   Serial.println("Please ready your chest belt.");   delay(1000);//   array_init();   Serial.println("Heart rate test begin.");   attachInterrupt(0, interrupt, RISING);//set interrupt 0,digital port 2 }</pre>	<p>Mempersiapkan program untuk menghitung detak jantung.</p>
<pre>void loop() { } void sum()//calculate the heart rate {   if(data_effect)   {     heart_rate=1200000/(temp[20]-temp[0]);//60*20*1000/20_total_time     Serial.print("Heart_rate_is:\t");</pre>	<p>Menghitung jumlah detak jantung dari <i>array temp</i>.</p>

- *Save* dan *Upload source code* tersebut. Jangan jalankan *serial monitor* yang sama dengan *COM port* arduino karena akan terjadi bentrok. Jika berhasil maka akan tampil seperti gambar berikut:



Gambar 4. Tampilan *Heart Rate Monitor*

<pre> Serial.println(heart_rate); } data_effect=1;//sign bit } </pre>	
<pre> void interrupt() { temp[counter]=millis(); state = !state; Serial.println(counter,DEC); Serial.println(temp[counter]); switch(counter) { case(0): sub=temp[counter]-temp[20]; Serial.println(sub); break; default: sub=temp[counter]-temp[counter-1]; Serial.println(sub); break; } if(sub&gt;max_heartpluse_duty)//set 2 seconds as max heart pluse duty { data_effect=0;//sign bit counter=0; Serial.println("Heart rate measure error,test will restart!" ); array_init(); } if (counter==20&amp;&amp;data_effect) { counter=0; sum(); } else if(counter!=20&amp;&amp;data_effect) counter++; else { counter=0; data_effect=1; } } } </pre>	<p>Melakukan pengecekan secara <i>looping</i> terus menerus terhadap terjadinya interupsi atau jika terjadi <i>overtrip</i>.</p>
<pre> void array_init() { for(unsigned char i=0;i!=20;++i) { temp[i]=0; } temp[20]=millis(); } </pre>	<p>Menginisialisasi nilai array temp agar bernilai 0.</p>

## 2. Sketch pada Program Processing

Source Code	Penjelasan
<pre> import processing.serial.*;  PFont f; Serial myPort; float[] vals; PrintWriter output; String heartrate = "Calculating your heart rate"; </pre>	<p>Mengimpor <i>library</i> untuk melakukan hubungan data melalui serial dan menginisialisasi variabel font, serial, output untuk menulis data ke dalam teks dan string awal.</p>
<pre> void setup() { size(400, 300); f = createFont("Arial", 18, true); } </pre>	<p>Prosedur setup ini dijalankan satu kali oleh program.</p>

<pre> frameRate(5); // List all the available serial ports println(Serial.list()); // Open whatever port is the one you're using. myPort = new Serial(this, Serial.list()[12], 9600); // don't generate a serialEvent() unless you get a newline character: myPort.bufferUntil('\n'); // An array of initiate values vals = new float[width]; for (int i = 0; i &lt; vals.length; i++) {   vals[i] = 250; } int d = day(); // Values from 1 - 31 int m = month(); // Values from 1 - 12 int y = year(); // 2003, 2004, 2005, etc. int se = second(); // Values from 0 - 59 int mi = minute(); // Values from 0 - 59 int h = hour(); // Values from 0 - 23 String s = String.valueOf(d) + String.valueOf(m) + String.valueOf(y) + String.valueOf(h) + String.valueOf(mi) + String.valueOf(se); output = createWriter("data" + s + ".txt"); } </pre>	<ul style="list-style-type: none"> <li>• Program akan membuat jendela tampilan dengan lebar 400px dan tinggi 300px, kemudian menggunakan font Arial berukuran 18 dengan frame Rate tampilan sebanyak 5 fps.</li> <li>• Lalu program akan menampilkan daftar <i>serial port</i> yang terhubung dengan computer, ubah data <code>Serial.list()[i]</code> sesuai dengan list <i>serial port</i> arduino, kali ini menggunakan 12 karena arduino terhubung pada COM port 27 dan berada pada list 12.</li> <li>• <code>myPort.buffer</code> untuk membaca data yang didapatkan dari <i>serial port</i> diatas.</li> <li>• Kemudian inialisasi waktu sistem untuk membuat file teks dengan format "data" + waktu sistem untuk menyimpan seluruh data yang diterima.</li> </ul>
<pre> void draw() {   background(0);   // Draw lines connecting all points   for (int i = 0; i &lt; vals.length-1; i++) {     stroke(255, 123, 0);     strokeWeight(1.5);     line(i, vals[i], i+1, vals[i+1]);   }   // Slide everything down in the array   for (int i = 0; i &lt; vals.length-1; i++) {     vals[i] = vals[i+1];   }   // get the ASCII string:   String inputString = myPort.readStringUntil('\n');   if (inputString != null) {     // trim off any whitespace:     String inString = trim(inputString);     output.println(inString);     output.flush();     // convert to an int and map to the screen height:     int inByte = int(inString);     // Add a new value     vals[vals.length-1] = 250 - (inByte/5);     println(inByte);     textFont(f, 18); // Step 4: Specify font to be used     fill(255); // Step 5: Specify font color     // Step 6: Display Text     text (inByte, 5, 290);   }   else {     vals[vals.length-1] = 250;   }   if (inputString != null) {     if (inputString.charAt(0) == 'Y') {       heartrate = inputString;     }   } }  textFont(f, 18); // Step 4: Specify font to be used </pre>	<p>Prosedur <i>draw</i> ini dijalankan berulang kali oleh program.</p> <ul style="list-style-type: none"> <li>• Program akan membuat garis untuk membuat grafik dari setiap titik point pada kurva grafik.</li> <li>• Program membaca data pada <i>serial port</i> kemudian menghapus <i>whitespace</i> (spasi) dan konversi ke dalam integer. Data inilah yang digunakan sebagai point (titik pada grafik), data ini juga ditampilkan pada program secara realtime. Data ini merupakan data kuatnya denyut jantung.</li> <li>• <code>output.println(inString);</code> dan <code>output.flush();</code> merupakan fungsi untuk menulis data pada teks file pada prosedur setup.</li> <li>• Jika data pada <i>serial port</i> berisi data rata-rata denyut jantung maka denyut jantung tersebut akan ditampilkan pada bagian kanan bawah program.</li> </ul>

```

fill(255); // Step 5: Specify font color
// Step 6: Display Text
text (heartrate, 180, 290);
}

```

## V. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Dari pembahasan yang sudah diuraikan, dapat disimpulkan bahwa perancangan dan pembuatan alat *portable* perekam frekuensi jantung dilakukan dengan menghubungkan Arduino Uno *Grove Chest Strap Heart Rate Sensor*. Selanjutnya membuat program yang berfungsi melakukan konversi data analog (dari sensor suhu) menjadi digital dengan kekuatan denyut jantung dan frekuensi jantung dengan satuan BPM (*beats per minute*) yang ditampilkan dengan grafik dan angka, dapat pula ditampilkan menggunakan serial monitor pada komputer. Indikator frekuensi jantung ditunjukkan dengan Suara pada *buzzer* yang akan berbunyi mengikuti mengikuti tempo frekuensi jantung yang terdeteksi, apabila frekuensi berhenti sekitar 5 detik, maka *buzzer* akan menghasilkan bunyi yang panjang, indikator ini juga bisa ditunjukkan dengan lampu LED yang akan menyala sesuai tempo frekuensi jantung, Hasil rekam denyut jantung akan tersimpan datanya secara otomatis dengan formati teks ekstensi .txt dengan format nama file "data" + waktu percobaan.

### 5.2 Saran

- Pembuatan alat sejenis ini dapat dibuat lebih akurat dengan jarak jangkauan penerimaan data frekuensi jantung yang lebih jauh.
- *Source code* aplikasi arduino yang digunakan belum memiliki fungsi yang lengkap, sehingga

baru bisa menghitung frekuensi denyut jantung saja, mungkin nantinya bisa dikembangkan untuk menghitung selain frekuensi jantung sehingga bisa lebih bermanfaat.

### DAFTAR PUSTAKA

1. Guyton, Hall. *Medical physiologi*, Eleventh Edition, 2006.
2. Bräunl, Thomas. *Embedded Robotics - Mobile Robot Design and Applications with Embedded Systems, Second Edition*. Springer, Jerman, Juni 2006. [3]
3. Byte Craft Limited. *First Steps with Embedded Systems*. Byte Craft Limited, Kanada, November 2012. [1]
4. Wilmshurst, Tim. *Designing Embedded Systems with PIC Microcontrollers - Principles and Applications*. Elsevier Ltd, Inggris, 2007. [2]
5. Arduino. 2010. *Beginning Arduino*. [Online] Tersedia: [www.arduino.cc](http://www.arduino.cc). [12 Desember 2011].
6. Hariyanto, Bambang. 2004. *Sistem Manajemen Basis Data: Pemodelan, Perancangan, dan Terapannya*. Bandung: Informatika.
7. Houde, Stephanie & Hill, Charles. 2004. *What do Prototypes Prototype?.* USA: Apple Computer, Inc.
8. Jogianto, HM. 2005. *Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi.
9. Kendall, Kenneth E. dan Kendall, Julie E. 2003. *Analisis dan Perancangan Sistem*. Jakarta: Pearson Education Asia Pte. Ltd dan PT. Prenhallindo.