

## BAB IV

### HASIL DAN PEMBAHASAN

Dalam bab ini penulis akan menguraikan dan menjelaskan hasil pengujian dari hasil penelitian tugas akhir ini.

#### 4.1 Kebutuhan Sistem

Sebelum melakukan transmisi sinyal suara jantung dibutuhkan perangkat lunak yang dapat menunjang penelitian. Perangkat keras dan lunak yang digunakan dapat dilihat pada Tabel 4.1 dan Tabel 4.2.

Tabel 4.1 Kebutuhan Perangkat Keras.

| Perangkat Keras | Spesifikasi   |
|-----------------|---------------|
| Processor       | CORE i7-4510U |
| Memori          | 8 Gb          |
| Sistem Operasi  | Windows 10    |

Tabel 4.2 Kebutuhan Perangkat Lunak.

| Perangkat Lunak | Uraian  |
|-----------------|---|
| Arduino IDE     | Aplikasi yang digunakan untuk mengolah sinyal suara jantung |

#### 4.2 Pengujian Xbee

Sebelum melakukan pengujian terhadap Xbee dibutuhkan *software* bernama X-CTU. *Software* ini dirancang oleh Digi untuk berinteraksi dengan Xbee yaitu mengkonfigurasi Xbee diawal sebelum melakukan komunikasi.

#### 4.2.1 Tujuan Pengujian Xbee

Pengujian ini dilakukan untuk mengetahui apakah Xbee yang digunakan dapat berfungsi dengan baik atau tidak.

#### 4.2.2 Alat yang Digunakan pada Pengujian Xbee

Untuk melakukan percobaan ini maka diperlukan beberapa alat sebagai berikut.

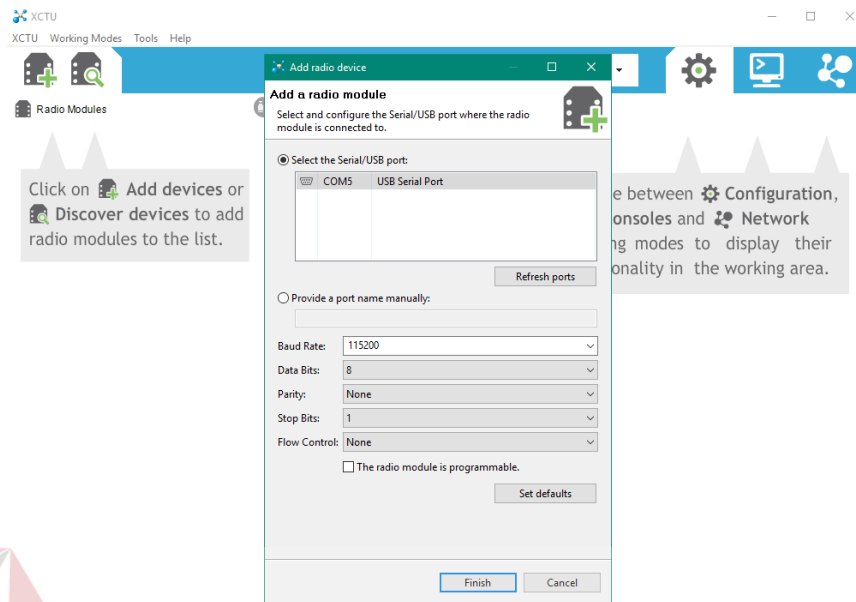
- a. Xbee adapter / Xbee *shield*
- b. Xbee S2
- c. Komputer / laptop
- d. *Software X-CTU*

#### 4.2.3 Prosedur Pengujian Xbee

Prosedur pengujian alat :

- a. Hubungkan Xbee adapter dengan kabel usb adapter.
- b. Hubungkan kabel usb adapternya ke komputer/laptop.
- c. Buka *software X-CTU* dan pilih “*add a radio module*”
- d. Akan muncul jendela baru untuk pemilihan PORT sehingga dapat diketahui apakah Xbee yang digunakan dapat terbaca oleh X-CTU atau tidak kemudia pilih “*finish*”.

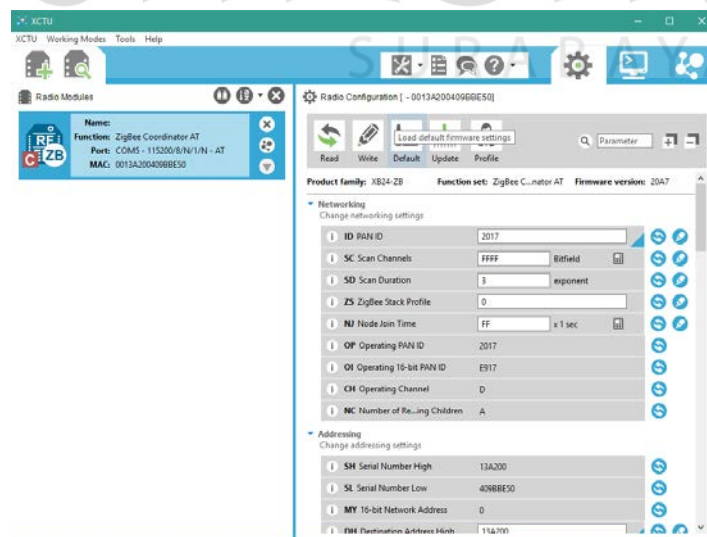
Berikut adalah tampilan awal pada *software X-CTU* yang terlihat pada Gambar 4.1.



Gambar 4.1 Tampilan Software X-CTU

#### 4.2.4 Hasil Pengujian Xbee

Seperti yang ditunjukkan pada Gambar 4.2 terlihat bahwa Xbee yang digunakan dapat terhubung dengan X-CTU. Sehingga Xbee dapat dikonfigurasi sesuai yang dibutuhkan.



Gambar 4.2 Tampilan Xbee dalam Keadaan Normal

### 4.3 Pengujian Komunikasi Xbee

Pengujian komunikasi Xbee dapat dilakukan dengan mengatur PAN ID, DL, DH. Komunikasi dapat terlihat ketika Xbee dari sisi *coordinator* dapat menerima pesan dari Xbee yang menjadi *router*.

#### 4.3.1 Tujuan Pengujian Komunikasi Xbee

Pengujian ini dilakukan untuk mengetahui apakah Xbee yang digunakan dapat berkomunikasi dengan baik.

#### 4.3.2 Alat yang Digunakan pada Pengujian Komunikasi Xbee

Alat yang digunakan untuk melakukan pengujian antara lain :

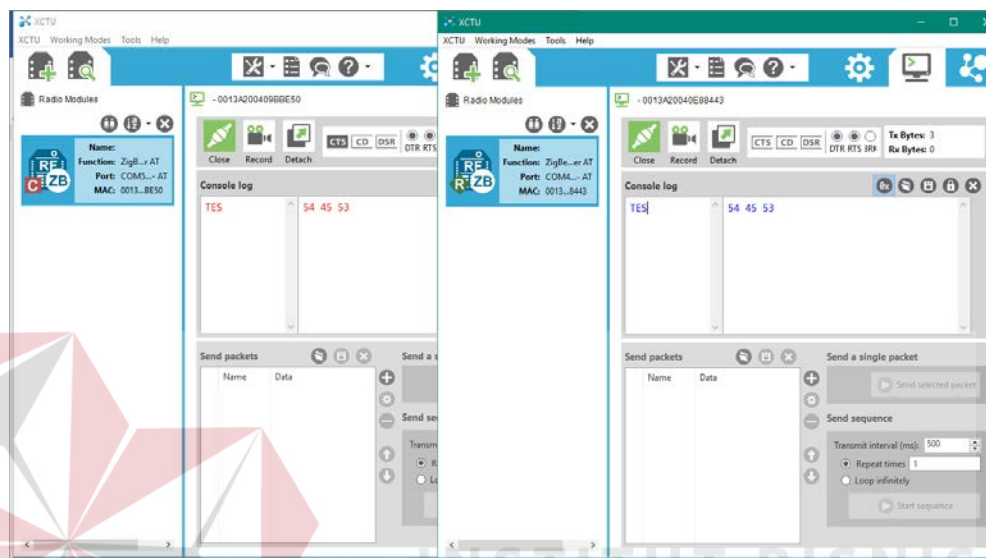
- a. Xbee adapter / Xbee shield
- b. Xbee S2
- c. Komputer / laptop
- d. Software X-CTU

#### 4.3.3 Prosedur Pengujian pada Pengujian Komunikasi Xbee

PAN ID kedua Xbee disamakan nilainya yaitu dengan nilai 2017, dan nilai DH pada Xbee diisikan dengan nilai SH pada Xbee yang lainnya, begitupun nilai DL pada Xbee diisikan dengan nilai SL pada Xbee yang lainnya.

#### 4.3.4 Hasil Pengujian pada Pengujian Komunikasi Xbee

Berikut adalah tampilan komunikasi *point to point* yang terlihat pada Gambar 4.3.



Gambar 4.3 Tampilan Komunikasi *Point-to-point* Xbee

Gambar diatas menunjukkan bahwa Xbee dapat berkomunikasi antara Xbee *router* dan *coordinator*. Hal ini ditandai dengan data yang diterima oleh Xbee *coordinator* sama dengan data yang dikirim oleh Xbee *router*.

#### 4.4 Pengujian Arduino

Pengujian pada Arduino dilakukan dengan meng-*upload script* program sederhana dengan menggunakan Arduino IDE. Jika berhasil mengeksekusi program yang dibuat, maka Arduino yang digunakan dalam kondisi yang baik.

#### 4.4.1 Tujuan Pengujian Arduino

Pengujian ini dilakukan untuk mengetahui apakah Arduino yang digunakan dalam kondisi baik atau mengalami kerusakan. Sehingga pada saat digunakan pada sistem dapat membantu sistem berjalan dengan baik.

#### 4.4.2 Alat yang Digunakan pada Pengujian Arduino

Alat yang digunakan untuk melakukan pengujian antara lain :

- a. Arduino Mega 2560
- b. Komputer/laptop
- c. Software Arduino IDE
- d. Kabel USB

#### 4.4.3 Prosedur Pengujian Arduino

- a. Hubungkan Arduino Mega 2560 dengan kabel USB
- b. Nyalakan komputer kemudian hubungkan Arduino Mega 2560 yang sudah terhubung dengan kabel USB tadi dengan komputer
- c. Buka *software* Arduino IDE dan isikan *script* dalam Bahasa C. Sebagai

contoh *script* yang dimasukkan penulis sebagai berikut :

```
void setup()
{
  Serial.begin(115200);
}

void loop()
{
  for(int a=0; a<5; a++)
  {
    Serial.print("TES - ");
    Serial.println(a);
  }
}
```

```

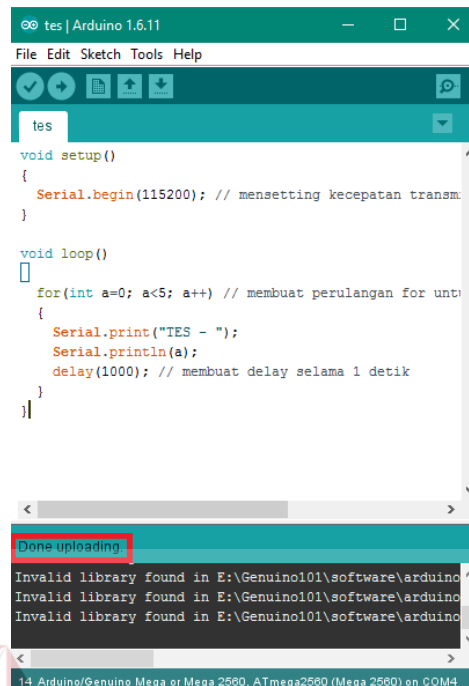
        delay(1000);
    }
}

```

- d. Apabila telah selesai untuk mengisi *script* nya, maka tekan “*Verify*” untuk mengecek apabila terdapat perintah yang salah. Kemudian pilih “*Upload*” untuk memasukkan *script* yang sudah dibuat kedalam Arduino mega 2560.
- e. Setelah program telah berhasil dimasukkan, maka tekan “*icon serial monitor*” pada kanan atas jendela Arduino IDE. Maka akan muncul tampilan *serial monitor*.
- f. Setelah windows *serial monitor* muncul, amati kiriman data *serial* oleh Arduino.

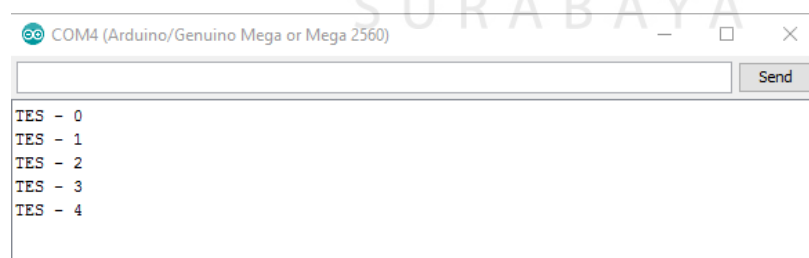
#### 4.4.4 Hasil Pengujian Arduino

Hasil dari pengujian pengisian *script* ke Arduino dapat dilihat seperti pada Gambar 4.4. Tulisan “*Done uploading*” menunjukkan bahwa Arduino yang digunakan berhasil diisi dengan *script* yang telah ditulis dalam *software* Arduino IDE.



Gambar 4.4 Tampilan *Upload Script* Berhasil

*Script* yang dimasukkan kedalam Arduino merupakan *script* untuk mengirimkan data menggunakan *serial*. Hasil dari *serial monitor* dapat dilihat seperti pada Gambar 4.5.



Gambar 4.5 *Script* Berhasil Berjalan

Pada Gambar 4.5 menunjukkan bahwa data yang dikirimkan sesuai dengan perintah *script* yang telah diisikan pada Arduino. Dengan begitu

Arduino tersebut dapat bekerja dengan baik dan dapat digunakan untuk sistem.

#### **4.5 Pengujian Besar *Buffer* pada Xbee**

Pada pengujian ini merupakan tahap untuk mengetahui berapakah besar *buffer* pada Xbee. Sehingga menjadi ketetapan untuk dilakukannya transmisi data hasil auskultasi jantung. Pengujian dilakukan dengan mengirimkan nilai urut sehingga dapat dilihat secara jelas pada posisi pengirim maupun penerima.

##### **4.5.1 Tujuan Pengujian Besar *Buffer* pada Xbee**

Pengujian ini bertujuan untuk mengetahui apakah paket data yang ditransmisikan dapat berjalan dengan aman. Sehingga dilakukan pengujian untuk mencoba seberapa besar data yang dapat diterima oleh *buffer* pada Xbee.

##### **4.5.2 Alat yang digunakan pada Pengujian Besar *Buffer* pada Xbee**

Alat yang digunakan untuk melakukan pengujian sistem ini antara lain:

- a. Arduino Mega 2560
- b. Xbee S2
- c. Xbee *shield* / Xbee adaptor
- d. Kabel USB

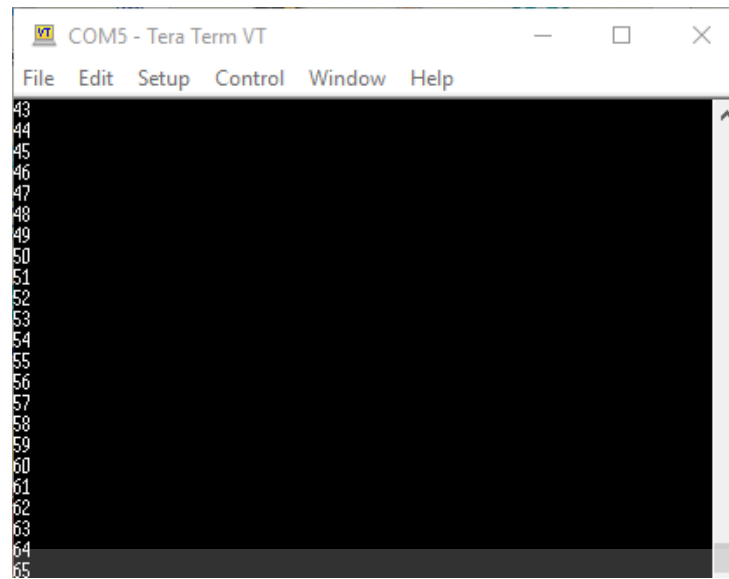
- e. *Computer* / laptop
- f. *Software* Arduino IDE
- g. *Software* Tera Term

#### 4.5.3 Prosedur Pengujian Besar *Buffer* pada Xbee

- a. Nyalakan *computer* dan buka program Arduino IDE
- b. Hubungkan Arduino dan *computer*
- c. *Upload script* yang digunakan dalam sistem
- d. Pengambilan data dilakukan selama 15 detik
- e. Amati pada *software* Tera Term dan *serial monitor* Arduino IDE, apakah data yang ditransmisikan dapat dilakukan secara baik
- f. Data yang ditransmisikan adalah data urut mulai dari 0-99

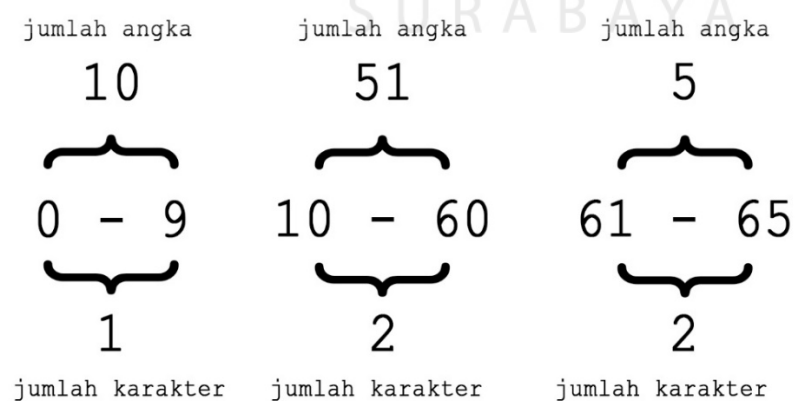
#### 4.5.4 Hasil Pengujian Besar *Buffer* pada Xbee

Pada penelitian ini, tampilan penerimaan data dilakukan dengan menggunakan aplikasi Tera Term. Terlihat jelas bahwa data yang diterima hanya sampai pada angka 65. Percobaan ini membuktikan bahwa *buffer* yang terdapat pada Xbee yaitu hanya 122 byte. Hasil percobaan pengujian ini dapat dilihat pada Gambar 4.6.



Gambar 4.6 Data yang Diterima pada Tera Term

Dari data yang diterima tersebut, dapat dilakukan perhitungan untuk mendapatkan nilai *buffer*. Perhitungan dilakukan dengan cara jumlah angka dikalikan jumlah karakter yang kemudian dijumlahkan. Maka nilai 122 ini didapat, seperti yang terlihat pada Gambar 4.7.



Gambar 4.7 Perhitungan *Buffer* pada Zigbee

#### **4.6 Pengujian *delay* antar paket data**

Pengujian ini merupakan tahap untuk mengetahui seberapa cepat *delay* yang dibutuhkan antar tiap paket data. Pengujian ini begitu penting sehingga dapat dipastikan data yang ditransmisikan berjalan baik.

##### **4.6.1 Tujuan Pengujian *Delay* Antar Paket Data**

Pengujian ini bertujuan untuk mengetahui berapakah *delay* yang dibutuhkan untuk setiap paket data yang dikirimkan melalui Xbee. Sehingga dengan mengetahui *delay*-nya proses tranmisi pada penelitian ini dapat berjalan dengan baik dan menjadi ketetapan untuk *delay* per paket datanya.

##### **4.6.2 Alat yang Digunakan pada Pengujian *Delay* Antar Paket Data**

Alat yang digunakan untuk pengujian sistem ini antara lain :

- a. Komputer / laptop
- b. Arduino Mega 2560
- c. Xbee S2
- d. Xbee *shield* / Xbee adaptor
- e. Kabel USB
- f. *Software* Tera Term
- g. *Software* Arduino IDE

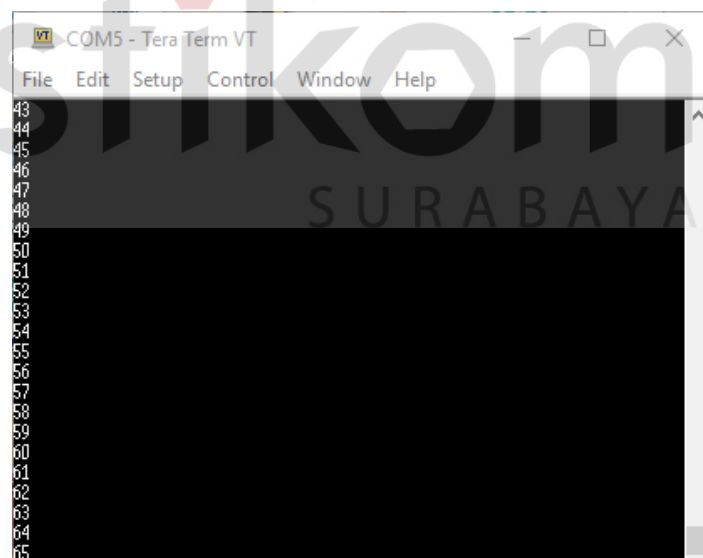
##### **4.6.3 Prosedur Pengujian *Delay* Antar Paket Data**

- a. Nyalakan komputer dan buka program Arduino IDE

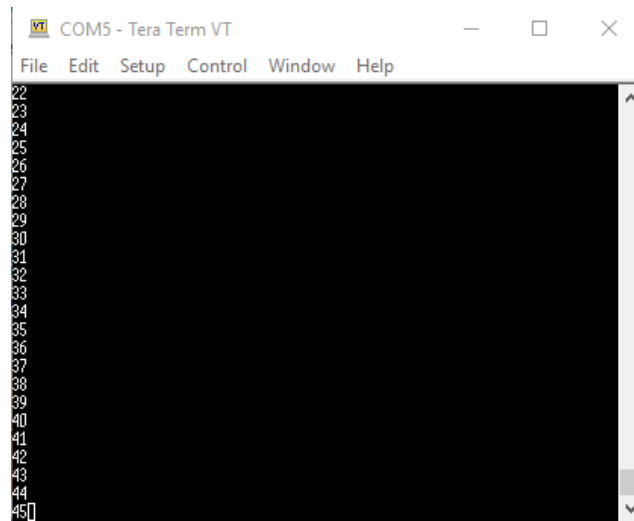
- b. Hubungkan Arduino dan komputer
- c. *Upload script* yang digunakan dalam sistem
- d. Amati pada *software* Tera Term dan *serial monitor* Arduino IDE, apakah data yang ditransmisikan dapat diterima dengan baik oleh penerima
- e. Data yang ditransmisikan adalah nilai urut mulai dari 0-65

#### 4.6.4 Hasil Pengujian *Delay* Antar Paket Data

Pada penelitian ini, *delay* antar tiap paket data sudah ditetapkan dari hasil pengujian ini. Sudah dilakukan beberapa kali percobaan untuk mendapatkan *delay* yang aman pada proses transmisi dilakukan. Sehingga didapatkan bahwa *delay* dalam penelitian ini adalah 60ms. Tampilan hasil dari proses pengujian ini dapat terlihat seperti Gambar 4.8 dan Gambar 4.9.



Gambar 4.8 Data yang Diterima Menggunakan *Delay* 60ms



Gambar 4.9 Data yang Diterima Menggunakan *Delay* Kurang dari 60ms

*Delay* dari pengujian diatas adalah *delay* yang digunakan dalam algoritma untuk membatasi antar paket data satu dengan yang lainnya. Sehingga ketika proses pengiriman paket data dilakukan akan berjalan dengan baik tanpa terjadi penumpukan paket data.

Pada penelitian ini *delay* yang dihitung adalah *delay* saat awal data dikirim hingga data pada sisi penerima diterima. Dilakukan dengan cara (jumlah data sebelum diterima x *interval sampling*).

#### 4.7 Pengujian Jumlah *Queue* pada Arduino Mega2560

Pengujian ini merupakan tahap untuk mengetahui seberapa besar memori yang dapat digunakan untuk *queue*. Sehingga jumlah *queue* dapat diatur sebagai pengolahan data sinyal jantung.

#### 4.7.1 Tujuan Pengujian Jumlah *Queue* pada Arduino Mega2560

Pengujian ini dilakukan untuk mengetahui banyak *queue* yang dapat disediakan dalam pengolahan data sinyal jantung.

#### 4.7.2 Alat yang Digunakan pada Pengujian Jumlah *Queue* pada Arduino Mega2560

Alat yang digunakan untuk pengujian sistem ini antara lain :

- a. Komputer / laptop
- b. Arduino Mega 2560
- c. Kabel USB
- d. *Software* Arduino IDE

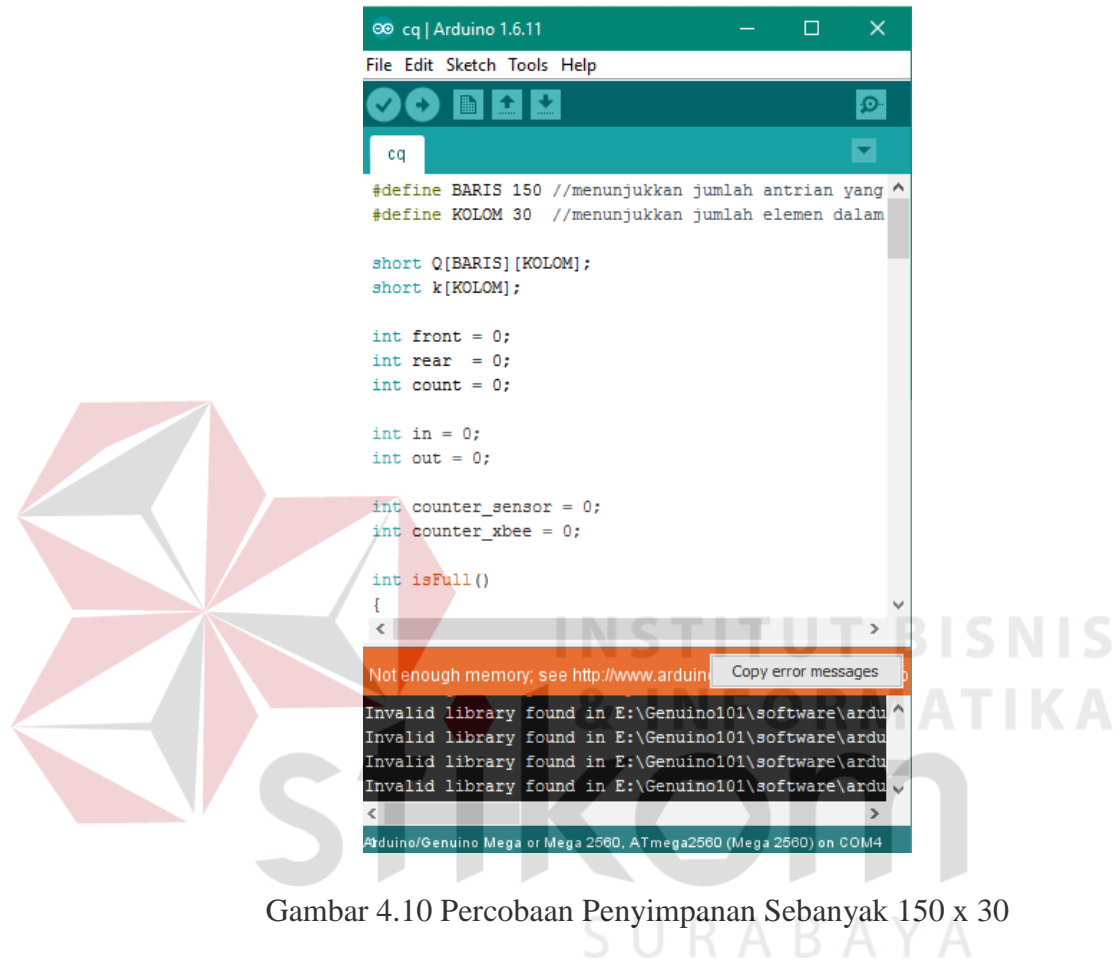
#### 4.7.3 Prosedur Pengujian Jumlah *Queue* pada Arduino Mega2560

- a. Nyalakan computer dan buka program Arduino IDE
- b. Hubungkan Arduino dan *computer*
- c. Upload *script* yang digunakan dalam sistem
- d. Ubah nilai BARIS sesuai yang dibutuhkan
- e. Amati pada Arduino IDE, apakah *script* yang di *upload* dapat berjalan baik

#### 4.7.4 Hasil Pengujian Jumlah *Queue* pada Arduino Mega2560

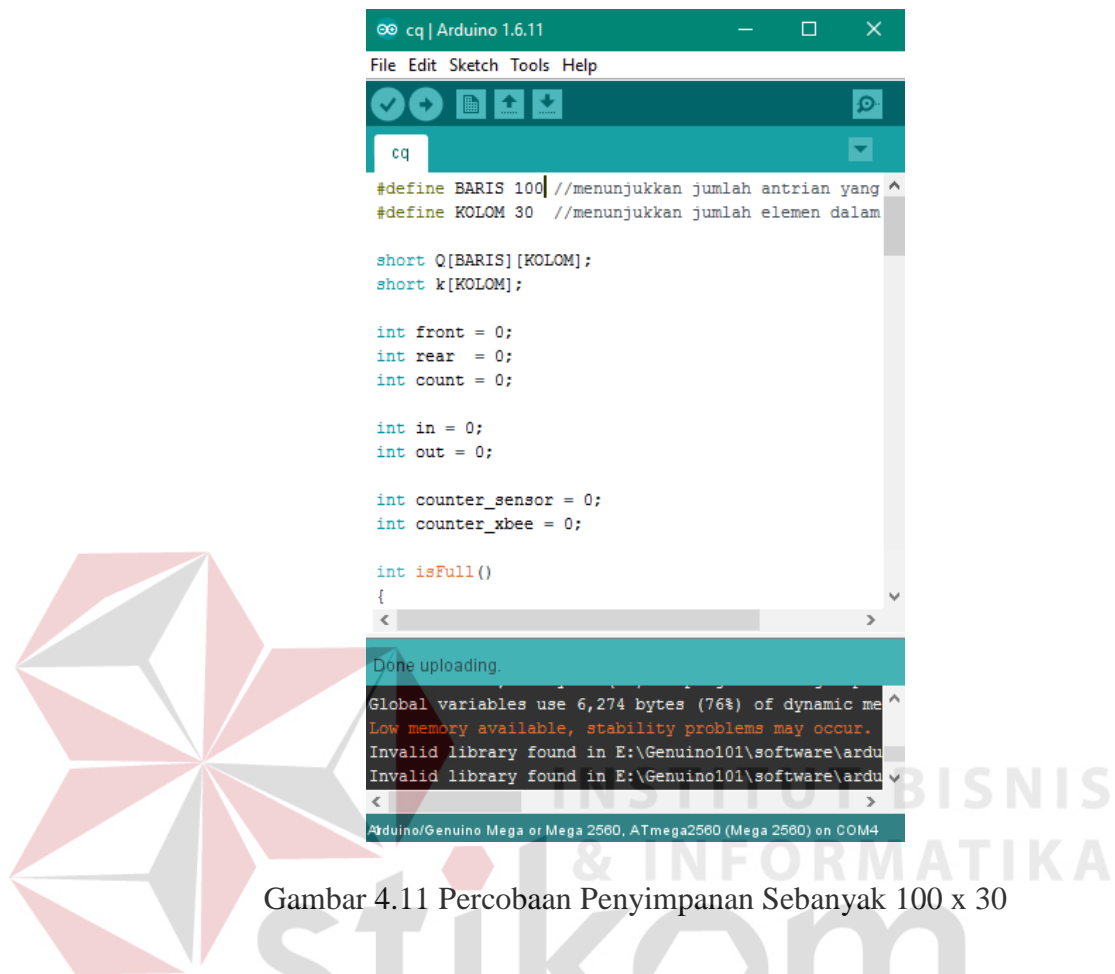
Pada penelitian ini, kapasitas maksimal memori yang digunakan untuk *queue* dengan array 2 dimensi adalah sebesar 97x30. Nilai yang didapat

adalah nilai paling maksimal yang dapat digunakan sebagai jumlah penyimpanan. Seperti yang ditunjukkan pada Gambar 10-12.



Gambar 4.10 Percobaan Penyimpanan Sebanyak 150 x 30

Dari pengujian diatas penggunaan array 2 dimensi dengan kapasitas penyimpanan sebanyak 150 x 30 tidak dapat mencukupi memori yang tersedia seperti tampilan pada gambar diatas.



Gambar 4.11 Percobaan Penyimpanan Sebanyak 100 x 30

Dari pengujian diatas penggunaan array 2 dimensi dengan kapasitas penyimpanan sebanyak 100 x 30 dapat disediakan oleh Arduino, tetapi terdapat pesan bahwa “*Low memory available, stability problems may occur*”. Sehingga proses transmisi yang dilakukan tidak dapat berjalan secara stabil.



Gambar 4.12 Percobaan Penyimpanan Sebanyak 97 x 30

Dari Gambar 4.12 diketahui bahwa penggunaan array 2 dimensi dengan kapasitas penyimpanan sebanyak 97 x 30 dapat disediakan oleh Arduino. Jumlah ini merupakan jumlah paling besar yang mampu disediakan oleh Arduino Mega2560.

## 4.8 Pengujian Sistem

Pengujian sistem ini merupakan pengujian untuk pengambilan data sinyal jantung agar data yang telah diambil dapat dianalisa *packet loss* dan *delay* dalam setiap mentransmisikan paket data sinyal jantung menuju sisi penerima.

### 4.8.1 Tujuan Pengujian Sistem

Bertujuan untuk mendapatkan data hasil transmisi auskultasi sinyal jantung. Serta mendapatkan berapa persen data yang hilang saat transmisi sinyal auskultasi jantung berlangsung. Sehingga dapat diketahui apakah transmisi auskultasi sinyal jantung pada sistem yang dibuat dapat berjalan dengan baik.

### 4.8.2 Alat yang Digunakan pada Pengujian Sistem

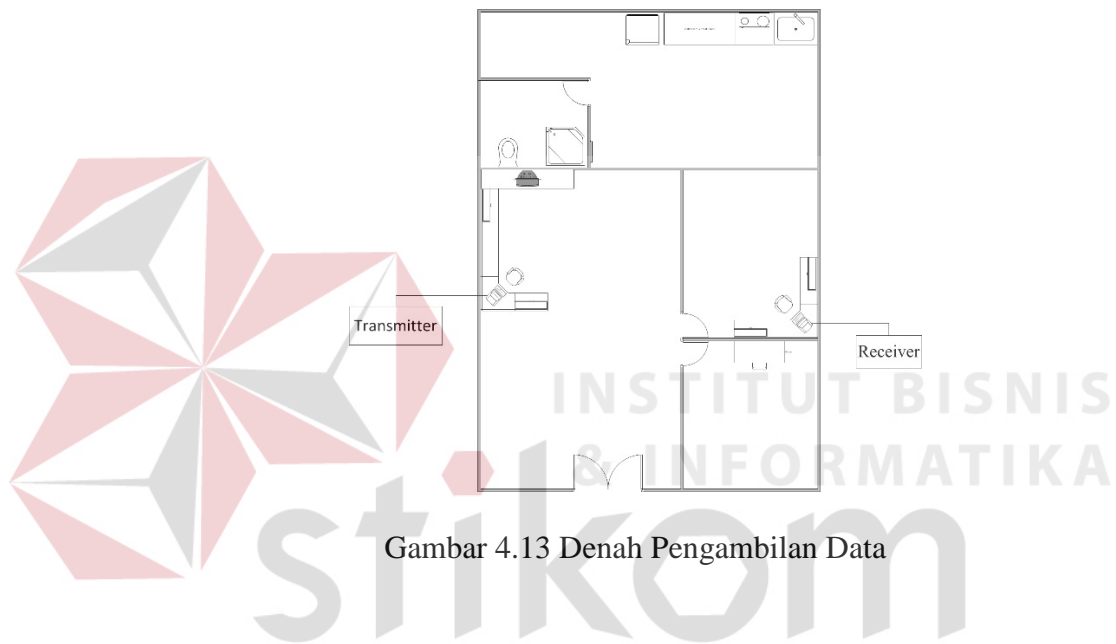
Alat yang digunakan untuk pengujian sistem ini antara lain :

- a. Komputer / laptop
- b. Arduino Mega 2560
- c. Xbee S2
- d. Xbee *shield* / Xbee adaptor
- e. *Heart Sound Sensor*
- f. Kabel USB
- g. *Timer*
- h. *Software Tera Term*

- i. *Software* Microsoft Excel

#### 4.8.3 Prosedur Pengujian Sistem

- a. Penentuan lokasi untuk pengambilan data auskultasi sinyal jantung yang terlihat pada Gambar 4.10.



Gambar 4.13 Denah Pengambilan Data

- b. Hubungkan Arduino dan komputer dengan menggunakan kabel USB
- c. Buka Arduino IDE dan kemudian buka *serial monitor*
- d. Buka aplikasi Tera Term pada sisi penerima
- e. Letakkan sensor pada *subject* uji agar mendapatkan sinyal jantung yang tepat
- f. Pengambilan data dilakukan selama 30 detik, untuk memperoleh sinyal jantung
- g. Amati data, apakah data dapat diterima oleh aplikasi dan sinyal yang diterima apakah merupakan sinyal jantung

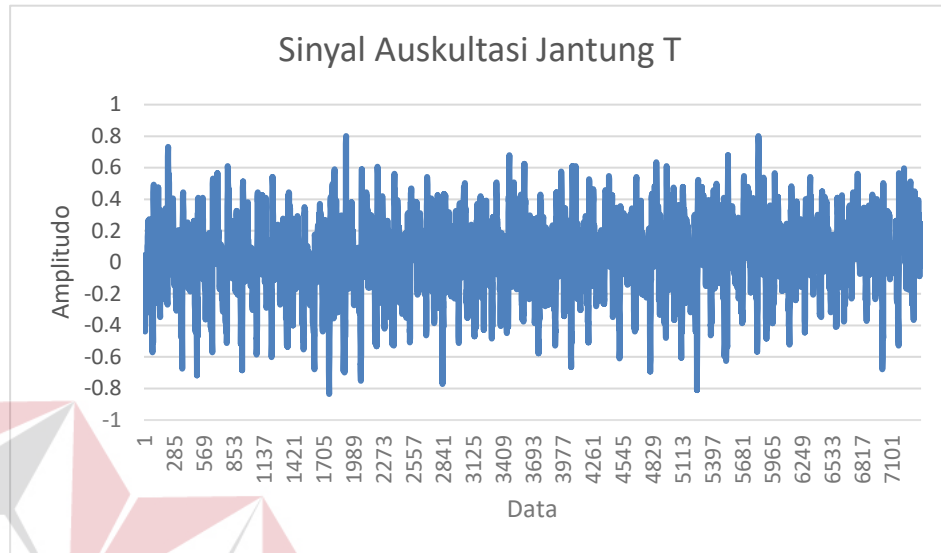
- h. Pada sisi penerima, amati hasil pengiriman data yang dilakukan pada aplikasi Tera Term
- i. Ketika sudah 30 detik hentikan proses pengambilan data dan kemudian *copy* data sinyal jantung pada sisi *transmitter* dan juga sisi *receiver* ke dalam file Excel
- j. Analisa data yang berasal dari *receiver* dan kemudian bandingkan dengan asal data untuk dianalisa

#### 4.8.4 Hasil Pengujian Sistem

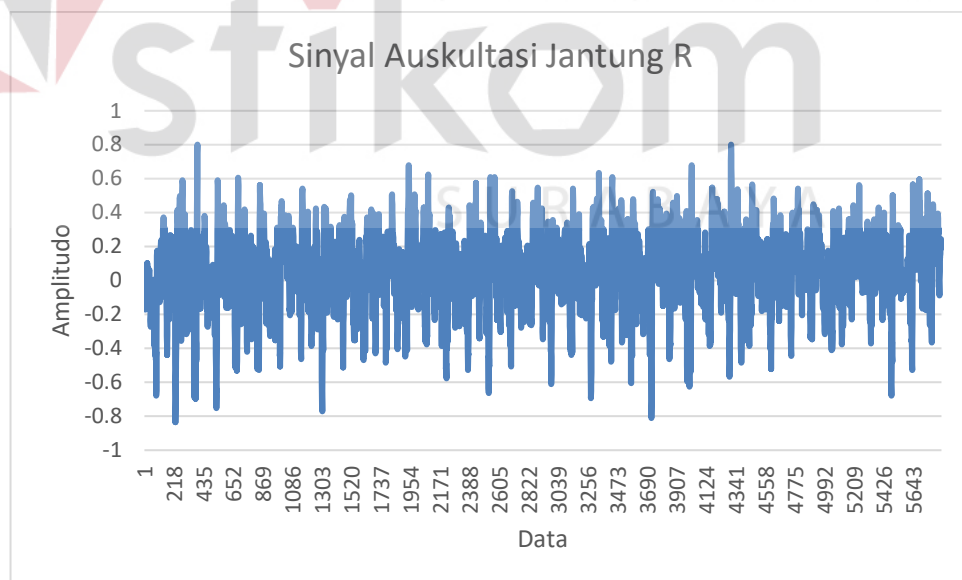
Pada penelitian ini, transmisi data hasil auskultasi sinyal jantung dilakukan beberapa kali variasi percobaan untuk mendapatkan kesimpulan dari penelitian yang telah dilakukan. Percobaan tersebut diantaranya adalah :

1. Pengiriman menggunakan *baudrate* 115200 (Tera Term)
  - a. Percobaan 1 dengan algoritma

Percobaan dilakukan dengan waktu 30 detik, dengan *interval sampling* 2ms seperti yang terlihat pada Gambar 4.14 dan Gambar 4.15.



Gambar 4.14 Percobaan 1 Data pada Sisi *Transmitter* dengan *Baudrate* 115200



Gambar 4.15 Percobaan 1 Data pada Sisi *Receiver* dengan *Baudrate* 115200

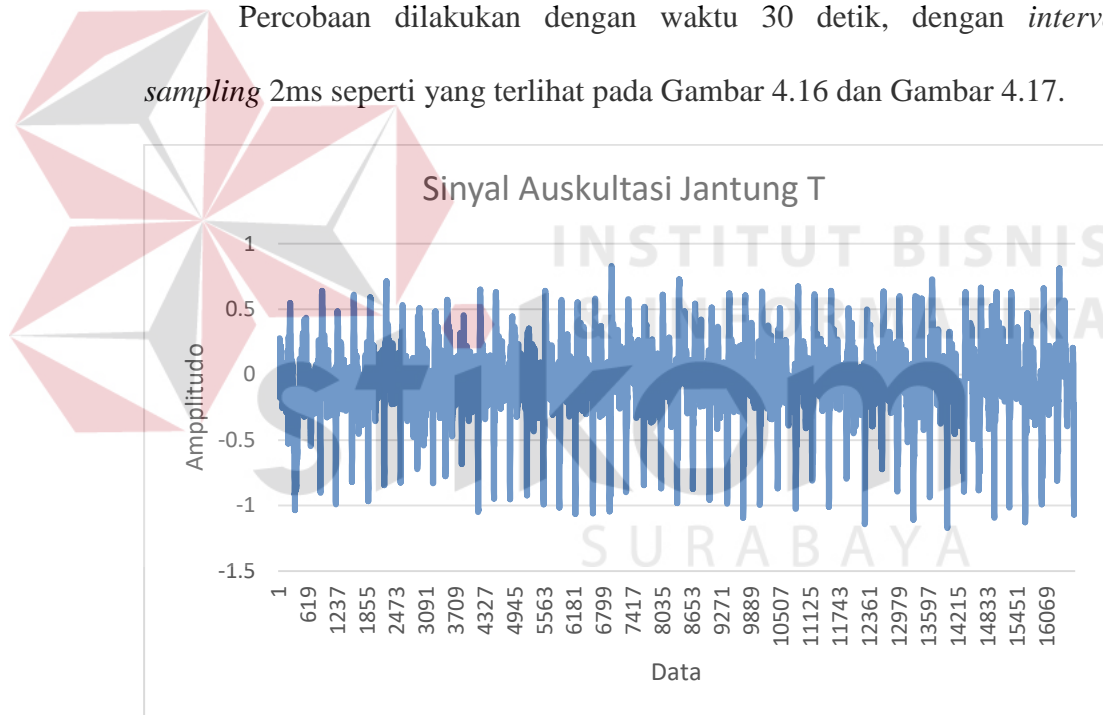
Dari Gambar 4.14 dan Gambar 4.15 didapatkan hasil perhitungan paket *loss* dan *delay* seperti Tabel 4.3 dibawah ini.

Tabel 4.3 Hasil Percobaan 1 dengan *Baudrate* 115200

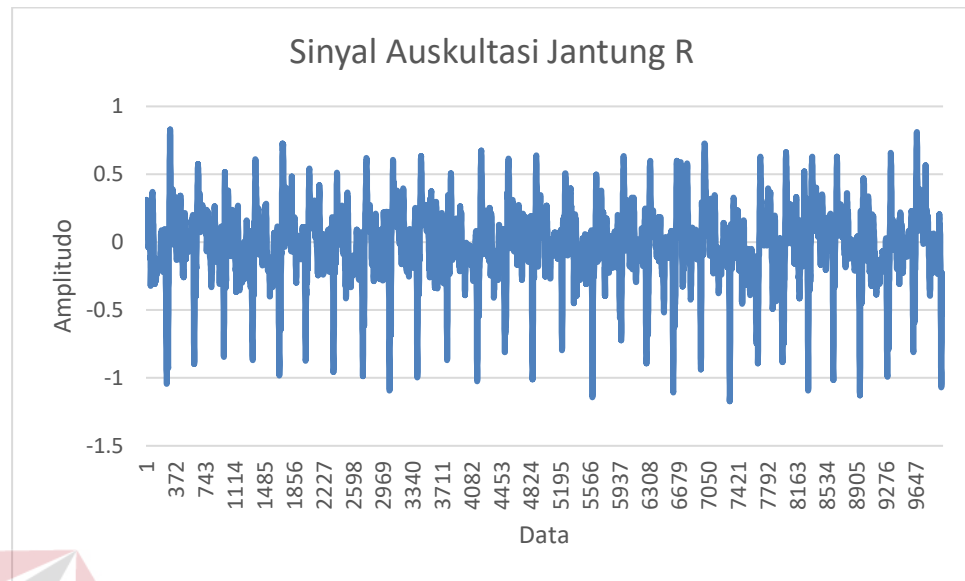
| <i>Delay</i> | <i>Packet loss</i> |
|--------------|--------------------|
| 0,102 s      | 20,73 %            |

b. Percobaan 2 tanpa algoritma

Percobaan dilakukan dengan waktu 30 detik, dengan *interval sampling* 2ms seperti yang terlihat pada Gambar 4.16 dan Gambar 4.17.



Gambar 4.16 Percobaan 2 Data pada Sisi *Transmitter* dengan *Baudrate* 115200



Gambar 4.17 Percobaan 2 Data pada Sisi *Receiver* dengan *Baudrate* 115200

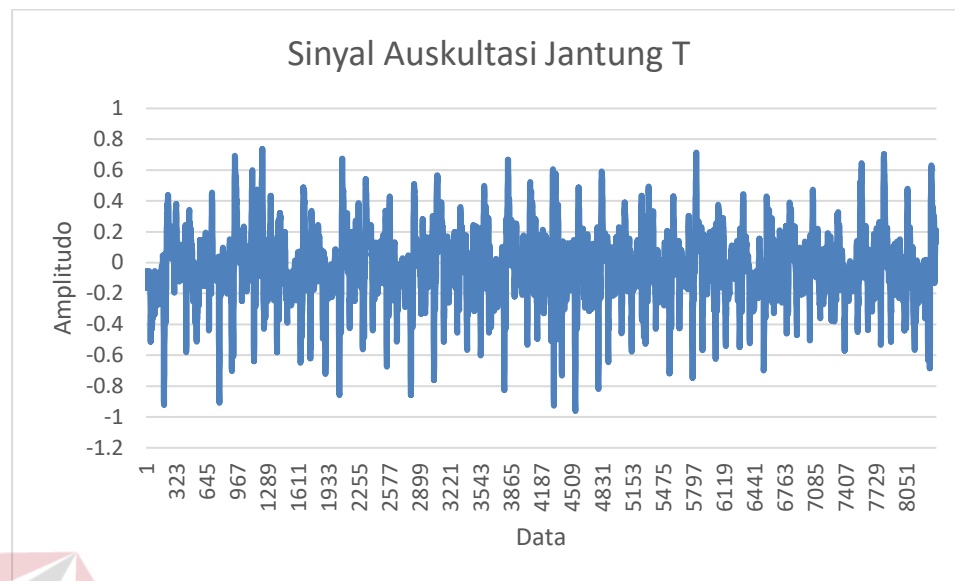
Dari Gambar 4.16 dan Gambar 4.17 didapatkan hasil perhitungan paket *loss* dan *delay* seperti Tabel 4.4 dibawah ini.

Tabel 4.4 Hasil Percobaan 2 dengan *Baudrate* 115200

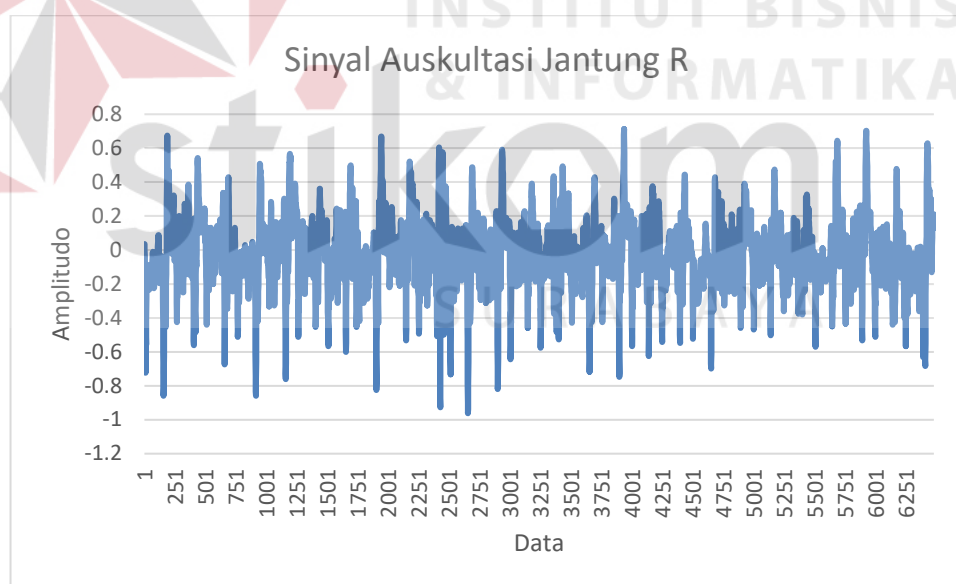
| <i>Delay</i> | <i>Packet loss</i> |
|--------------|--------------------|
| 13,348 s     | 40,04 %            |

c. Percobaan 3 dengan algoritma

Percobaan dilakukan dengan waktu 30 detik, dengan *interval sampling* 1ms seperti yang terlihat pada Gambar 4.18 dan Gambar 4.19.



Gambar 4.18 Percobaan 3 Data pada Sisi *Transmitter* dengan *Baudrate* 115200



Gambar 4.19 Percobaan 3 Data pada Sisi *Receiver* dengan *Baudrate* 115200

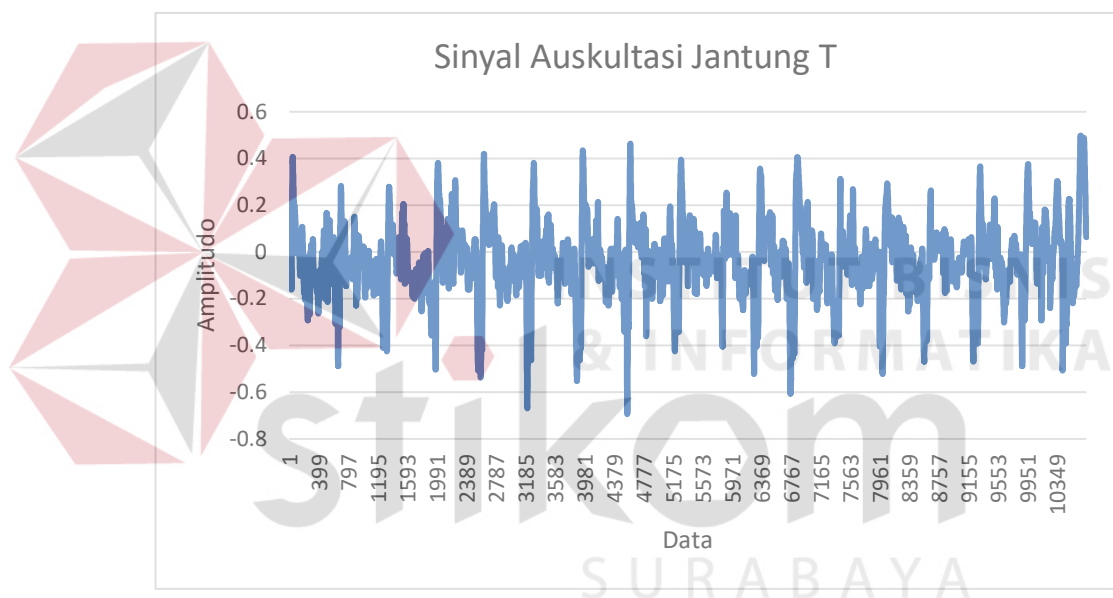
Dari Gambar 4.18 dan Gambar 4.19 didapatkan hasil perhitungan paket *loss* dan *delay* seperti tabel 4.5 dibawah ini.

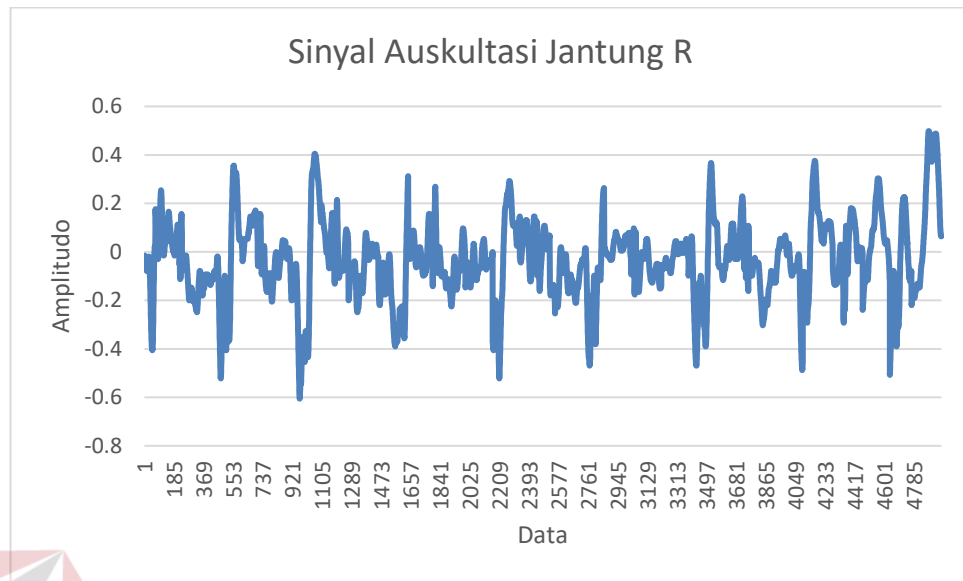
Tabel 4.5 Hasil Percobaan 3 dengan *Baudrate* 115200

| <i>Delay</i> | <i>Packet loss</i> |
|--------------|--------------------|
| 0,063 s      | 22,58 %            |

## d. Percobaan 4 tanpa algoritma

Percobaan dilakukan dengan waktu 30 detik, dengan *interval sampling* 1ms seperti yang terlihat pada Gambar 4.20 dan Gambar 4.21.

Gambar 4.20 Percobaan 4 Data pada Sisi *Transmitter* dengan *Baudrate* 115200



Gambar 4.21 Percobaan 4 Data pada Sisi *Receiver* dengan *Baudrate* 115200

Dari Gambar 4.20 dan Gambar 4.21 didapatkan hasil perhitungan paket *loss* dan *delay* seperti Tabel 4.6 dibawah ini.

Tabel 4.6 Hasil Percobaan 4 dengan *Baudrate* 115200

| <i>Delay</i> | <i>Packet loss</i> |
|--------------|--------------------|
| 5,786 s      | 53,87 %            |

Dari percobaan transmisi sinyal auskultasi jantung dengan menggunakan *baudrate* 115200 pada *interval sampling* 2ms dan 1ms dalam 4 kali percobaan dapat dilihat pada Tabel 4.7 dan Tabel 4.8.

Tabel 4.7 Hasil *Delay* Pengiriman dan juga *Packet Loss* pada *Baudrate* 115200 dengan *Interval Sampling* 2ms

| Teknik transmisi sinyal auskultasi jantung | <i>Packet loss</i> | <i>Delay</i> |
|--|--------------------|--------------|
| Dengan algoritma                           | 20,73 %            | 0,102 s      |
| Tanpa algoritma                            | 40,04 %            | 13,348 s     |

Tabel 4.8 Hasil *Delay* Pengiriman dan juga *Packet Loss* pada *Baudrate* 115200 dengan *Interval Sampling* 1ms

| Teknik transmisi sinyal auskultasi jantung | <i>Packet loss</i> | <i>Delay</i> |
|--|--------------------|--------------|
| Dengan algoritma                           | 22,58 %            | 0,063 s      |
| Tanpa algoritma                            | 53,87 %            | 5,786 s      |

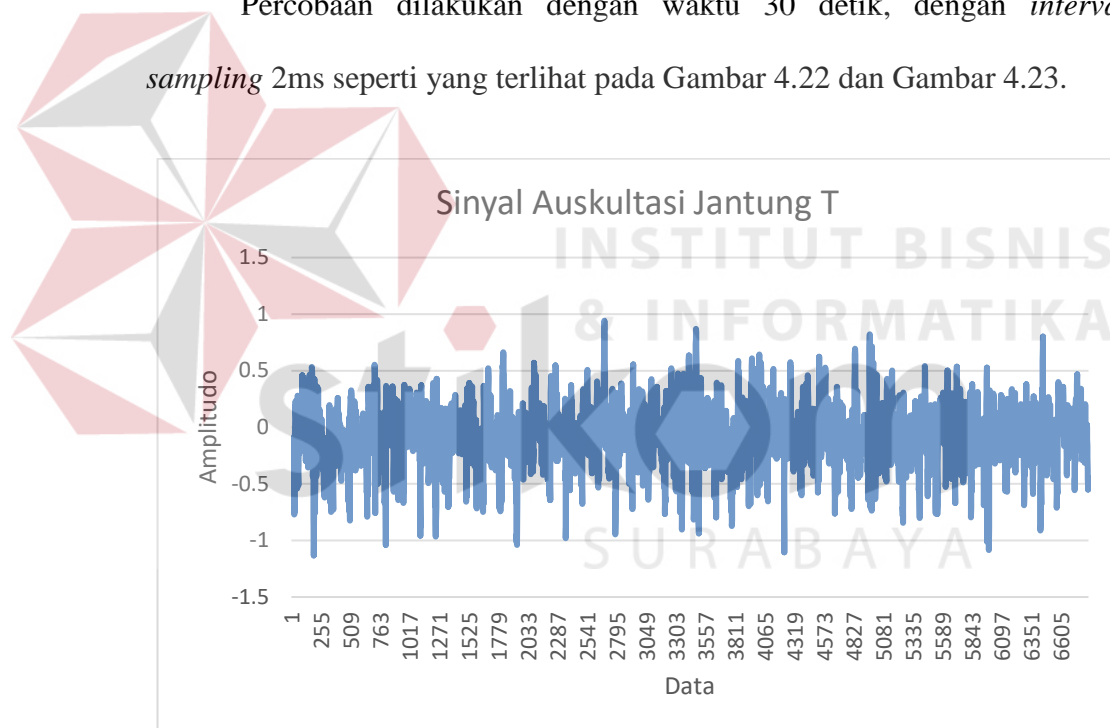
Dari data pada Tabel 4.7 dan Tabel 4.8 dapat disimpulkan bahwa transmisi sinyal auskultasi jantung dengan menggunakan algoritma dan tanpa menggunakan algoritma hasilnya sangatlah berbeda. Pada percobaan dengan algoritma pada *interval sampling* 2ms *packet loss* yang dihasilkan hanya 20,73% dan *delay* 0,102s sedangkan pada *interval sampling* 1ms *packet loss* yang dihasilkan 22,58% dan *delay* 0,063s. Mengacu pada Tabel 3.1 maka jumlah *packet loss* untuk sistem dengan algoritma menghasilkan kategori “sedang” untuk *interval sampling* 1ms dan 2ms. Sedangkan untuk sistem tanpa algoritma jumlah *packet loss* dikategorikan “jelek” untuk *interval sampling* 1ms dan 2ms.

Pengamatan terhadap Tabel 4.7 dan Tabel 4.8 menunjukkan bahwa packet loss dan delay lebih kecil untuk pengiriman data melalui Zigbee dengan algoritma yang diusulkan dibandingkan dengan pengiriman data tanpa menggunakan algoritma. Hal ini juga berlaku untuk interval sampling yang berbeda.

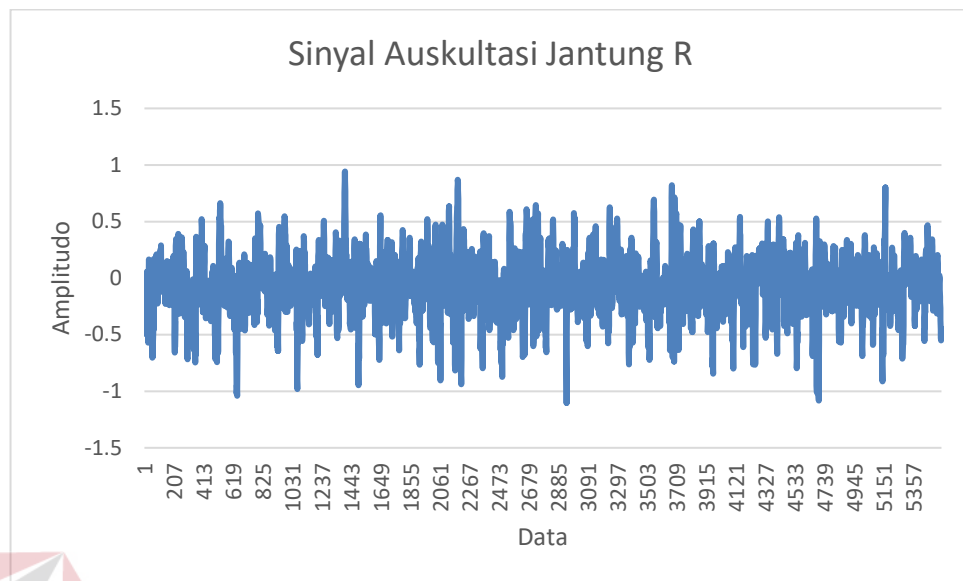
## 2. Pengiriman menggunakan *baudrate* 57600 (Tera Term)

### a. Percobaan 1 dengan algoritma

Percobaan dilakukan dengan waktu 30 detik, dengan *interval sampling* 2ms seperti yang terlihat pada Gambar 4.22 dan Gambar 4.23.



Gambar 4.22 Percobaan 1 Data pada Sisi *Transmitter* dengan *Baudrate* 57600



Gambar 4.23 Percobaan 1 Data pada Sisi *Receiver* dengan *Baudrate* 57600

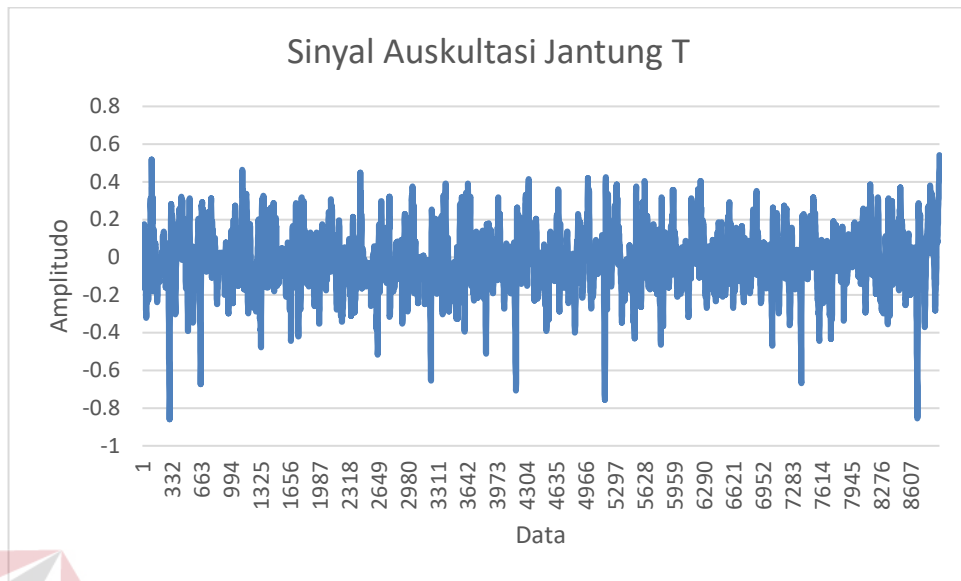
Dari Gambar 4.22 dan Gambar 4.23 didapatkan hasil perhitungan paket *loss* dan *delay* seperti Tabel 4.7 dibawah ini.

Tabel 4.9 Hasil Percobaan 1 dengan *Baudrate* 57600

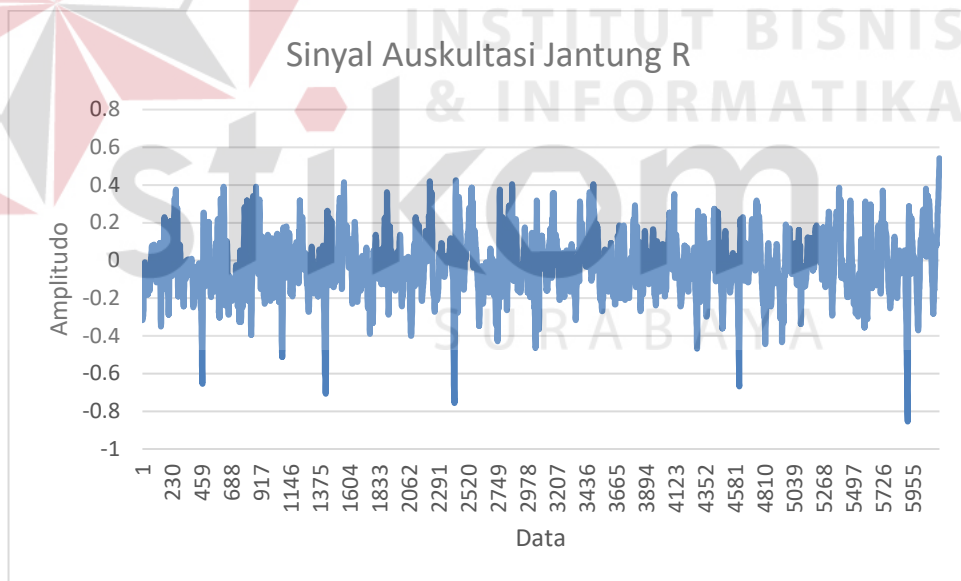
| <i>Delay</i> | <i>Packet loss</i> |
|--------------|--------------------|
| 0,086 s      | 18,85 %            |

b. Percobaan 2 tanpa algoritma

Percobaan dilakukan dengan waktu 30 detik, dengan *interval sampling* 2ms seperti yang terlihat pada Gambar 4.24 dan Gambar 4.25.



Gambar 4.24 Percobaan 2 Data pada sisi *Transmitter* dengan *Baudrate* 57600



Gambar 4.25 Percobaan 2 Data pada Sisi *Receiver* dengan *Baudrate* 57600

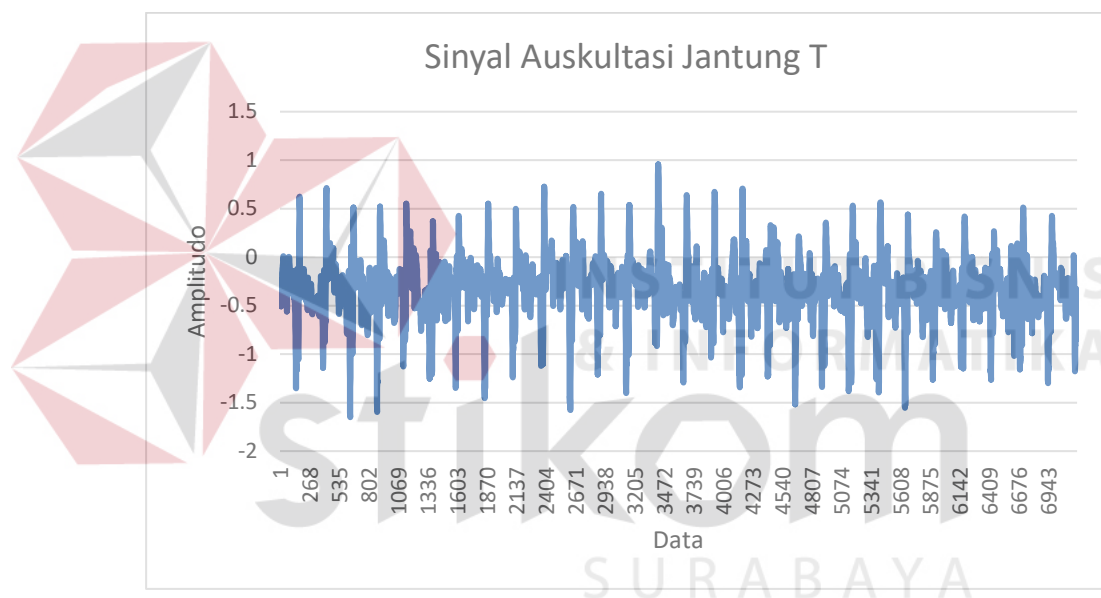
Dari Gambar 4.24 dan Gambar 4.25 didapatkan hasil perhitungan paket *loss* dan *delay* seperti Tabel 4.10 dibawah ini.

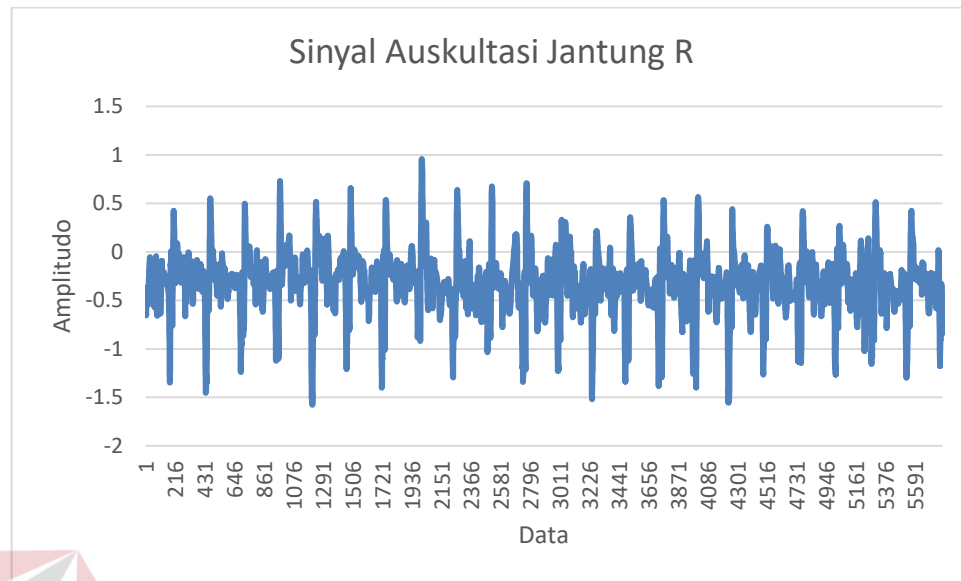
Tabel 4.10 Hasil Percobaan 2 dengan *Baudrate* 57600

| <i>Delay</i> | <i>Packet loss</i> |
|--------------|--------------------|
| 5,536 s      | 31 %               |

## c. Percobaan 3 dengan algoritma

Percobaan dilakukan dengan waktu 30 detik, dengan *interval sampling* 1ms seperti yang terlihat pada Gambar 4.26 dan Gambar 4.27.

Gambar 4.26 Percobaan 3 Data pada Sisi *Transmitter* dengan *Baudrate* 57600



Gambar 4.27 Percobaan 3 Data pada Sisi *Receiver* dengan *Baudrate* 57600

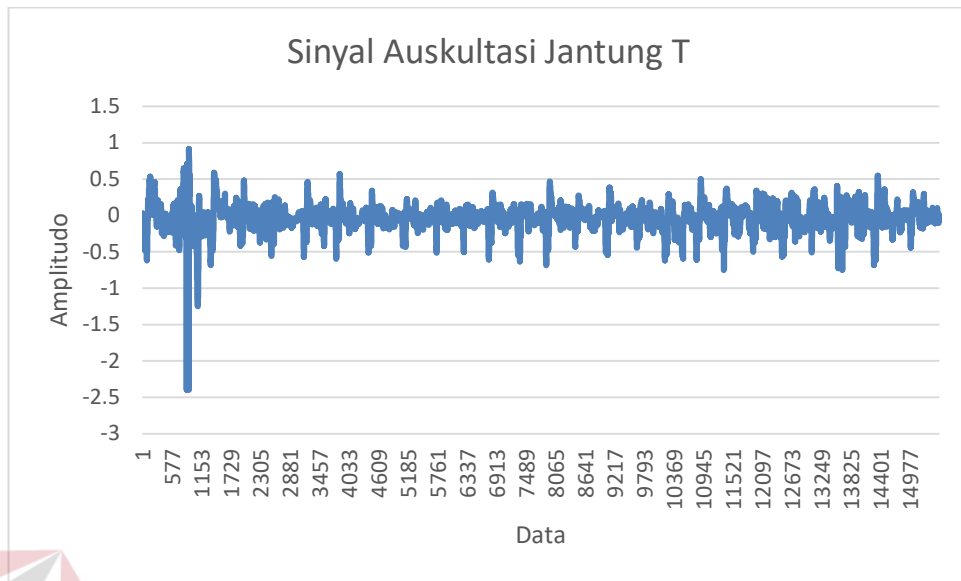
Dari Gambar 4.26 dan Gambar 4.27 didapatkan hasil perhitungan paket *loss* dan *delay* seperti Tabel 4.11 dibawah ini.

Tabel 4.11 Hasil Percobaan 3 dengan *Baudrate* 57600

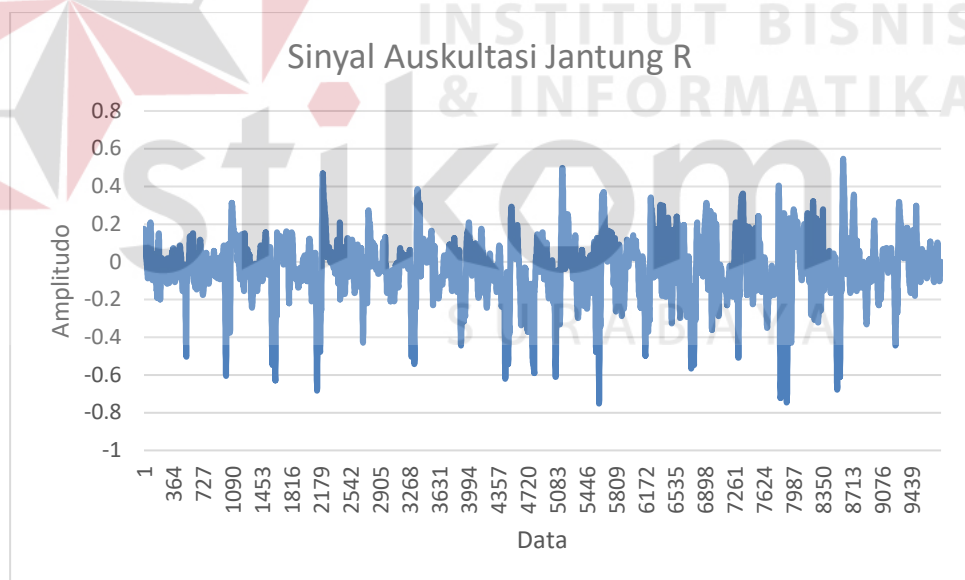
| <i>Delay</i> | <i>Packet loss</i> |
|--------------|--------------------|
| 0,047 s      | 19,58 %            |

d. Percobaan 4 tanpa algoritma

Percobaan dilakukan dengan waktu 30 detik, dengan *interval sampling* 1ms seperti yang terlihat pada Gambar 4.28 dan Gambar 4.29.



Gambar 4.28 Percobaan 4 Data pada Sisi *Transmitter* dengan *Baudrate* 57600



Gambar 4.29 Percobaan 4 Data pada Sisi *Receiver* dengan *Baudrate* 57600

Dari Gambar 4.28 dan Gambar 4.29 didapatkan hasil perhitungan paket *loss* dan *delay* seperti Tabel 4.12 dibawah ini.

Tabel 4.12 Hasil Percobaan 4 dengan *Baudrate* 57600

| <i>Delay</i> | <i>Packet loss</i> |
|--------------|--------------------|
| 5,751 s      | 36,98 %            |

Dari percobaan transmisi sinyal auskultasi jantung dengan menggunakan *baudrate* 57600 pada *interval sampling* 2ms dan 1ms dalam 4 kali percobaan dapat dilihat pada Tabel 4.13 dan Tabel 4.14.

Tabel 4.13 Hasil *Delay* Pengiriman dan juga *Packet Loss* pada *Baudrate* 57600 dengan *Interval Sampling* 2ms

| <b>Teknik transmisi sinyal auskultasi jantung</b> | <i>Packet loss</i> | <i>Delay</i> |
|---|--------------------|--------------|
| Dengan algoritma                                  | 18,85 %            | 0,086 s      |
| Tanpa algoritma                                   | 31 %               | 5,536 s      |

Tabel 4.14 Hasil *Delay* Pengiriman dan juga *Packet Loss* pada *Baudrate* 57600 dengan *Interval Sampling* 1ms

| <b>Teknik transmisi sinyal auskultasi jantung</b> | <i>Packet loss</i> | <i>Delay</i> |
|---|--------------------|--------------|
| Dengan algoritma                                  | 19,58 %            | 0,047 s      |
| Tanpa algoritma                                   | 36,98 %            | 5,751 s      |

Dari data pada Tabel 4.13 dan Tabel 4.14 diketahui bahwa hasil percobaan dengan menggunakan *baudrate* 57600 dengan *baudrate* 115200 berbeda. Pada percobaan yang dilakukan dengan algoritma pada *baudrate* 57600 dan *interval sampling* 2ms *packet loss* yang dihasilkan sebesar 18,85%

dan *delay* 0,086s. Sedangkan dengan *interval sampling* 1ms *packet loss* yang dihasilkan hanya 19,58% dan *delay* 0,047s. Mengacu pada Tabel 3.1 maka jumlah *packet loss* untuk sistem dengan algoritma menghasilkan kategori “sedang” untuk *interval sampling* 1ms dan 2ms. Sedangkan untuk sistem tanpa algoritma jumlah *packet loss* dikategorikan “jelek” untuk *interval sampling* 1ms dan 2ms.

Pengamatan terhadap Tabel 4.13 dan Tabel 4.14 menunjukkan bahwa *packet loss* dan *delay* lebih kecil untuk pengiriman data melalui Zigbee dengan algoritma yang diusulkan dibandingkan dengan pengiriman data tanpa menggunakan algoritma. Hal ini juga berlaku untuk *interval sampling* yang berbeda. Begitu juga untuk baudrate yang berbeda menunjukkan bahwa *packet loss* dan *delay* untuk baudrate 57600 mendapat hasil yang lebih bagus dibandingkan dengan baudrate 115200.

#### 4.9 Hasil Analisa Keseluruhan Sistem

Setelah melakukan percobaan – percobaan dengan beberapa variasi teknik pengambilan data dan juga *interval sampling* yang berbeda maka dapat disimpulkan bahwa proses transmisi dengan adanya pengolahan data sinyal auskultasi jantung mendapatkan hasil yang lebih baik dibanding tanpa dilakukan pengolahan.

Dengan adanya pengolahan data sangat berpengaruh terhadap data yang ditransmisikan dan juga terhadap besar kapasitas transmisi yang digunakan. Pengolahan ini dilakukan agar aliran data dapat dikendalikan

untuk mencukupi besar *buffer* Zigbee. Sehingga data sinyanya auskultasi jantung dapat ditransmisikan dengan baik walaupun dalam penelitian ini masih terdapat *packet loss*. Pada penelitian ini *packet loss* yang dihasilkan masih begitu kecil ketika dibandingkan dengan proses transmisi yang dilakukan tanpa adanya pengolahan data seperti pada penelitian sebelum-sebelumnya.

Pengaruh lain mengenai *interval sampling* yang digunakan. Ketika *interval sampling* 2ms proses transmisi dengan algoritma atau tanpa algoritma dapat berjalan baik tanpa terputus-putus. Sedangkan dengan menggunakan *delay* 1ms proses transmisi tanpa algoritma berjalan secara terputus-putus. Hal ini disebabkan oleh kecepatan pembacaan sensor dengan kecepatan pengiriman tidak seimbang, maka mengakibatkan proses transmisi jadi terputus-putus. Semakin kecil *interval sampling* akan mempengaruhi *delay* dan besar *packet loss* yang terjadi, begitu juga dengan *baudrate* yang digunakan. Seperti yang terlihat pada Tabel 4.15 dan Tabel 4.16.

Tabel 4.15 Hasil Perbandingan antara Menggunakan Algoritma dan Tanpa Algoritma dengan *Interval Sampling* 1ms dan 2ms pada *Baudrate* 115200

| Parameter Perbandingan | 1 ms      |                 | 2 ms      |                 |
|------------------------|-----------|-----------------|-----------|-----------------|
|                        | Algoritma | Tanpa Algoritma | Algoritma | Tanpa Algoritma |
| <i>Packet Loss (%)</i> | 22,58     | 53,87           | 20,73     | 40,04           |
| <i>Delay (s)</i>       | 0,063     | 5,786           | 0,102     | 13,348          |

Tabel 4.16 Hasil Perbandingan antara Menggunakan Algoritma dan Tanpa Algoritma dengan *Interval Sampling* 1ms dan 2ms pada *Baudrate* 57600

| Parameter Perbandingan | 1 ms      |                 | 2 ms      |                 |
|------------------------|-----------|-----------------|-----------|-----------------|
|                        | Algoritma | Tanpa Algoritma | Algoritma | Tanpa Algoritma |
| <i>Packet Loss (%)</i> | 19,58     | 36,98           | 18,85     | 31              |
| <i>Delay (s)</i>       | 0,047     | 5,751           | 0,086     | 5,536           |

