

BAB IV

HASIL DAN PEMBAHASAN

Pada bagian bab ini akan menjelaskan tentang analisis hasil pengujian alat yang dimaksudkan untuk mengetahui sistem kerja alat secara keseluruhan perangkat keras dan perangkat lunak. Pengujian dilakukan kepada masing-masing blik sistem rangkaian agar sistem kerja masing-masing bagian dapat diketahui secara rinci tentang data dan kebutuhannya. Pengamatan dilakukan dengan membandingkan alat penelitian dengan alat kedokteran yang ada ke berbagai responden. Dari perolehan data yang didapat dari masing-masing pengujian akan dilakukan analisis data dan melakukan perbandingan data dengan teori-teori terkait untuk memperoleh hasil maksimal dari rangkaian.

4.1 Pengujian Fungsional

Pengujian fungsional bagian demi bagian dan sistem keseluruhan yang terdiri dari pengujian rangkaian catu daya, rangkaian *pulse sensor*, rangkaian *bluetooth*, rangkaian sensor inframerah MLX90614

4.1.1 Pengujian Rangkaian Catu Daya

Pengujian rangkaian catu daya dilakukan untuk memastikan sumberdaya yang masuk ke rangkaian dapat digunakan alat untuk beroperasi karena catu daya memiliki peranan yang penting dalam kelangsungan kerja alat (*hardware*). Catu daya yang diperlukan rangkaian adalah tegangan DC +5V yang dicatu dari baterai 9V yang kemudian dihubungkan dengan tegangan regulator 7805 yang akan

menghasilkan *output* tegangan +5V. Adapun data pengujian tegangan catu daya seperti diperlihatkan pada tabel

Tabel 4.1 Pengujian Catu Daya

No	Bagian	Output Tegangan (V)
1	Baterai	9
2	Input Regulator	9
3	Output regulator	5
4	Ground	0

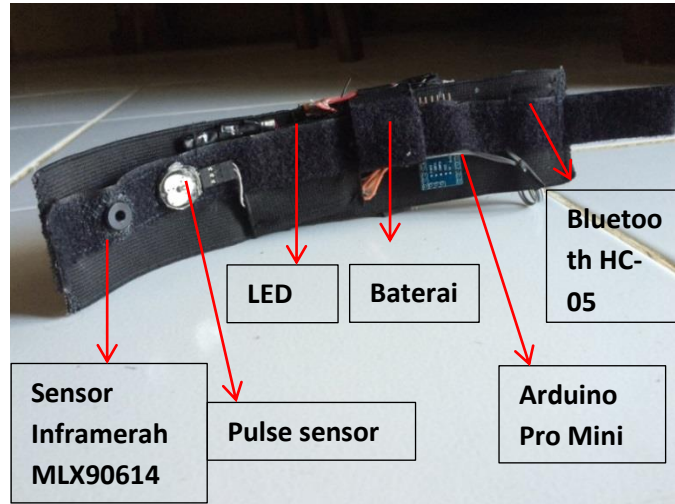
Untuk mengetahui apakah rangkaian catu daya telah berfungsi atau belum digunakanlah lampu LED sebagai indikator *power* yang terdapat pada alat penelitian, ketika saklar dinyalakan maka LED indikator *power* menyala maka tegangan sudah mengalir pada rangkaian.

4.1.2 Pengujian Rangkaian Mikrokontroller

Mikrokontroller merupakan unit kontrol utama pada perancangan sistem ini, mikrokontroller yang digunakan dalam perancangan sistem ini menggunakan arduino Pro Mini karena ukurannya praktis untuk diaplikasikan di pergelangan tangan. Rangkaian arduino ini digunakan untuk mengotrol rangkaian *bluetooth*, rangkaian *pulse sensor*, dan rangkaian sensor inframerah MLX90614.

Pengujian yang dilakukan dengan menghubungkan arduino ke rangkaian *pulse sensor* untuk mendapatkan detak jantung pengguna, rangkaian sensor inframerah MLX90614 untuk mendapatkan suhu pengguna, sedangkan rangkaian

bluetooth digunakan untuk menghubungkan perangkat keras(*hardware*) dengan perangkat android untuk melihat hasil pengukuran detak jantung dan suhu tubuh pengguna.



Gambar 4.1 Bagian-bagian alat penelitian.

4.1.3 Pengujian Rangkaian Bluetooth HC-06

Pengujian pada rangkaian bluetooth sebagai media komunikasi antara perangkat keras dan perangkat lunak. Dengan menggunakan pin TX bluetooth HC-06 yang terhubung dengan pin RX arduino dan pin RX bluetooth HC-06 yang terhubung dengan pin TX arduino. Input tegangan dari rangkaian bluetooth HC-06 ini terhubung dengan VCC arduino pro mini

Pengujian pada bluetooth HC-06 dimaksudkan untuk mengetahui seberapa jauh Bluetooth HC-06 dapat melakukan pemantauan sehingga dapat diketahui jarak maksimal yang dapat dilakukan Bluetooth dalam melakukan pemantauan. Berikut pengujian modul Bluetooth HC-06 yang dapat dilihat pada Tabel

Tabel 4.2 Pengujian modul Bluetooth HC-06

Jarak (meter)	Status	Keterangan
5	Terhubung	Koneksi lancar
10	Terhubung	Koneksi lancar
15	Terhubung	Koneksi lancar
20	Terhubung	Koneksi lancar
25	Terhubung	Koneksi lancar
30	Terhubung	Koneksi putus-putus
>35	Putus	Tidak ada koneksi

Pada saat pengujian modul bluetooth HC-06 dengan jarak lebih dari 35 m koneksi bluetooth HC-06 dengan perangkat android tidak dapat terhubung karena adanya keterbatasan kemampuan komunikasi modul *bluetooth* HC-06.

4.1.4 Pengujian Rangkaian Pulse Sensor

Pada pengujian rangkaian *pulse sensor* ini dilakukan dengan dua kali pengujian, pengujian pertama dilakukan untuk melihat sensor *pulse sensor* dapat bekerja dengan yang diharapkan dan yang kedua dengan membandingkan alat perancangan dengan alat kedokteran tensimeter dengan merk “omron”. Adapun pengujian rangkaian *pulse sensor* sebagai berikut

a. Pengujian sensor tanpa alat pembanding

Pengujian sensor tanpa alat pembanding disini dilakukan untuk melihat apakah sensor dapat bekerja atau tidak dengan mendeteksi dan mengukur

detak jantung ketika perangkat keras dipakai, hasil pengujian rangkaian *pulse sensor* sebagai berikut

Tabel 4.3 Pengujian sensor detak jantung

No	Detik ke-	Koresponden 1	Koresponden 2	Koresponden 3	Koresponden 4
1	0	0	0	0	0
2	2	70	101	96	82
3	4	71	86	75	85
4	6	72	79	69	67
5	8	71	75	65	67
6	10	72	72	63	72
7	12	71	74	66	73
8	14	70	71	73	76
9	16	69	66	99	76
10	18	80	63	114	81
11	20	81	71	108	80
12	22	77	75	112	79
13	24	73	79	91	77
14	26	71	85	80	76
15	28	75	87	77	74
16	30	70	92	73	75
17	32	71	86	79	77

Tabel 4.3 Pengujian sensor detak jantung (lanjutan)

No	Detik ke-	Koresponden 1	Koresponden 2	Koresponden 3	Koresponden 4
18	34	70	72	95	78
19	36	73	71	81	69
20	38	72	70	91	67
21	40	74	76	90	62
22	42	72	97	87	78
23	44	71	92	84	78
24	46	70	93	65	78
25	48	77	94	58	81
26	50	73	86	81	78
27	52	70	78	81	82
28	54	75	65	87	65
29	56	81	62	66	71
30	58	89	81	78	80
31	60	90	74	74	86
Rata-rata		74	79	82	73

Pada pengujian dilakukan dengan cara menghitung detak jantung koresponden memakai alat penelitian dalam waktu 60 detik, pengujian dimaksudkan untuk menguji kestabilan *pulse sensor*. Uji coba ini dilakukan dengan cara mengambil nilai rata-rata hasil pengukuran detak jantung selama 60

detik. Dari hasil pengujian uji coba, pembacaan bpm terhitung stabil antara 60-100 bpm, sesuai dengan denyut jantung normal.

b. Pengujian sensor dengan alat pembanding

Pengujian sensor dengan alat pembanding dilakukan dengan cara mengambil data dari empat orang koresponden dengan variasi umur dan tidak dalam aktivitas yang normal Pengambilan data pada tiap responden dilakukan waktu 60 detik. Pengujian ini dimaksudkan untuk menguji tingkat keberhasilan alat ketika diujikan pada banyak responden. Adapun data yang diperoleh dengan cara membandingkan alat penelitian dengan alat instrumen pembanding kedokteran tensimeter merk “omron” sebagai berikut

Tabel 4.4 Perbandingan pengukuran detak jantung per menit

Responden	Jenis Kelamin	Umur (tahun)	Hasil Pengukuran		Perbandingan	
			Instrumen pembanding (bpm)	Alat Penelitian (bpm)	Akurasi (%)	Ralat (%)
1	Perempuan	52	72	74	98,77	2
2	Perempuan	21	72	73	98.70	1,4
3	Laki-laki	23	90	82	91.10	8.9
4	Laki-laki	62	84	79	94.10	5.95
Rata-rata					95.67	4.65

Data hasil yang didapatkan dari Tabel 4.4 menunjukkan data hasil perbandingan pengukuran antara alat penelitian dengan instrumen pembandingan dan disertai dengan nilai akurasi serta nilai ralat. Nilai akurasi dan nilai ralat didapatkan menggunakan rumus sebagai berikut

$$Akurasi (\%) = 1 - \left| \frac{Nilai alat pembandingan - Nilai alat penelitian}{Nilai alat pembandingan} \right| \times 100 \quad (4.1)$$

$$Nilai Ralat = \left| \frac{Nilai alat pembandingan - Nilai alat penelitian}{Nilai alat pembandingan} \right| \times 100 \quad (4.2)$$

Berdasarkan data hasil yang diperoleh pada Tabel 4.4 dapat dilihat bahwa alat penelitian masih terdapat nilai selisih dengan instrumen pembandingan sehingga tingkat akurasi masih kurang dalam pengukuran detak jantung. Munculnya nilai ralat ini disebabkan oleh beberapa faktor diantaranya kondisi tubuh responden, posisi pergelangan tangan yang selalu bergerak sehingga menimbulkan noise dalam pembacaan bpm. Namun dari data hasil pada Tabel 4.4 membuktikan bahwa denyut jantung orang dewasa yaitu 60 – 100 detak per menit.



Gambar 4.2 Tensimeter

Pada gambar 4.2 menunjukkan hasil pengujian dengan menggunakan instrumen pembanding tensimeter pada koresponden pertama, terlihat bahwa hasil pengukuran detak jantung per menit menunjukkan pada angka 72 per menit sedangkan alat hasil alat penelitian menunjukkan hasil pengukuran 74 per menit sehingga terdapat selisih / ralat yang tidak terlalu jauh perbedaannya mendekati presisi. Adapun faktor yang mempengaruhi pembacaan sensor yaitu cahaya dari lingkungan, aktivitas pasien, dan posisi sensor.

4.1.5 Pengujian Rangkaian Sensor Inframerah MLX90614

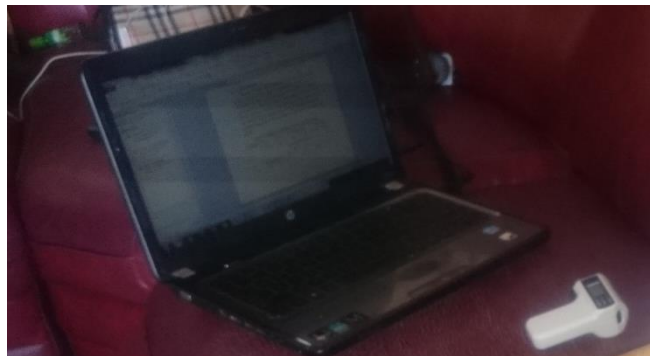
Pengujian sensor inframerah MLX90614 dimaksudkan untuk mengetahui apakah sensor ini dapat digunakan atau tidak. Pengujian dan pengambilan data dilakukan dengan membandingkan dengan instrumen alat pembanding dengan alat penelitian yang dipakai oleh koresponden. Adapun data pengukuran suhu yang diperoleh dari beberapa koresponden dibawah ini

Tabel 4.5 Perbandingan pengukuran suhu tubuh

Responden	Jenis Kelamin	Umur (tahun)	Hasil Pengukuran		Perbandingan	
			Instrumen pembanding (°C)	Alat Penelitian (°C)	Akurasi (%)	Ralat (%)
1	Perempuan	52	37	36.5	98,65	1.35
2	Laki-laki	23	36.2	36.5	99.17	0.83
3	Laki-laki	62	38	37.6	98.95	1.05
Rata-rata					98.45	1.55

Berdasarkan hasil perbandingan pengukuran suhu tubuh terhadap instrumen pembanding kinerja alat penelitian dalam pengukuran suhu tubuh cukup baik. Terdapat nilai error disebabkan oleh beberapa faktor diantaranya keadaan tubuh responden.

Secara keseluruhan alat sudah dapat bekerja dengan cukup baik sesuai dengan harapan. Terdapat sedikit beberapa hasil pengukuran karena adanya beberapa faktor yang mempengaruhi nilai pembacaan sensor. Adapun faktor-faktor yang mempengaruhi pembacaan sensor diantaranya aktivitas pasien dan posisi sensor.



Gambar 4.3 Instrumen pembanding termometer inframerah

4.2 Analisa Data

Perhitungan Detak Jantung

Berdasarkan persamaan (2.4)

$$IBI = \frac{60}{BPM}$$

Persamaan (2.4) merupakan suatu rumus yang akan dipakai untuk mendapatkan detak jantung. Dimana IBI adalah selang waktu pada denyut jantung

dalam satuan mili detik dengan waktu momen sesaat dari jantung berdetak atau *interbeat interval* dan jumlah detak jantung per menit didefinisikan HR.

Contoh hasil pembacaan detak jantung yang diambil adalah rata-rata pembacaan detak jantung koresponden pertama dalam waktu 60 detik pada tabel 4.3 sebagai beriku

$$IBI = \frac{60}{BPM}$$

$$IBI = \frac{60}{74}$$

$$IBI = 0.810$$

Jadi waktu yang diperlukan dalam melakukan pengukuran detak jantung dalam waktu 0.810 detik

4.3 Pembahasan

Pada pembahasan ini akan menjelaskan keseluruhan tentang *hardware* dan *software* secara keseluruhan yang ada pada alat penelitian ini

4.3.1 Pengujian *Hardware* sistem secara keseluruhan

Pada penelitian ini didapatkan suatu alat yang dapat digunakan untuk memonitoring detak jantung dan suhu tubuh berbasis interaksi mikrokontroler Arduino Pro Mini dengan *interface* ponsel android dan *web* melalui komunikasi Bluetooth HC-06. Alat ini diciptakan untuk menampilkan detak jantung per menit, suhu tubuh dan kondisi tubuh pada manusia yang ditampilkan dalam *interface*

android dan *web*. Alat ini terdiri dari beberapa komponen pendukung yaitu sensor MLX90614, *pulse sensor*, Arduino Pro Mini, Android Studio, Bluetooth HC-06, *hosting*, dan *domain*.

Pulse sensor dipakai untuk mengukur dan mendeteksi detak jantung per menit dan sensor inframerah MLX90614 digunakan sebagai pengukur suhu tubuh. Arduino Pro Mini berperan sebagai mikrokontroler yang digunakan untuk mengkonversi besaran listrik dari sensor menjadi besaran digital sehingga dapat diolah dengan rumus. Pemrograman alat penelitian ini dikerjakan pada program Android Studio untuk membuat aplikasi android yang dapat dipasang pada perangkat android. Ponsel android sebagai *interface* yang akan menampilkan hasil pengukuran sensor sesuai dengan *output* dari Arduino dengan komunikasi Bluetooth HC-06. Perhitungan detak jantung dilakukan dengan menggunakan rumus *Interbear Interval* (IBI). IBI merupakan selang waktu pada detak jantung dalam mili detik dengan waktu momen sesaat dari jantung berdetak. Rumus IBI ini dimasukkan ke dalam pemrograman Arduino. Pemrograman Arduino mengolah gelombang pulsa yang dikirim oleh *pulse sensor*. Gelombang pulsa yang dikirim oleh *pulse sensor* akan diolah di dalam fungsi Timer2 yang tersimpan dalam register dengan syarat-syarat yang telah ditentukan, kemudian setelah mendapatkan gelombang pulsa yang sesuai, barulah dicari nilai BPM (*Beats Per Minute*) dengan cara 60000 dibagi dengan jumlah *runningTotal*. Nilai BPM ini akan ditampung ke dalam bentuk Array dan ditampilkan pada Serial Monitor Arduino.

Sementara dalam pengukuran suhu tubuh memakai sensor MLX90614. Diperlukan komunikasi I2C untuk dapat menerima dan mengirim data. Langkah

pertama saat program arduino dijalankan yaitu menginisialisasi alamat perangkat sensor yang telah ditentukan oleh pabrik, apabila proses komunikasi telah selesai maka dilakukan tahap *packet error checking* disini arduino akan melakukan perintah *write* pada I2C. Proses pembacaan data pada I2C terdiri dari data *low* sebesar 8 bit LSB dan *high* 8 bit MSB serta terdiri dari PEC yang akan memberi keterangan bahwa pengiriman data yang terdiri dari LSB dan MSB telah selesai. Selanjutnya terjadi proses masking untuk mengosongkan nilai *data_high* dan menggeser 8 bit ke kiri dan ditambahkan nilai *data_low* karena menurut *datasheet* nilai *data_high* merupakan nilai *error*. Proses yang terakhir adalah mengalikan data *output* dengan nilai resolusi sehingga didapatkan nilai suhu dalam satuan Kelvin. Untuk memperoleh nilai suhu ke dalam satuan Celcius didapatkan dengan mengurangi 272.15.

Selanjutnya dilakukan pengujian alat untuk memastikan alat dalam kondisi baik dan dapat dipakai. Pengujian alat dilakukan untuk pengambilan data keluran dari sensor. Ketika melakukan pengujian pengukuran suhu tubuh, pembacaan sensor MLX90614 tidak presisi dengan instrumen pembanding, oleh karena itu dicari nilai *error* dan nilai tersebut ditambahkan ke dalam program arduino bagian pengolahan suhu. Nilai error yang didapatkan sebesar 6.45%, sehingga diperoleh nilai yang mendekati presisi dengan instrumen pembanding dan dilakukan pengambilan data ulang. Selanjutnya melakukan dilakukan pengukuran detak jantung per menit alat penelitian dengan instrumen pembanding. Hasil yang didapatkan alat penelitian cukup baik dalam pengukuran detak jantung, namun *pulse sensor* akan mengalami *noise* setelah cukup lama digunakan. Perolehan hasil pengukuran ini akan ditampilkan di android yang telah terinstall dengan

aplikasi android yang telah dirancang. Arduino Pro Mini mempunyai port RX-TX yang berfungsi sebagai komunikasi serial antara Arduino dengan Bluetooth sehingga data pengukuran dapat dilihat secara langsung di android.

4.3.2 Pembahasan *Software* sistem secara keseluruhan

Pada pembahasan *software* ini akan membahas tentang semua *software* yang digunakan untuk membuat alat penelitian ini. Adapun pembahasan *software* ini diantaranya mencakup pembuatan program alat monitoring detak jantung dan suhu tubuh ini dirancang dengan berbasis mikrokontroler arduino Pro Mini dengan menggunakan bahasa pemrograman bahasa C, kemudian untuk membuat *interface* di ponsel android menggunakan *software* Android Studio dengan bahasa pemrograman java, dan untuk membuat *interface* pada halaman *website* dengan bahasa pemrograman PHP. Tujuan pengujian program keseluruhan adalah untuk mengetahui apakah program yang telah dibuat dapat ditampilkan sesuai dengan yang dikehendaki.

Adapun pembahasan *software* secara rinci sebagai berikut:

4.3.2.1 Pembahasan pemrograman Arduino

Alat monitoring detak jantung dan suhu tubuh ini dirancang dengan berbasis mikrokontroler arduino Pro Mini dengan menggunakan bahasa pemrograman bahasa C. Perangkat keras akan berjalan sesuai fungsinya apabila alat sudah diprogram menggunakan program Arduino yaitu Arduino IDE. Pengujian perangkat lunak ini dilakukan pada program Arduino dengan menggunakan Arduino IDE. Pengujian program ini dilakukan untuk melihat

kesesuaian program dengan diagram alir yang telah dibuat sebelumnya. Pengujian ini akan menjelaskan semua bagian program Arduino secara terpisah sesuai dengan fungsi yang digunakan.

Program Arduino terdiri dari Fungsi Setup, Fungsi interruptSetup, Fungsi sendAndroid, dan Fungsi Loop.

1. Fungsi Setup

Fungsi setup merupakan struktur vital dalam pemrograman arduino yang berperan untuk menginisialisasi *variable*, mode pin dan memulai menggunakan *library* / deklarasi pustaka. *Library* arduino sangat penting dalam pemrograman arduino, *variable* adalah suatu identitas yang dideklarasikan yang memiliki tipe data tertentu.

Struktur penulisan fungsi setup terdiri dari beberapa blok, yaitu:

a. Deklarasi Pustaka

Intruksi yang pertama digunakan untuk mendeklarasikan pustaka berdasar *library* yang dipakai dalam pembuatan alat penelitian ini, dalam contoh ini digunakan *library* sensor inframerah MLX90614 sehingga *library* yang dibutuhkan adalah “#include <Adafruit_MLX90614.h>” sedangkan untuk memuat *library wire* yang digunakan sebagai komunikasi serial I2C “#include <Wire.h>

```
#include <Wire.h> /memuat library wire <komunikasi serial>
```

```
#include <Adafruit_MLX90614.h> //memuat library MLX90614
```

b. Deklarasi input output

Intruksi berikutnya adalah mengkonfigurasi port port pada mikrokontroler, fungsi ini digunakan untuk menentukan apakah suatu port atau pin pada AVR digunakan sebagai masukan atau keluaran.

```
int pulsePin = 0; // menghubungkan input pulse sensor ke pin A0
```

```
int blinkPin = 13; //digunakan pada LED yang berkedip tiap detik
```

blinkPin adalah led otomatis yang akan menyala ketika pin ini dipanggil, sedangkan pin pulsePin adalah pin untuk *pulse sensor* sebagai input yang akan masuk ke pin A0 arduino dengan menggunakan tipe data integer.

c. Deklarasi variabel

Setiap variabel yang akan digunakan pada program harus dideklarasikan di awal program agar variabel yang digunakan tersebut dapat dikenali ketika di-*compile* oleh arduino.

Berikut adalah *listing* penulisan deklarasi variabel yang digunakan

```
volatile int BPM; // variabel untuk nilai beat per minute
```

```
volatile int Signal; // variabel untuk signal
```

```
volatile int IBI = 600; // variabel unuk Inter beat Interval
```

```
volatile boolean Pulse = false; //variabel bernilai benar saat gelombang tinggi, salah saat rendah
```

```
volatile boolean QS = false; // variabel bernilai benar saat arduino menemukan detakan
```



```

volatile int rate[10]; // variabel untuk pembuatan array perhitungan
detak jantung

volatile unsigned long sampleCounter = 0; // var menyimpan data
array perhitungan detak jantung

volatile unsigned long lastBeatTime = 0;

volatile int P = 512;

volatile int T = 512;

volatile int thresh = 512;

volatile int amp = 100;

volatile boolean firstBeat = true;

volatile boolean secondBeat = false;

```

variabel yang dideklarasikan dia atas merupakan variabel untuk membuat program detak jantung dengan *pulse sensor*.

d. Deklarasi fungsi setup

Fungsi setup digunakan untuk membuka serial port, mengatur kecepatan data ke 9600 bps dalam arduino dan menampilkan hasil pengukuran detak jantung dan suhu tubuh

Berikut *listing* penulisan fungsi setup

```

void setup(){

Serial.begin(9600); // membuka serial port, mengatur kecepatan data
ke 9600 bps

mlx.begin(); // untuk menampilkan suhu tubuh

interruptSetup(); // untuk menuju ke fungsi interruptSetup

```

2. Fungsi InterruptSetup

Fungsi InterruptSetup merupakan fungsi khusus untuk mengatur waktu interrupt dalam pembuatan gelombang pulsa detak jantung. Berikut *listing* program fungsi InterruptSetup

```
1. void interruptSetup(){
2.   TCCR2A = 0x02; // set nilai register TCCR2A
3.   TCCR2B = 0x06; // set nilai register TCCR2B
4.   OCR2A = 0x7C;
5.   TIMSK2 = 0x02; // membangkitkan timer 2 interrupt
6.   sei();
7. }
8. ISR(TIMER2_COMPA_vect){
9.   cli();
10.  Signal = analogRead(pulsePin);
11.  sampleCounter += 2;
12.  int N = sampleCounter - lastBeatTime;
13.  if(Signal < thresh && N > (IBI/5)*3){
14.    if (Signal < T){
15.      T = Signal;
16.    }
17.  }
18.  if(Signal > thresh && Signal > P){
19.    P = Signal;
20. }
21. if (N > 250){
22.   if ( (Signal > thresh) && (Pulse == false)
23.     && (N > (IBI/5)*3) ){
24.     Pulse = true;
25.     digitalWrite(blinkPin,HIGH);
26.     IBI = sampleCounter - lastBeatTime;
27.     lastBeatTime = sampleCounter;
28.     if(secondBeat){
29.       secondBeat = false;
30.       for(int i=0; i<=9; i++){
31.         rate[i] = IBI;
32.       }
33.     }
34.     if(firstBeat){
35.       firstBeat = false;
36.       secondBeat = true;
37.       sei();
38.       return;
39.     }
40.     word runningTotal = 0;
41.     for(int i=0; i<=8; i++){
42.       rate[i] = rate[i+1];
43.       runningTotal += rate[i];
44.     }
45.     rate[9] = IBI;
46.     runningTotal += rate[9];
47.     runningTotal /= 10;
48.     BPM = 60000/runningTotal;
49.     QS = true;
50.   }
51. }
52. if (Signal < thresh && Pulse == true){
53.   digitalWrite(blinkPin,LOW);
54.   Pulse = false;
55.   amp = P - T;
56.   thresh = amp/2 + T;
57.   P = thresh;
58.   T = thresh;
59. }
```

```

52.     if (N > 2500){
53.         thresh = 512;
54.         P = 512;
55.         T = 512;
56. lastBeatTime = sampleCounter;
57.         firstBeat = true;
58.         secondBeat = false;}
59.         sei();

```

Gambar 4.4 *listing* program fungsi *interruptSetup*

Pada *listing* program diatas menunjukkan *listing* program fungsi *interruptSetup* yang berperan untuk mengontrol data supaya tidak saling bertabrakan. Program detak jantung ini mengambil data dengan cepat dalam waktu 2ms dan bersifat sementara atau volatile, dan pengaturan waktu interrupt ini disimpan dalam register yang ada di arduino. Kode program baris ke 2-6 adalah kode arduino untuk pengaturan Interrupt. Pokok dari program detak jantung ini yaitu untuk mencari nilai *heart rate* yang berasal dari gelombang pulsa. Gelombang pulsa Timer2 akan dibandingkan dengan gelombang pulsa InterruptSetup. Gelombang pulsa interrupt setup menggunakan IBI dengan nilai 600, sedangkan gelombang pulsa Timer2 menggunakan IBI dengan nilai 360. Gelombang pulsa Timer2 ini diatur dalam fungsi if yang tertulis pada kode program baris 13-37 Gambar 4.4. Kedua variabel ini akan terus dibandingkan untuk mendapatkan output dengan nilai error yang kecil dan membentuk gelombang pulsa yang sesuai dan diatur dalam IBI 600. Output tersebut adalah *runningTotal*. Nilai *runningTotal* ini, akan dimasukkan ke dalam Array. Array dapat menampung 10 bit data. Jadi untuk menentukan detak jantung per menit atau BPM yaitu 60000 dibagi dengan nilai *runningTotal*. Data detak jantung akan muncul dalam waktu 2ms. Saat BPM tertampil di serial monitor, lampu LED pada *blinkPin* akan menyala dan ketika

memasukkan detak jantung selanjutnya, lampu LED akan mati namun nilai BPM tetap terbaca.

3. Fungsi *sendAndroid*

Fungsi *sendAndroid* berfungsi untuk menampilkan data yang akan ditampilkan pada serial monitor dan dikirim oleh bluetooth kemudian ditampilkan pada android. Berikut *listing* program fungsi *sendAndroid*

```
1. void sendAndroid(){
2.   int Batre; // variabel nilai baterai
3.   int Tegangan = (analogRead(A1)/1024.00)*100;
4.   Serial.print('#');
5.   Serial.print(BPM);
6.   Serial.print('+');
7.   Serial.print(mlx.readObjectTempC());
8.   Serial.print('$');
9.   Serial.print(Tegangan);
10.  Serial.print("%");
11.  Serial.print('~');
12.  Serial.println();
13.  delay(1000);
}
```

Gambar 4.5 *listing* program fungsi *sendAndroid*

Pada program baris ke 2-3 merupakan program untuk menampilkan persentase baterai dengan tipe data integer. Kode program ini memakai rumus ADC. Baterai masuk ke dalam pin analog A1 arduino kemudian dibagi dengan nilai ADC saat tegangan baterai masuk utuh maka hasilnya adalah tegangan sisa dari baterai tersebut. Hasil dari perhitungan tersebut dikalikan 100 sehingga didapatkan nilai dalam bentuk persen. Data yang ditampilkan pada serial monitor yaitu detak jantung (BPM), suhu tubuh dan persentase baterai.

Data-data tersebut diberi batas dengan simbol-simbol yang ada pada kode program baris ke 4, 6, 8, dan 10 agar data dapat terbaca dengan baik dan dapat di-*substring* dalam android.

4. Fungsi *loop*

Fungsi *loop* mempunyai peranan penting dalam pemrograman Arduino fungsi ini digunakan untuk mengerjakan program yang dipanggil di dalamnya secara berulang-ulang atau *looping*. Fungsi *sendAndroid* dimasukkan ke dalam fungsi *loop* karena di dalam fungsi *sendAndroid* berisi semua data. Sehingga serial monitor akan menampilkan data pada fungsi *sendAndroid* secara berulang-ulang dengan output data yang *update*. Berikut *listing* program fungsi *loop* pada gambar 4.6

```
1. void loop() {  
2. sendAndroid(); }
```

Gambar 4.6 *listing* program fungsi *loop*

4.3.2.2 Pembahasan pemrograman Android

Pada pembuatan program aplikasi berbasis Android terdiri dari tiga komponen utama dan menggunakan dua macam bahasa pemrograman. Tiga komponen utama yang terdapat pada aplikasi Android ini yaitu pembuatan komponen layout program yang berfungsi sebagai user interface aplikasi, yang dibuat dengan menggunakan bahasa pemrograman xml.

Pembuatan komponen halaman AndroidManifest.xml halaman ini dibuat secara otomatis untuk menampung dan mendaftarkan bagian-bagian penting yang

digunakan dalam aplikasi, seperti library, application, activity, dan class yang akan dijadikan main class (class yang dijalankan lebih dulu).

Pembuatan komponen isi program, yaitu halaman isi program yang menggunakan bahasa pemrograman java dan pembuatan *layout*

1. Pembuatan Halaman Pembuka

Halaman pembuka dalam aplikasi ini menggunakan *splash screen*, *splash screen* ini digunakan sbagai *feedback* pada aplikasi masih pada berada dalam proses *loading*. Dalam perancangan *splash screen*, menggunakan gambar logo dengan animasi waktu satu detik.

a. *Layout* halaman pembuka

Pembuatan *layout* ini akan menampilkan gambar logo aplikasi selama beberapa detik

```
<?xml version="1.0" encoding="utf-8"?>
  <RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
  <ImageView
    android:src="@drawable/splash"//memuat gambar splashscreen
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#339966"/>
  </RelativeLayout>
```

Gambar 4.7 *splashscreen layout*

Pada *splashscreen layout* ini digunakan untuk memuat gambar logo aplikasi dalam waktu tertentu, sumber gambar *splash* yang tersimpan dalam folder *drawable*

b. Kode program halaman pembuka

Pembuatan kode program halaman pembuka ini dimaksudkan untuk mengatur berapa lamanya *activity* tampilan *layout splashscreen* muncul untuk membuka *activity* selanjutnya.

SplashScreen.java, pada *class activity* ini digunakan untuk mempresentasikan *layout splashscreen.xml* sebagai *user interface splashscreen*

```
Public class SplashScreen extends Activity{
    private static final int waktu_splash = 2 * 5000; // 1 detik
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.splashscreen);
    }
}
```

Gambar 4.8 splashscreen.java

Pada variabel *waktu_splash* inilah yang mengatur berapa lamanya tampilan halaman *splashscreen* akan muncul yaitu selama satu detik, selanjutnya untuk memuat halaman *layout XML splashscreen* terdapat *method setContentView*, selama *splashscreen* berlangsung terdapat *delay* atau waktu jeda *splashscreen* menuju ke *activity TampilanUtama.java* terdapat animasi *fade in* dan *fade out*.

```

//method yang berfungsi untuk memberikan efek transisi animasi
fade_in dan fade_out
        overridePendingTransition(R.anim.fade_in,
R.anim.fade_out);
    }
}, 1000);

```

Gambar 4.9 animasi splashscreen

2. Pembuatan Halaman Menu Utama

a. *Layout* halaman menu utama

Halaman menu utama pada aplikasi android ini merupakan halaman yang muncul pertama kali ketika aplikasi ini dijalankan yang merupakan panel menu-menu navigasi utama, menu utama terdapat tiga buah *button* yaitu tombol bluetooth, tombol instruction, dan tombol about. Pembuatan tampilan antarmuka pada halaman menu utama di implementasikan dalam bentuk XML. Setiap elemen dalam tampilan antarmuka perlu ditambahkan atribut pengenalan, sehingga elemen tersebut akan di *generate* dalam atribut XML dan memudahkan untuk digunakan pada kelas yang memerlukan adapun kode xml adalah sebagai berikut:

```

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_weight="1"
    android:background="#cccccc"
    android:weightSum="1">
    <!-- News Feed Button -->
    <Button
        android:id="@+id/btn_bt"
        style="@style/DashboardButton"
        android:drawableTop="@drawable/btn_bt"
        android:text="Bluetooth"
        android:textColor="#339966"

```



```

        android:layout_width="wrap_content"
        android:paddingRight="@dimen/activity_horizontal_margin"
        android:layout_weight="0.3"
    />
    <Button
        android:id="@+id/btn_ins"
        style="@style/DashboardButton"
        android:drawableTop="@drawable/ins_btn"
        android:text="Instruksi"
        android:textColor="#339966"
        android:layout_width="wrap_content"
        android:layout_gravity="center_vertical"
        android:paddingRight="@dimen/activity_horizontal_margin"
        android:layout_weight="0.8"
    />
    <Button
        android:id="@+id/btn_cp"
        style="@style/DashboardButton"
        android:drawableTop="@drawable/cp_btn"
        android:text="About Us"
        android:textColor="#339966"
        android:layout_width="wrap_content"
        android:paddingRight="@dimen/activity_horizontal_margin"
        android:layout_gravity="center_vertical"
    />
</LinearLayout>

```

Gambar 4.10 frangment_layout

Pada fragment_layout ini digunakan untuk medeklarasikan pembuatan identitas pengenalan untuk pemanggilan *button* id, adapun terdapat beberapa atribut pengenalan atau id diantaranya `android:id="@+id/btn_bt`, `android:id="@+id/ins_btn`, dan `android:id="@+id/cp_btn` atribut ini digunakan untuk atribut pengenalan pembuatan *button* atau tombol yang akan dipanggil dalam kelas java yang memanggilnya nanti,

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```

    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:src="@drawable/h3"
        android:layout_gravity="top" />
</LinearLayout>

```

Gambar 4.11 *layout wall*

Pembuatan *layout wall* ini digunakan untuk menampilkan tampilan gambar pada tampilan menu utama. Gambar yang ditampilkan disimpan dalam folder `drawable/h3`.

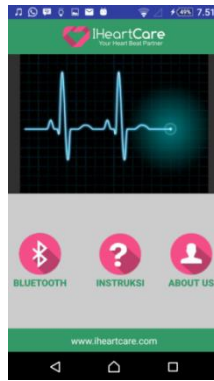
```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    style="@style/FooterBar" >
    <TextView android:text="www.iheartcare.com"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textColor="#ffffff"
        android:gravity="center"
        android:paddingTop="10dip"/>
</LinearLayout>

```

Gambar 4.12 *footer_layout*

Pembuatan *footer_layout* ini digunakan untuk menampilkan tampilan text yang terdapat di bawah pada tampilan menu utama. Adapun tampilan *layout* menu utama dalam *file layout XML dashboard_layout*.



Gambar 4.13 Tampilan *layout* menu utama

b. Kode program halaman menu utama

File class java mempresentasikan *layout* XML yang telah dibuat menjadi sebuah *interface*. Penulisan program java membuat fungsi-fungsi dari atribut XML menjadi sebuah antarmuka yang mempunyai *event* / fungsi ataupun tidak.

```

package com.example.jalu.iheartcare;
import android.os.Bundle;
import android.view.Window;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.widget.Button;
public class TampilanUtama extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        this.requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.dashboard_layout);
        Button btn_cp= (Button) findViewById(R.id.btn_cp);
        btn_cp.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(),
About.class);
                startActivity(i);
            }
        });
    }
}

```

```

    }
    });
    Button btn_bt= (Button) findViewById(R.id.btn_bt);
    btn_bt.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            startActivity(new Intent(TampilanUtama.this,
MenuBluetooth.class));
        }
    });
    Button btn_ins= (Button) findViewById(R.id.btn_ins);
    btn_ins.setOnClickListener(new View.OnClickListener() {

        public void onClick(View v) {
            Intent i = new Intent(getApplicationContext(),
Instruction.class);
            startActivity(i);
        }
    });
}
}

```

Gambar 4.14 TampilanUtama.java

Kode program java untuk membuat *event* atau *action* ketika tombol di tekan, *method* `setOnClickListener` digunakan untuk membuka *activity* lain yang mengarahkan ke *activity class* yang dituju. Pemanggilan atribut pengenalan tombol yang terdapat pada *file layout* ini sebagai identitas

3. Pembuatan Menu Bluetooth

Menu Bluetooth merupakan *activity* yang digunakan untuk menampilkan daftar perangkat bluetooth pada ponsel android yang sedang aktif disekitar alat penelitian.

a. *Layout* menu bluetooth

Komponen penyusun dari *bt_layout* terdiri dari tampilan *TextView* yang digunakan untuk menampilkan teks *header* (Pilih Perangkat Bluetooth), kemudian tampilan *ListView* yang digunakan untuk menampilkan daftar perangkat bluetooth yang sedang aktif, selanjutnya tampilan *TextView* yang digunakan untuk menampilkan teks “menyambungkan”, ketika perangkat bluetooth dipilih

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#cccccc" >
    <RelativeLayout
        android:id="@+id/relativeLayout1"
        style="@style/FooterBar" >
    </RelativeLayout>
    <TextView android:id="@+id/title_paired_devices"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Pilih Perangkat Bluetooth"
        android:paddingTop="10dip"
        android:gravity="center"
        android:visibility="gone"
        android:background="#339966"
        android:textColor="#ffffff"
        android:paddingLeft="5dp"
        android:textSize="@dimen/bawah" />
    <ListView android:id="@+id/paired_devices"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:stackFromBottom="false"
        android:layout_weight="1"
        android:textColor="#000000"
        android:background="#cccccc" />
```

```

<TextView
    android:id="@+id/connecting"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:background="#cccccc"
    android:textColor="#339966"
    android:textSize="@dimen/menyambungkan" />
<TextView
    android:id="@+id/infoText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#339966"
    android:text="Jika Perangkat Bluetooth tidak tersedia silahkan
cek pengaturan untuk pairing bluetooth"
    android:textSize="@dimen/bawah"
    android:textAppearance="?android:attr/textAppearanceLarge" />
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center">
</LinearLayout>
</LinearLayout>

```

Gambar 4.15 bt_layout

b. Kode program menu bluetooth

Kode program menu bluetooth untuk mempresentasikan *activity* tampilan XML *layout bt_layout*, terdapat langkah-langkah yang perlu dilakukan dalam pemrogramannya. Langkah pertama yaitu perlu menambahkan *permission* Bluetooth pada *file* *AndroidManifest.xml*. *Permission* bluetooth ini merupakan hak akses penggunaan layanan bluetooth dalam android untuk dapat berkomunikasi dengan perangkat bluetooth lainnya. Berikut adalah kode program bluetooth *permission* pada *file* *AndroidManifest.xml*

```

<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />

```

Gambar 4.16 Bluetooth Permission di AndroidManifest.xml

Selanjutnya mengecek status bluetooth android pengguna sudah aktif atau dalam keadaan *enable* atau belum, apabila bluetooth belum dinyalakan maka akan ada peringatan untuk mengaktifkan bluetooth

```

private void checkBTState() {
    mBtAdapter=BluetoothAdapter.getDefaultAdapter();
    if(mBtAdapter==null) {
        Toast.makeText(getApplicationContext(), "Device does not support
Bluetooth", Toast.LENGTH_SHORT).show();
    } else {
        if (mBtAdapter.isEnabled()) {
            Log.d(TAG, "...Bluetooth ON...");
        } else {
            //menampilkan dialog untuk menyalakan Bluetooth apabila blm aktif
            Intent enableBtIntent = new
            Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(enableBtIntent, 1);
        }
    }
}

```

Gambar 4.17 listing program cek bluetooth

Pada program di atas terdapat *method checkBTState* yang digunakan untuk mengecek status keadaan bluetooth, apabila bluetooth mati atau *disable* maka terdapat *toast* (kotak dialog) perintah untuk mengaktifkan bluetooth / *enable* yaitu pada *ACTION_REQUEST_ENABLE*. Setelah pengguna telah mengaktifkan bluetooth maka selanjutnya ponsel android akan menampilkan daftar perangkat bluetooth yang ter-*pairing* ke dalam

bentuk *listview* yang sedang aktif di sekitar ponsel android. Berikut *listing* program untuk mendapatkan perangkat bluetooth yang ter-*pairing* dengan ponsel

```
ListView pairedListView = (ListView)
findViewById(R.id.paired_devices);
pairedListView.setAdapter(mPairedDevicesArrayAdapter);
pairedListView.setOnItemClickListener(mDeviceClickListener);
// mendapatkan local bluetooth adapter
mBtAdapter = BluetoothAdapter.getDefaultAdapter();
//ambil paired/bonded device, akan ditampung di arraylist
Set<BluetoothDevice> pairedDevices = mBtAdapter.getBondedDevices();
// menambahkan perangkat yg terhubung ke array
if (pairedDevices.size() > 0) {
    //Loop semua paired devices
    findViewById(R.id.title_paired_devices).setVisibility(View.VISIBLE); //make title viewable
    for (BluetoothDevice device : pairedDevices) {
        //tambahkan ke listview
        mPairedDevicesArrayAdapter.add(device.getName() + "\n" +
device.getAddress());
    }
} else {
    String noDevices =
getResources().getText(R.string.none_paired).toString();
    mPairedDevicesArrayAdapter.add(noDevices);
}
}
```

Gambar 4.18 *listing* program paired device

Pada *listing* program di atas terdapat *method* *getBondedDevice* digunakan untuk mencari nama-nama perangkat bluetooth yang terbaca pada ponsel, nama-nama perangkat bluetooth dapat terbaca karena adanya *method* *setVisibility*, kemudian dari nama-nama perangkat bluetooth yang terbaca disimpan dalam bentuk *layout* tampilan *listview* menggunakan *ArrayAdapter* yang tersimpan dalam *file* XML *bt_layout.xml*. sehingga

pengguna dapat memilih salah satu perangkat bluetooth. Berikut adalah *listing* program saat pengguna akan memilih nama perangkat bluetooth

```
private OnItemClickListener mDeviceClickListener = new
OnItemClickListener() {
    public void onItemClick(AdapterView<?> av, View v, int arg2, long
arg3) {
        textView1.setText("Menyambungkan...");
        String info = ((TextView) v).getText().toString();
        String address = info.substring(info.length() - 17);
        // membuat intent untuk memulai activity bersamaan dengan
mengambil extra MAC Add
        Intent i = new Intent(MenuBluetooth.this, MainActivity.class);
        i.putExtra(EXTRA_DEVICE_ADDRESS, address);
        startActivity(i);
    }
};
```

Gambar 4.19 *listing* program memilih perangkat bluetooth

Pada program Gambar 4.16 menjelaskan apabila pengguna memilih / klik nama perangkat bluetooth yang ada pada *listview* maka akan muncul status “Menyambungkan”, dalam hal ini bluetooth pada ponsel android akan menghubungkan dengan perangkat bluetooth, selanjutnya nama bluetooth yang dipilih akan dikirimkan ke *activity* MainActivity akan membuka *intent* yang ada pada di *activity* MainActivity melalui *putExtra* jika berhasil melakukan koneksi dengan perangkat bluetooth, maka aplikasi akan membuka *activity* selanjutnya.

Bluetooth memakai prinsip master-slave atau *client server* dan dapat bertukar posisi. Aplikasi android ini berperan sebagai *client* untuk menerima data sensor yang dikirim oleh Bluetooth HC-06. Untuk dapat saling berkomunikasi antar bluetooth menggunakan *BluetoothSocket*,

setelah saling terhubung, *client* bluetooth dapat membaca data yang dikirimkan oleh bluetooth lain menggunakan fungsi *method* `socket.getInputStream()`, sedangkan data yang akan dikirimkan oleh bluetooth *client* menggunakan *method* `socket.getOutputStream()`.

Komunikasi antara *bluetooth client* (aplikasi android) dengan *bluetooth server* bluetooth HC-06 menggunakan komunikasi serial untuk dapat saling terhubung. Adapun program sebagai berikut

```
//instance variabel
final int handlerState = 0;
private BluetoothAdapter btAdapter = null;
private BluetoothSocket btSocket = null;
private StringBuilder recDataString = new StringBuilder();
private ConnectedThread mConnectedThread;
private static final UUID BTMODULEUUID = UUID.fromString("00001101-0000-
1000-8000-00805F9B34FB");
private static String address;
```

Gambar 4.20 listing program Serial bluetooth

UUID (Universally Unique Identifier) dalam program serial bluetooth ini digunakan untuk mengidentifikasi secara unik layanan bluetooth aplikasi.

```
private BluetoothSocket createBluetoothSocket(BluetoothDevice device)
throws IOException {
    return device.createRfcommSocketToServiceRecord(BTMODULEUUID);
}
```

Gambar 4.21 createBluetoothSocket

BluetoothSocket ini digunakan untuk mempresentasikan antarmuka untuk *socket* bluetooth (mirip dengan TCP *socket*) untuk saling bertukar data dengan bluetooth lain melalui InputStream dan OutputStream

```
private class ConnectedThread extends Thread {
    private final InputStream mmInStream;
    private final OutputStream mmOutStream;
    public ConnectedThread(BluetoothSocket socket) {
        InputStream tmpIn = null;
        OutputStream tmpOut = null;
        try {
            tmpIn = socket.getInputStream();
            tmpOut = socket.getOutputStream();
        } catch (IOException e) { }
        mmInStream = tmpIn;
        mmOutStream = tmpOut;
    }
    public void run() {
        byte[] buffer = new byte[256];
        int bytes;
        while (true) {
            try {
                bytes = mmInStream.read(buffer);
                String readMessage = new String(buffer, 0, bytes);
                bluetoothIn.obtainMessage(handlerState, bytes, -1,
readMessage).sendToTarget();
            } catch (IOException e) {
                break;
            }
        }
    }
    //write method
    public void write(String input) {
        byte[] msgBuffer = input.getBytes();
        try {
            mmOutStream.write(msgBuffer);
        } catch (IOException e) {
            Toast.makeText(getApplicationContext(), "Connection Failure",
Toast.LENGTH_LONG).show();
            finish();
        }
    }
}
```

```
}  
}
```

Gambar 4.22 Thread untuk transfer data

Ketika kedua perangkat masing-masing memiliki `BluetoothSocket` yang terhubung, disinilah fungsi subclass dari thread untuk menmbagi proses dalam aplikasi pada waktu yang sama untuk saling bertukar data, pertukaran data omo melalui transmisi lewat *socket* melalui `getInputStream()` dan `getOutputStream()`

4. Pembuatan Program Utama

Kode program bluetooth yang akan menerima data dari arduino terletak pada *activity* program utama atau `MainActivity`. Pada *activity* ini akan menampung dan sekaligus menampilkan hasil pembacaan data sensor yang dihasilkan.

a. *Layout* MainActivity

Pembuatan *layout* ini digunakan untuk mempresentasikan hasil pembacaan detak jantung dan suhu ubuh melalui komunikasi antara alat dengan aplikasi android. Bluetooth akan menerima data dari arduino, tampilan pembacaan detak jantung, suhu tubuh, baterai, dan kondisi yang terpisah dalam suatu blok.

```
<RelativeLayout  
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:tools="http://schemas.android.com/tools"  
android:layout_width="match_parent"  
android:layout_height="match_parent"
```

```

android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin" >
<View
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:id="@+id/centerVerticalShim"
    android:layout_centerVertical="true"
    android:visibility="visible" />
<View
    android:layout_width="20dp"
    android:layout_height="match_parent"
    android:id="@+id/centerHorizontalShim"
    android:layout_centerHorizontal="true"
    android:visibility="visible" />
<TextView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_above="@+id/centerVerticalShim"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_toLeftOf="@+id/centerHorizontalShim"
    android:background="#008B8B"
    android:gravity="center"
    android:id="@+id/textView6" />
<TextView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_above="@+id/centerVerticalShim"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_toRightOf="@+id/centerHorizontalShim"
    android:background="#008B8B"
    android:gravity="center"
    android:id="@+id/textView10" />
<TextView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/centerVerticalShim"
    android:layout_toLeftOf="@+id/centerHorizontalShim"

```

```

        android:background="#008B8B"
        android:gravity="center"
        android:id="@+id/textView5" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/centerVerticalShim"
    android:layout_toRightOf="@+id/centerHorizontalShim"
    android:background="#008B8B"
    android:gravity="center"
    android:id="@+id/textView4" />

<GridView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:columnWidth="200dp"
    android:numColumns="2"
    android:verticalSpacing="300dp"
    android:horizontalSpacing="100dp"
    android:stretchMode="columnWidth"
    android:id="@+id/gridView"
    android:gravity="center"
    android:padding="5dp" />

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="SAVE"
    android:id="@+id/button4"
    android:background="#0000"
    android:layout_above="@+id/textView5"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />

<TextView
    android:id="@+id/testView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=" "
    android:textSize="25sp"
    android:textColor="#000"
    android:layout_alignBaseline="@+id/textView10"
    android:layout_alignBottom="@+id/textView10"

```

```

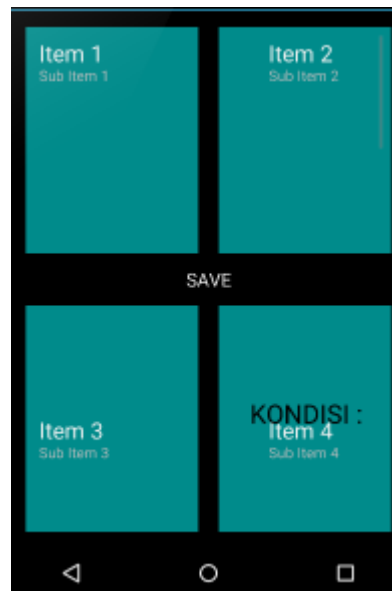
        android:layout_toRightOf="@+id/centerHorizontalShim"
        android:layout_toEndOf="@+id/centerHorizontalShim" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=" "
    android:textSize="25sp"
    android:textColor="#000"
    android:id="@+id/txtString"
    android:layout_alignBaseline="@+id/textView6"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_toRightOf="@+id/centerHorizontalShim"
    android:layout_toEndOf="@+id/centerHorizontalShim" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=" "
    android:textSize="25sp"
    android:textColor="#000"
    android:id="@+id/textView2"
    android:layout_alignBaseline="@+id/textView5"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_toRightOf="@+id/centerHorizontalShim"
    android:layout_toEndOf="@+id/centerHorizontalShim"/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="\t \t \tKONDISI : "
    android:textColor="#000"
    android:textSize="25sp"
    android:id="@+id/textView3"
    android:layout_alignBaseline="@+id/textView4"
    android:layout_alignBottom="@+id/textView4"
    android:layout_toRightOf="@+id/centerHorizontalShim"
    android:layout_toEndOf="@+id/centerHorizontalShim" />
</RelativeLayout>

```

Gambar 4.23 main_activity layout

Pada pembuatan *layout* *main_activity* ini menggunakan komponen tampilan android GridView, sehingga data yang ditampilkan dari pembacaan detak jantung, suhu tubuh, baterai, dan kondisi dalam bentuk format *grid* yang terdiri dari beberapa kolom dan baris.

Pada pembuatan *layout* ini juga dilengkapi dengan tombol *save* yang fungsinya untuk menyimpan hasil pembacaan dari alat ke dalam bentuk *table* SQLite



Gambar 4.24 Tampilan *main_activity layout*

b. Kode program MainActivity

Pada program MainActivity.java ini yang mengatur jalannya sistem aplikasi mulai dari koneksi bluetooth ponsel android ke alat penelitian, menampilkan data dari hasil pembacaan alat, dan menyimpan hasil pembacaan ke dalam bentuk SQLite dan dikirim ke *web*. Berikut adalah

kode program bluetooth menerima data dari arduino ditunjukkan pada gambar 4.22



```
bluetoothIn = (Handler) handleMessage(msg) -> {
    if (msg.what == handlerState) {
        String readMessage = (String) msg.obj;
        recDataString.append(readMessage);
        int endOfLineIndex = recDataString.indexOf("~");
        if (endOfLineIndex > 0) {
            int spacePOS = recDataString.indexOf("#");
            int spacePOS2 = recDataString.indexOf("$");
            int spacePOS3 = recDataString.indexOf("$");

            String dataInPrint = recDataString.substring(spacePOS + 1, spacePOS2);
            txtString.setText("\t \t \t \t \t \tBPM:\n \t \t \t" + dataInPrint + "bpm");
            dataInPrint = recDataString.substring(spacePOS2 + 1, spacePOS3);
            txtStringLength.setText("\t \t \t \t \t \tSUHU:\n" +
                " \t \t \t \t" + dataInPrint + "*C");
            dataInPrint = recDataString.substring(spacePOS3 + 1, endOfLineIndex);
            TextView2.setText("\t \t \t \t \t \tBATERAI:\n" +
                " \t \t \t \t \t \t" + dataInPrint);

            String regex = "[0-9]+";
            int BPM = 100;
            int SUHU = 3000;
            try{
                BPM = Integer.parseInt(recDataString.substring(spacePOS + 1, spacePOS2).trim().replaceAll("\\D+", ""));
                SUHU = Integer.parseInt(recDataString.substring(spacePOS2 + 1, spacePOS3).trim().replaceAll("\\D+", ""));
            }catch (Exception e){
            }
            String keterangan = "NORMAL";
            if ( (BPM>=60 && BPM<=100) && (SUHU>=3600 && SUHU<=3750) ) {
                keterangan = "SEHAT";
            }else{
                keterangan = "SAKIT"; //+String.valueOf(BPM)+" "+String.valueOf(SUHU);
            }
        }
    }
}
```

Gambar 4.25 listing program terima data dari arduino

Gambar 4.22 adalah program untuk memisahkan data-data dari arduino dan akan ditampilkan pada android. Teknik memisahkan data di android studio menggunakan indeks pemisah data, sehingga data dapat diketahui urutan data tersebut, kemudian data dari arduino akan dipisah atau di *substring* agar dapat ditempatkan sesuai tempatnya. Substring menggunakan fungsi `int spacePOS` dengan panjang data yang dihasilkan arduino yaitu 13 karakter. Sebagai contoh data dari arduino adalah `#85+36.19$87~` fungsi `spacePOS +1` sebagai kepala data yang masuk dan sebagai batas akan dimulainya substring, jadi karakter (`#`), (`$`), dan (`~`) tidak akan ikut ditampilkan pada android, selanjutnya hasil pembacaan data di konversi dalam bentuk *textview* yang dimasukkan ke dalam

gridview, berikut adalah program untuk mempresentasikan hasil pembacaan alat yang dimasukkan ke dalam *gridview* dalam bentuk teks yang ditunjukkan pada gambar 4.26

Selain menampilkan detak jantung dan suhu tubuh pada arduino, aplikasi ini dapat menampilkan kondisi. Terdiri dari dua kondisi yaitu kondisi sehat dan sakit, parameter yang digunakan untuk menentukan kondisi “SEHAT” akan muncul ketika BPM berkisar antara 60-100 dan suhu tubuh menunjukkan 36-37°C. Diluar parameter itu kondisi akan berubah menjadi “SAKIT”.

```

setContentView(R.layout.main_activity);

GridView gridView = (GridView) findViewById(R.id.
gridView.setAdapter(new ImageAdapter(this));

gridView.setOnItemClickListener((parent, view, position, id) → {
    Toast.makeText(MainActivity.this, "" + position, Toast.LENGTH_LONG).show();
});

txtString = (TextView) findViewById(R.id.txtString);
txtStringLength = (TextView) findViewById(R.id.txtString1);
TextView2 = (TextView) findViewById(R.id.txtView2);
TextView3 = (TextView) findViewById(R.id.txtView3);

mydb = new DBHelper(this);

btnSAVE = (Button) findViewById(R.id.button4);
btnSAVE.setOnClickListener((v) → {
    SimpleDateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss" );
    Date date = new Date();
    SimpleDateFormat dateFormat2 = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss" );
    mydb.insertSensor(dateFormat.format(date), txtString.getText().toString().replace("BPM:", ""), txtStringLength.getText().toS
insertdata(dateFormat2.format(date), txtString.getText().toString().replace("BPM:", "").replace("bpm", "").trim(), txtString
startActivity(new Intent(MainActivity.this, SAVE.class));
});

```

Gambar 4.26 konversi data dari arduino ke android

Pada program ini akan mengeksekusi pembuatan tampilan *gridview* yang telah termuat pada *layout* *main_activity*, selanjutnya pada objek *txtString*, *txtStringLength*, *textView2*, dan *textView3* digunakan untuk merepresentasikan nilai pada pembacaan bpm, suhu, baterai, dan kondisi, kemudian *method* *getText* digunakan untuk mendapatkan nilai hasil pembacaan data yang diterima dari arduino melalui bluetooth, sehingga

data yang disimpan dalam bentuk format data text. Data kemudian disimpan dalam bentuk tabel

5. Pembuatan database SQLite

Selain dapat digunakan sebagai alat monitoring pemantau detak jantung, suhu tubuh dan kondisi, aplikasi ini juga dilengkapi dengan fitur penyimpanan data tersebut dalam bentuk tabel *database* SQLite. Sehingga pengguna dapat menyimpan data tersebut setiap hari dan dapat digunakan sebagai perbandingan kondisi saat ini dengan kondisi yang telah lalu. Format hasil penyimpanan data berupa tabel yang berisi tanggal penyimpanan, detak jantung, suhu tubuh dan kondisi.

a. Pembuatan *layout database*

Pada pembuatan *layout database* ini melakukan pembuatan tampilan dua layout, tampilan pertama yaitu pembuatan tampilan tabel untuk menampilkan format penyimpanan data, yang kedua yaitu pembuatan tampilan untuk menuliskan data dalam tabel tersebut, adapun pembuatan *layout* tampilan tabel pada gambar 4.27 berikut

```

<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/No"
        android:layout_height="wrap_content"
        android:layout_width="0dp"
        android:layout_weight="0.5"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/tanggal"
        android:layout_height="wrap_content"
        android:layout_width="0dp"
        android:layout_weight="0.5"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/heartRate"
        android:layout_height="wrap_content"
        android:layout_width="0dp"
        android:layout_weight="0.5"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/temperature"
        android:layout_height="wrap_content"
        android:layout_width="0dp"
        />

```

Gambar 4.27 column_row layout

Pada gambar 4.27 menunjukkan pembuatan format kolom dan baris dalam penyimpanan dalam bentuk tabel SQLite, format penyimpanan terdiri dari No, tanggal, HeartRate, temperature, dan kondisi, selanjutnya untuk menuliskan isi tampilan tabel adalah sebagai berikut

```

<ScrollView
    android:layout_width="wrap_content"
    android:layout_height="300dp"
    android:id="@+id/scrollView" />

<ListView
    android:layout_width="match_parent"
    android:layout_height="296dp"
    android:id="@+id/listView2"
    android:layout_marginTop="40dp"
    android:layout_below="@+id/textView11"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="No"
    android:id="@+id/textView16"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Date"
    android:id="@+id/textView11"
    android:layout_marginLeft="43dp"
    android:layout_marginStart="43dp"

```

Gambar 4.28 simpan_layout

Pembuatan file XML simpan_layout ini yang akan menyimpan data dari pembacaan alat dalam bentuk list pada table sesuai dengan format penyimpanan pada *layout* column_row

b. Kode program *layout database* SQLite

Pembuatan kode program *layout database* SQLite ini menggunakan kelas java DBHelper, kelas ini tempat untuk membuat program penyimpanan data, berikut *listing* program SQLite pada kelas java DBHelper ditunjukkan pada Gambar 4.26



```
package com.example.jalu.ihheartcare;

import ...

public class DBHelper extends SQLiteOpenHelper {

    public static final String DATABASE_Bpm = "DB_Sensor.db";
    public static final String SENSOR_TABLE = "sensor";
    public static final String SENSOR_COLUMN_ID = "id";
    public static final String SENSOR_COLUMN_BPM = "bpm";
    public static final String SENSOR_COLUMN_SUHU = "suhu";
    public static final String SENSOR_COLUMN_KONDISI = "kondisi";
    public static final String SENSOR_COLUMN_TANGGAL = "tgl";
    //public static final String SENSOR_COLUMN_NOMOR = "no";

    public DBHelper(Context context) { super(context, DATABASE_Bpm, null, 1); }
    @Override
    //membuat database
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table "+SENSOR_TABLE+" (" + "(id INTEGER PRIMARY KEY AUTOINCREMENT, tgl text, bpm text, suhu text, k
    )
    @Override
    //untuk memperbaiki table
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS "+SENSOR_TABLE);
        onCreate(db);
    }
}
```

Gambar 4.29 Listing program tabel SQLite

Langkah pertama dalam pembuatan *file database* dengan nama DB_Sensor, data yang tersimpan akan masuk ke dalam *database* ini, setelah dibuat *file database*, kemudian dibuat konstanta yang digunakan sebagai *query* format penyimpanan data dalam bentuk tabel, setelah itu mendeskripsikan tipe data dari konstanta yang telah dibuat tadi adapun konstanta yang dibuat sebagai format *query* penyimpanan data yaitu, id, bpm, suhu, kondisi, dan tgl, masing-masing format tersebut memiliki tipe data yang berbeda-beda

Data yang akan dimasukkan akan dipanggil dengan fungsi *method* insertSensor. Fungsi *method* insertSensor ini diletakkan pada kelas java MainActivity dan berikut adalah kode programnya pada Gambar 4.30

```
mydb.insertSensor(dateFormat.format(date), txtString.getText().toString().replace("BPM:", ""), txtStringLength.getText().toString().replace("SUHU:", ""), TextView3.getText().toString().replace("KONDISI", ""));
startActivity(new Intent(MainActivity.this, SAVE.class));
```

Gambar 4.30 *listing* program insertSensor

Gambar 4.30 menunjukkan bahwa data yang ditampilkan pada *layout* simpan_layout yaitu berupa bpm, suhu, dan kondisi akan dimasukkan ke dalam *database*. Kemudian data yang akan ditampilkan ini dimasukkan ke dalam tabel yang telah dibuat. Berikut kode program untuk menambahkan data ke tabel *database* yang ditunjukkan pada gambar 4.31

```
//untuk menambahkan data ke tabel sensor
public boolean insertSensor(String tgl, String bpm, String suhu, String kondisi){
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put("tgl", tgl);
    contentValues.put("bpm", bpm);
    contentValues.put("suhu", suhu);
    contentValues.put("kondisi", kondisi);
    db.insert(SENSOR_TABLE, null, contentValues);
    return true;
}
```

Gambar 4.31 *listing* program menambahkan data

Data yang tersimpan dan yang telah dimasukkan ke dalam tabel akan ditampung dengan menggunakan ArrayList. Berikut adalah program ArrayList pada gambar 4.32

```

main_activity.xml x  fragment_layout.xml x  AndroidManifest.xml x  MenuBluetooth.java x  MainActivity.java x  DBHelper.java x
}
private ArrayList<HashMap<String, String>> list;
public ArrayList<HashMap<String, String>> getAllSensor_Column(){
    list = new ArrayList<HashMap<String, String>>();
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor res = db.rawQuery("select * from "+SENSOR_TABLE, null);
    res.moveToFirst();
    int no = 1;
    while (res.isAfterLast() == false){
        HashMap<String, String> temp = new HashMap<>();
        temp.put(FIFTH_COLUMN, String.valueOf(no));
        temp.put(FIRST_COLUMN, res.getString(res.getColumnIndex(SENSOR_COLUMN_BPM)));
        temp.put(SECOND_COLUMN, res.getString(res.getColumnIndex(SENSOR_COLUMN_SUHU)));
        temp.put(THIRD_COLUMN, res.getString(res.getColumnIndex(SENSOR_COLUMN_KONDISI)));
        temp.put(FOURTH_COLUMN, res.getString(res.getColumnIndex(SENSOR_COLUMN_TANGGAL)));
        list.add(temp);
        res.moveToNext();
        no=no+1;
    }
    return list;
}
}

```

Gambar 4.32 *listing* program ArrayList SQLite

Pengguna juga dapat menghapus data yang telah disimpan, tetapi dalam aplikasi ini tombol hapus akan menghapus semua data yang telah disimpan jadi tabel penyimpanan data akan kosong, berikut adalah program untuk menghapus data pada gambar 4.33

```

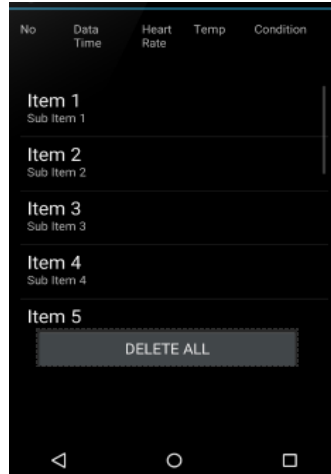
public Integer deleteSensor(long id)
{
    SQLiteDatabase db = this.getWritableDatabase();
    return db.delete(SENSOR_TABLE, SENSOR_COLUMN_ID+"=?", new String[]{Long.toString(id)});
}

public Integer deleteAll()
{
    SQLiteDatabase db = this.getWritableDatabase();
    return db.delete(SENSOR_TABLE, null, null);
}

```

Gambar 4.33 *listing* program hapus data

Hasil penyimpanan data yaitu disimpan dalam satu *database* yang dimasukkan ke tabel setelah pengguna menekan tombol simpan pada *layout* simpan_layout. Berikut hasil penyimpanan data SQLite dapat dilihat pada Gambar 4.34



Gambar 4.34 Tampilan tabel penyimpanan data

6. Pembuatan koneksi *database* SQLite ke MySQL

Pembuatan koneksi *database* SQLite ke MySQL ini bertujuan untuk menghubungkan data yang telah tersimpan dalam SQLite dapat ditampilkan melalui *website*. Sebelum ditampilkan ke dalam *website* data dari SQLite disimpan dahulu ke dalam *database* MySQL, selanjutnya PHP akan mengolah pengiriman data dari MySQL untuk dikirimkan ke dalam *website*. Pembuatan koneksi *database* SQLite ke MySQL ini tidak menggunakan tampilan *layout*. Adapun program untuk koneksi *database* SQLite ke MySQL sebagai berikut

```
ProgressDialog pd;

String ServerURL = "http://iheartcare.net/index.php" ;
Button btnSave;
```

Gambar 4.35 serverURL

Langkah pertama yaitu membuat variabel dan objek yang digunakan untuk membuat koneksi dengan alamat *website*, dalam hal ini alamat *website* disimpan dalam variabel `ServerURL` dengan tipe data string dengan nama url *website* nya(<http://iheartcare.net/index.php>), selanjutnya objek progress dialog untuk membuat kotak dialog progres dari penyinkronan data SQLite ke MySQL.

```

public void insertdata(final String Dt, final String HeartRate, final
String Temp, final String Condition) {
    StringRequest PostRequest = new
StringRequest(Request.Method.POST,
"http://iheartcare.net/insertdata.php", new
Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Toast.makeText(MainActivity.this, "Berhasil " +
response, Toast.LENGTH_SHORT).show();
            pd.dismiss();
        }
    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(MainActivity.this, "Error",
Toast.LENGTH_SHORT).show();
        }
    }
){
    protected Map<String, String> getParams(){
        Map<String, String> params = new HashMap<>();
        params.put("Date", Dt);
        params.put("HeartRate", HeartRate);
        params.put("Temp", Temp);
        params.put("Condition", Condition);
        return params;
    }
};
pd = ProgressDialog.show(MainActivity.this, "Please Wait",

```

```

"Connecting", true);
    pd.setCancelable(true);
    Volley.newRequestQueue(this).add(PostRequest);
}
}

```

Gambar 4.36 Kode sinkronisasi SQLite ke MySQL

Pada kode sinkronisasi data SQLite ke MySQL ini menggunakan *method* insertSensor lagi karena pada proses ini *method* insertSensor akan mengalami dua kerja yaitu kerja yang pertama *method* insertSensor akan mengirimkan data ke SQLite dan yang kedua mengirimkan data ke MySQL

Untuk dapat berkomunikasi dengan server MySQL yaitu pada alamat <http://iheartcare.net/insertdata.php> dalam program ini menggunakan *method* POST sehingga data dapat dengan secara langsung mengirimkan data melalui *library* Volley, untuk dapat mengirimkan data dari aplikasi android dengan *web* diperlukan suatu koneksi internet, sehingga perlu ditambahkan permission di AndroidManifest.xml

```

<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.INTERNET"/>

```

Gambar 4.37 Internet permission

Proses menghubungkan aplikasi dengan server *web* ditandai dengan progress dialog, progress dilog ini yang menunjukkan proses penghubungan antara aplikasi dan *website* sedang berlangsung ditandai dengan (“Please Wait”, Connecting”). Adapun program sebagai berikut pada gambar 4.38

```

    pd = ProgressDialog.show(MainActivity.this, "Please Wait", "Connecting", true);
    pd.setCancelable(true);

    Volley.newRequestQueue(this).add(PostRequest);
}
}

```

Gambar 4.38 Progress Dialog koneksi bluetooth

Pengiriman data dari aplikasi android dengan *web* akan berhasil atau dengan kata lain server merespon dari yang diminta *client* jika pada saat menyimpan data dari tombol “SAVE” pada android ditandai dengan adanya progress dialog dengan indikator tulisan “berhasil”, jika tidak terhubung dengan server biasanya tidak ada koneksi internet sehingga error

```

    public void onResponse(String response) {
        Toast.makeText(MainActivity.this, "Berhasil " + response, Toast.LENGTH_SHORT).show();

        pd.dismiss();
    }
},
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(MainActivity.this, "Error", Toast.LENGTH_SHORT).show();
        }
    }
}

```

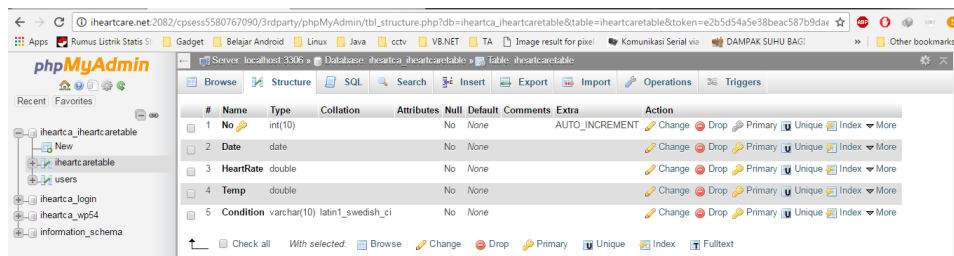
Gambar 4.39 Response sever MySQL

4.3.2.3 Pembahasan pemrograman Web

Pembahasan pemrograman *website* ini akan membahas tentang, membuat *database* MySQL, pembuatan tampilan *website*, sinkronisasi SQLite ke MySQL, dan membuat tampilan grafik

1. Pembuatan *database* MySQL

Pembuatan *database* MySQL digunakan sebagai penyimpanan data hasil pembacaan suhu, detak jantung, dan kondisi yang akan dipanggil oleh *web client* untuk menampilkan dalam bentuk grafik. Pembuatan *database* ini dibuat di phpMyAdmin pada cpanel.



Gambar 4.40 Struktur tabel iheartcetable

Pada struktur tabel iheartcetable ini tersimpan dalam *file database* iheartca_iheartcetable, adapun struktur tabel penyimpanan *database* sebagai berikut

- No, merupakan suatu urutan data yang masuk ke dalam *database* dengan tipe data integer dengan keterangan AUTO_INCREMENT artinya data yang masuk akan berurutan.
- Date, merupakan tempat penyimpanan tanggal ketika terjadi penyimpanan ke dalam *database* dengan tipe data *date*.
- HeartRate, merupakan tempat penyimpanan hasil pembacaan detak jantung dari alat penelitian dengan tipe data *double* karena data yang dihasilkan tidak bernilai genap atau berkoma.
- Temp, merupakan tempat penyimpanan hasil pembacaan suhu dari alat penelitian dengan tipe data *double*.

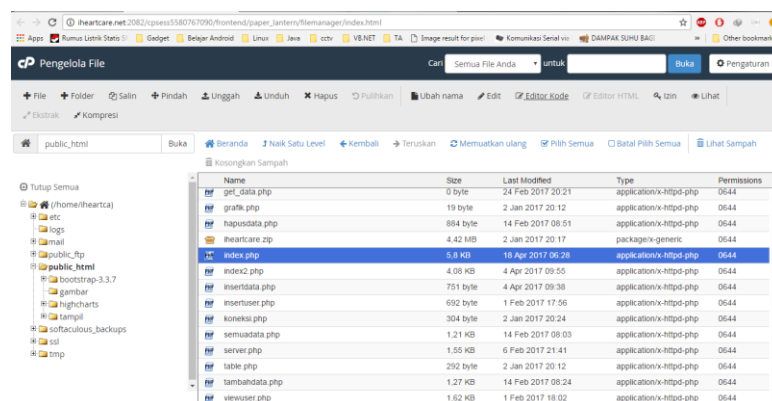
- Condition, merupakan tempat penyimpanan hasil pembacaan kondisi berdasarkan parameter detak jantung dan suhu tubuh, dan menggunakan tipe data *varchar* (karakter).

2. Pembuatan tampilan *website*

Pembuatan tampilan *website* ini digunakan untuk memuat tampilan *table* hasil pembacaan alat penelitian yang disimpan dalam *database* MySQL dan menampilkan tampilan grafik dari pengolahan data dari *database* MySQL. Bahasa pemrograman yang digunakan untuk membuat tampilan *website* ini menggunakan PHP.

a. Antarmuka *website*

Pembuatan antarmuka *website* ini tersimpan dalam *file* *index.php*, *file* ini otomatis dibuka saat membuka *website* dengan memanggil nama *domain website* letak *file* *index* ini berada pada direktori utama atau direktori root sebuah *website*



Gambar 4.41 *index.php*

Tampilan antarmuka *website* ini akan menampilkan *tab menu* yang terdiri dari menu home, contact, dan about, adapun program pada gambar 4.42

```

114 <div class="navbar-header">
115 <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
116 <span class="sr-only">Toggle navigation</span>
117 <span class="icon-bar"></span>
118 <span class="icon-bar"></span>
119 <span class="icon-bar"></span>
120 </button>
121 <a class="navbar-brand" href="#">Iheartcare</a>
122 </div>
123
124 <!-- Collect the nav links, forms, and other content for toggling -->
125 <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
126 <ul class="nav navbar-nav navbar-right">
127 <li><a href="#">home</a></li>
128 <li><a href="#">contact</a></li>
129 <li><a href="#">about</a></li>
130 </ul>
131 </div>
132 </nav>
133 </div>
134 </nav>
135 <div class="container-fluid">
136 
137 </div>

```

Gambar 4.42 *tab menu.php*

b. Tampilan tabel

Pembuatan tampilan tabel ini menggunakan pemrograman PHP MySQL, untuk menampilkan data dalam bentuk tabel diperlukan untuk mengambil data dari MySQL dengan menggunakan bahasa pemrograman PHP, adapun program sebagai berikut

```

1 <?php
2 //include("koneksi.php");
3
4 $dbhost = "localhost";
5 $dbuser = "iheartca_ihatable";
6 $dbpass = "r1na1d1963969";
7 $dbname = "iheartca_iheartcetable";
8
9 mysql_connect($dbhost,$dbuser,$dbpass);
10 mysql_select_db($dbname);
11

```

Gambar 4.43 *mysql_connect*

Pembuatan program di atas digunakan untuk menghubungkan *server* MySQL dan mengakses data yang ada pada *file database* *iheartca_iheartcetable*.

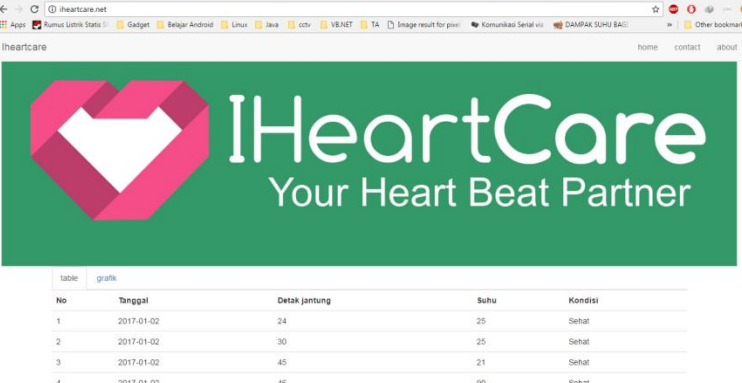
```

echo "<div class='table-responsive'>
<table class='table'>
<tr>
<th>No</th>
<th>Tanggal</th>
<th>Detak jantung</th>
<th>Suhu</th>
<th>Kondisi</th>
</tr>";
while($r=mysql_fetch_array($q)){
echo"<tr>
<td>$r[No]</td>
<td>$r[Date]</td>
<td>$r[HeartRate]</td>
<td>$r[Temp]</td>
<td>$r[Condition]</td>";
}
echo "</table>";
</div>"

```

Gambar 4.44 table.PHP

Pembuatan *table* PHP ini digunakan untuk merepresentasikan data yang diambil dari *server* MySQL, struktur tabel penyimpanan terdiri dari No, Tanggal, Detak Jantung, Suhu, dan Kondisi. Sehingga tampilan tabel nya sebagai berikut

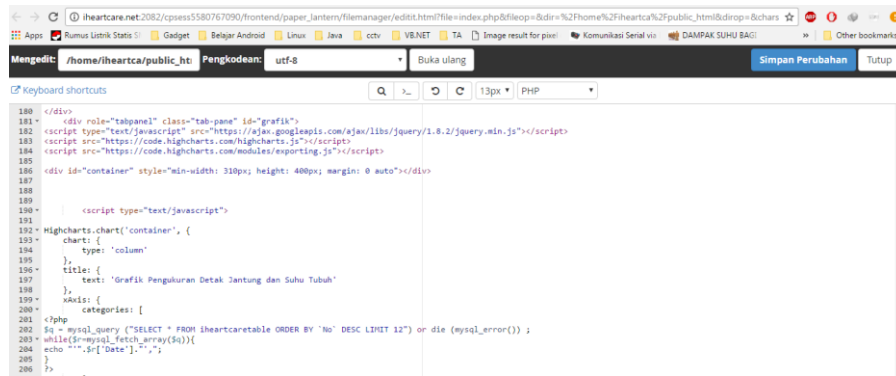


No	Tanggal	Detak jantung	Suhu	Kondisi
1	2017-01-02	24	25	Sehat
2	2017-01-02	30	25	Sehat
3	2017-01-02	45	21	Sehat
4	2017-01-02	46	20	Sehat

Gambar 4.45 tampilan *table website*

c. Tampilan grafik

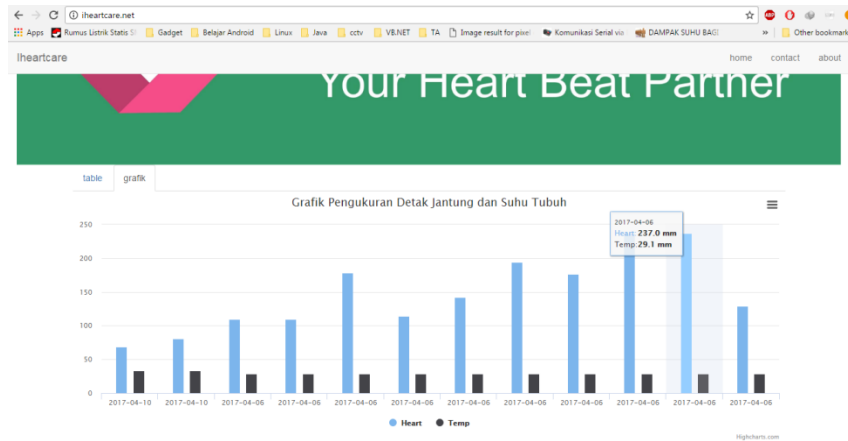
Dari hasil penyimpanan data dari MySQL data tersebut diolah kembali untuk ditampilkan dalam bentuk grafik, dengan adanya grafik pengguna lebih mudah dalam memantau kesehatannya



```
180 </div>
181 <div role="tabpanel" class="tab-pane id="grafik">
182 <script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
183 <script src="https://code.highcharts.com/highcharts.js"></script>
184 <script src="https://code.highcharts.com/modules/exporting.js"></script>
185
186 <div id="container" style="min-width: 310px; height: 400px; margin: 0 auto"></div>
187
188
189
190 <script type="text/javascript">
191
192 Highcharts.chart('container', {
193   chart: {
194     type: 'column'
195   },
196   title: {
197     text: 'Grafik Pengukuran Detak Jantung dan Suhu Tubuh'
198   },
199   xaxis: {
200     categories: [
201 <php
202 $q = mysql_query ("SELECT * FROM iheartcaretable ORDER BY 'No' DESC LIMIT 12") or die (mysql_error());
203 while($=mysql_fetch_array($q)){
204 echo "'".$q['Date']."',";
205 }
206 >
207 ]
208 }
209 }
210 </script>
```

Gambar 4.46 Grafik.php

Pembuatan tampilan grafik menggunakan *library* dari *highcharts* karena penggunaannya lebih mudah dan fleksibel, dalam pembuatan grafik ini pada sumbu X menggunakan kategori tanggal, sedangkan pada sumbu Y menggunakan kategori detak jantung dan suhu tubuh, dalam menampilkan data grafik akan menampilkan hasil dari 12 penyimpanan terakhir penyimpanan dalam MySQL, sehingga data sebelumnya akan hilang dan tidak ditampilkan



Gambar 4.47 Antarmuka grafik

3. Sinkronisasi SQLite ke MySQL

Sinkronisasi SQLite ke MySQL ini bertujuan untuk menyamakan data yang tersimpan dalam SQLite sama dengan MySQL. Method POST ini digunakan untuk merespon data yang masuk dari SQLite, setelah keduanya saling terhubung maka data akan dimasukkan ke dalam *database* MySQL.

```

12 /*
13 $Date = $_POST['Date'];
14 $HeartRate = $_POST['HeartRate'];
15 $Temp = $_POST['Temp'];
16 $Condition = $_POST['Condition'];
17
18 $Sql_Query = "insert into GetDataTable (Date,HeartRate,Temp,Condition) values ('$Date','$HeartRate','$Temp','$Condition')";
19
20 if(mysql_connect($con,$Sql_Query)){
21     echo 'Data Submit Successfully';
22 }
23 }
24 else{
25     echo 'Try Again';
26 }
27 }
28 }
29 }
30 mysql_close($con);
31 */

```

Gambar 4.48 Sinkronisasi SQLite ke MySQL

4.4 Pengujian kompatibilitas

Pengujian kompatibilitas ini dilakukan dengan menguji beberapa *platform* android yang beredar di pasar dari versi android 4.4.2 (KitKat) sampai dengan android 6.0 (Marshmallow) adapun pengujian *platform* adalah sebagai berikut

Tabel 4.6 Pengujian kompatibilitas

Tipe Smartphone	Platform	Hasil
Lenovo S850	Android KitKat 4.4.2	
Sony Xperia T2 Ultra	Android Lolipop (5.0)	
Sony Xperia Z2	Android Marshmallow (6.0)	

Pengujian aplikasi ini dapat diaplikasikan pada *platform* Android KitKat sampai dengan *platform* Android Marshmallow