

BAB V

PEMBAHASAN

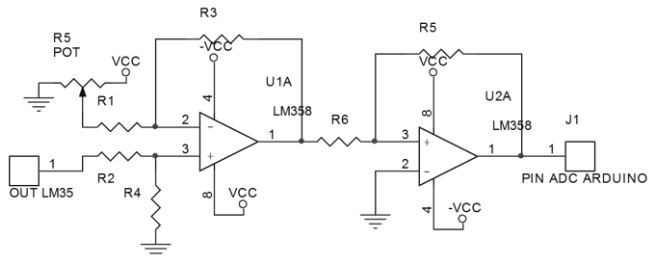
5.1 Rangkaian dan Program

5.1.1 Rangkaian dan Program Suhu

Spesifikasi rangkaian *Differential Amplifier* dan penguatan *Non-inverting Amplifier* adalah:

1. Membutuhkan tegangan *input* tegangan 8VDC
2. Menggunakan *Op-amp* LM358

Sehingga didapatkan rangkaian seperti gambar di bawah ini :



Gambar 5.1 Rangkaian *Differential Amplifier* dan *Non-Inverting Amplifier*

Cara Kerja Rangkaian:

Modul mendapatkan tegangan 8V, sehingga membuat rangkaian *Differential Amplifier* dan *Non-inverting Amplifier*. *Output* dari sensor LM35 merupakan tegangan analog. *Output* dari LM35 masuk ke rangkaian

Differential Amplifier untuk *offset* tegangan sebesar 0.25 V lalu *output* dari *Differential Amplifier* masuk pada pin *Non-inverting Amplifier* LM358 untuk dikuatkan nilai tegangannya.

Penguatan dilakukan sebanyak 10,1 kali. Berikut adalah rumusan penguatan dari rangkaian *Non-inverting Amplifier*:

$$Acl = \frac{Rf}{Rin} + 1$$

$$Acl = \frac{R5}{R6} + 1$$

$$Acl = \frac{9k1}{1k} + 1$$

$$Acl = 10,1 \text{ kali}$$

Proses konversi dari tegangan menjadi data yang ditampilkan:

$$\begin{aligned} \text{Tegangan ADC} &= \frac{\text{Tegangan referensi}}{\text{Banyaknya data}} \\ &= \frac{2,56 \text{ V}}{1024} = 0,0025 \text{ V} \end{aligned}$$

Suhu pada LM35 (contoh) = 26.025°C

- Tegangan LM35 pada suhu tersebut
= 0.26025 V.....(a)
- Tegangan *offset* pada *Differential Amplifier*
= 0.25 V.....(b)

- *Output* rangkaian *Differential Amplifier* (a-b)
= 0.01025 V
- *Output* rangkaian *Differential Amplifier* kemudian masuk ke rangkaian *Non-inverting Amplifier* dengan penguatan 10.1 kali.
- *Output* rangkaian *Non-inverting Amplifier*
= 10.1 x 0.01025
= 0.103525 V
- Data yang masuk arduino
= 0.103525 V : tegangan ADC
= 0.103525 : 0.0025
= 41.42 \approx 41 data
- Data yang ditampilkan *display*
= 25°C + (41 x resolusi)
= 25°C + (41 x 0.025)
= 25°C + 1.025°C
= 26.025°C

Program pada Arduino:

```
float T1,T2,T3,T4,T5,TM;
int ADC0,ADC1,ADC2,ADC3,ADC4,ADC5;
void setup() {
  analogReference(INTERNAL2V56);
}
void loop() {
```

```

//PIN ADC
ADC0=analogRead(A0);
ADC1=analogRead(A1);
ADC2=analogRead(A2);
ADC3=analogRead(A3);
ADC4=analogRead(A4);
ADC5=analogRead(A5);
//ADC MAP
ADC0=map(ADC0,0,1023,0,1033);
ADC1=map(ADC1,0,1023,0,1033);
ADC2=map(ADC2,0,1023,0,1033);
ADC3=map(ADC3,0,1023,0,1033);
ADC4=map(ADC4,0,1023,0,1033);
ADC5=map(ADC5,0,1023,0,1033);
//RUMUS
if (digitalRead(52)==LOW){
  T1=25-1.125+(ADC0*0.025);}
else { T1=0;}
if (digitalRead(50)==LOW){
  T2=25-0.175+(ADC1*0.025);}
else { T2=0;}
if (digitalRead(48)==LOW){
  T3=25+(ADC2*0.025);}
else { T3=0;}
if (digitalRead(46)==LOW){
  T4=25-0.175+(ADC3*0.025);}
else { T4=0;}
if (digitalRead(44)==LOW){
  T5=25+0.3+(ADC4*0.025);}
else { T5=0;}
if (digitalRead(42)==LOW){
  TM=25-0.15+(ADC5*0.025);}
else { TM=0;} }

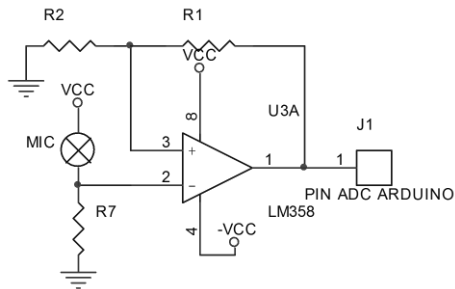
```

5.1.2 Rangkaian dan Program Kebisingan

Spesifikasi rangkaian penguatan *inverting* adalah:

1. Membutuhkan tegangan *input* tegangan 8VDC
2. Menggunakan *Op-amp* LM358

Sehingga didapatkan rangkaian seperti gambar di bawah ini :



Gambar 5.2 Rangkaian *Inverting Amplifier*

Cara Kerja Rangkaian:

Modul mendapatkan tegangan 8V, sehingga membuat rangkaian penguat aktif. *Output* dari *Mic Condenser* berupa tegangan analog masuk ke rangkaian *Inverting Amplifier* untuk dikuatkan nilai tegangannya.

Penguatan dilakukan sebanyak 2200 kali. Berikut adalah rumusan penguatan dari rangkaian *Inverting Amplifier*:

$$Acl = -\frac{Rf}{Rin}$$

$$Acl = -\frac{R1}{R2}$$

$$Acl = -\frac{220k}{100}$$

$$Acl = -2200 \text{ kali}$$

Proses konversi dari tegangan menjadi data yang ditampilkan:

$$\begin{aligned} \text{Tegangan ADC} &= \frac{\text{Tegangan referensi}}{\text{Banyaknya data}} \\ &= \frac{2,56 \text{ V}}{1024} = 0,0025 \text{ V} \end{aligned}$$

Untuk menentukan rumus konversi desibel penulis mengamati perbandingan tegangan adc pada modul yang ditampilkan pada serial dengan data yang ditampilkan oleh INCU II sehingga didapatkan rumus sebagai berikut:

$$\text{dB} = 39 + (\text{ADC} \times 0.04)$$

$$\text{Kebisingan (contoh)} = 40 \text{ dB}$$

$$40 = 39 + (\text{ADC} \times 0.04)$$

$$0.04\text{ADC} = 1$$

$$\text{ADC} = 25$$

$$\text{Vout Penguat} = 25 \times \text{tegangan ADC}$$

$$\text{Vout Penguat} = 25 \times 0,0025 \text{ V}$$

$$= 0.0625 \text{ V} = 62.5 \text{ mV}$$

$$\begin{aligned}
 V_{out} \text{ Mic Condenser} &= 62.5 \text{ mV/penguatan} \\
 &= 62.5 \text{ mV}/2200 \\
 &= 0.02 \text{ mV}
 \end{aligned}$$

Penulis menggunakan rangkaian *Inverting Amplifier* dikarenakan tegangan yang dikuatkan merupakan tegangan bolak-balik. Arduino hanya perlu membaca tegangan positif, oleh karena itu tegangan keluaran negatif *amplifier* tidak dipergunakan.

Program pada Arduino:

```

float NOISE;
int ADC7;
void setup() {
  Serial.begin(9600);}
void loop() {
  ADC7=analogRead(A7);
  if (digitalRead(3)==HIGH){
    NOISE=39+(ADC7*0.04);}
  else { NOISE=0;} }

```

5.2 Pengiriman *Bluetooth*

Modul *Bluetooth* HC05 dipasang pada modul TA dengan menyambungkan pin rx HC-05 ke tx Arduino dan pin tx HC05 ke rx Arduino serta tersambung dengan +5V dan dipastikan *bluetooth* pada *android* sudah aktif. Kemudian melakukan *pairing* atau menyambungkan

antara *bluetooth* HC-05 dan *bluetooth* yang ada pada *android*.

Agar modul *bluetooth* dapat bekerja diperlukan perintah untuk melakukan konfigurasi pin rx dan tx pada arduino seperti berikut :

```
SoftwareSerial bt(15, 14); // pin rx dan tx arduino
```

Selanjutnya perintah untuk memulai serial pada *bluetooth* sebagai berikut :

```
bt.begin(9600);
```

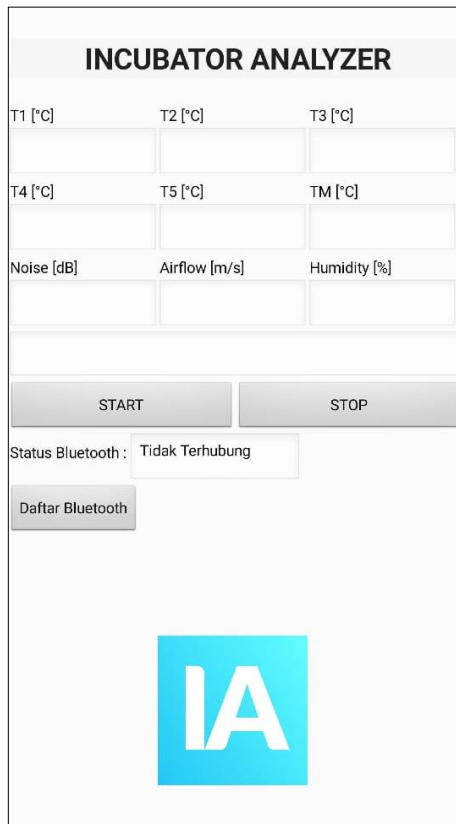
Kemudian untuk pengiriman pada serial dapat dijalankan dengan perintah :

```
bt.print();
```


Program Pengiriman Data pada Arduino:

```
bt.print(T1,3);  
bt.print("|");  
bt.print(T2,3);  
bt.print("|");  
bt.print(T3,3);  
bt.print("|");  
bt.print(T4,3);  
bt.print("|");  
bt.print(T5,3);  
bt.print("|");  
bt.print(TM,3);  
bt.print("|");  
bt.print(NOISE,2);  
bt.print("|");  
bt.print(FLOW,2);  
bt.print("|");  
bt.println(HUM,1);  
delay(1000);
```

5.3 Tampilan *Android*



Gambar 5.3 Tampilan pada *Android*

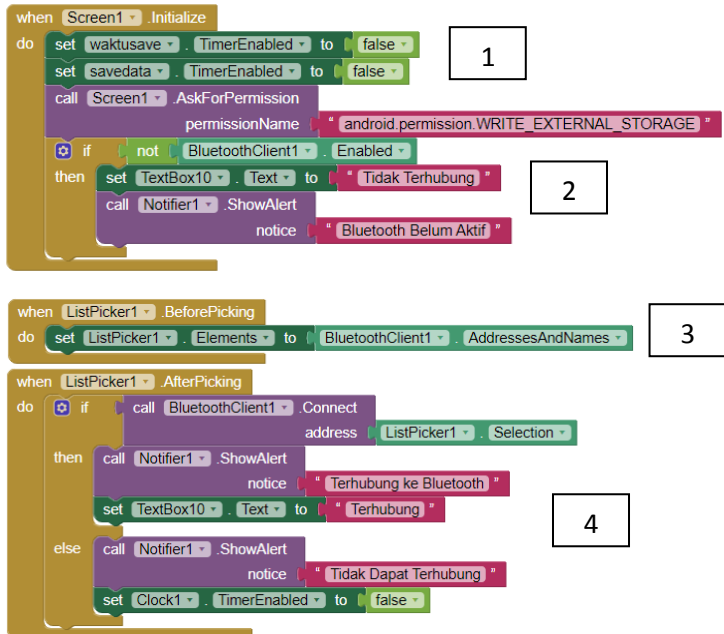
Tampilan pada *android* menggunakan Mit App Inventor dengan pengiriman melalui *bluetooth* HC-05 dari modul.

Langkah-langkah Pengoperasian Alat:

1. Nyalakan *bluetooth* pada *android*
2. Melihat pengaturan atau *setting bluetooth* yang ada pada *android*, lalu cari *bluetooth* dengan nama HC-05
3. Lakukan *pairing bluetooth* dari *android* dengan *bluetooth* dari modul

5.4 Program Android

5.4.1 Bluetooth

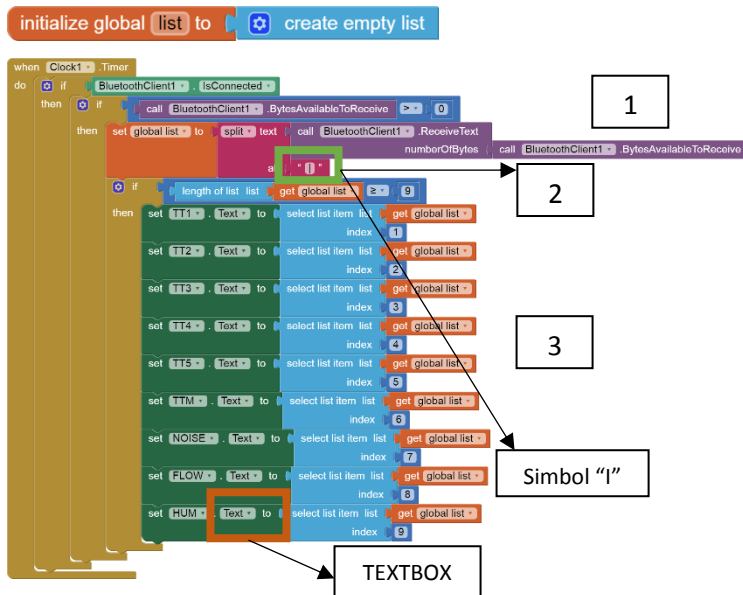


Gambar 5.4 Program untuk *bluetooth*

Perintah diatas merupakan program untuk :

1. Mematikan *timer* sehingga data belum bisa disimpan.
2. Inisialisasi *bluetooth*.
3. Memilih perangkat *bluetooth* yang akan terhubung
4. Memberi notifikasi status koneksi *bluetooth*.

5.4.2 Tampilan



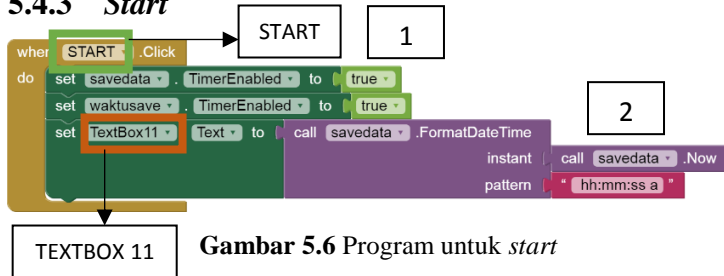
Gambar 5.5 Program untuk tampilan

Perintah diatas merupakan program untuk :

1. Menerima data serial dari modul *bluetooth* alat.
2. Memisahkan data yang diterima dalam bentuk list dan dipisahkan dengan simbol "|".

3. Menampilkan data pada “TEXTBOX” yang telah ditentukan.

5.4.3 Start

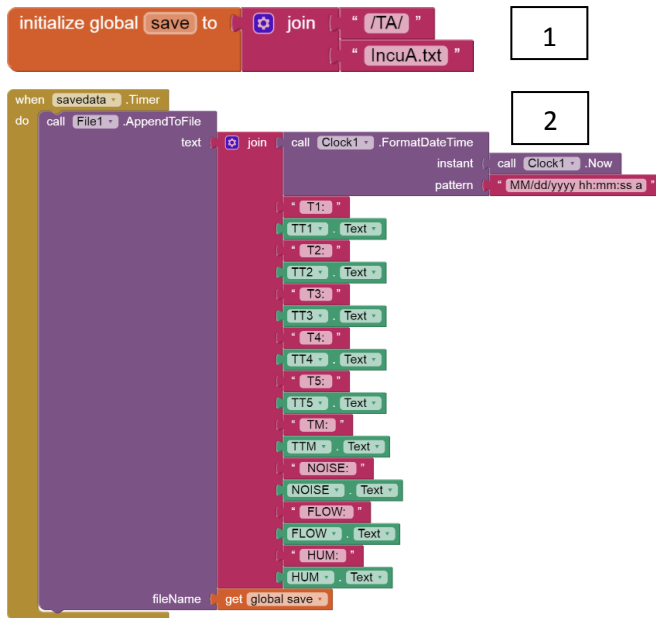


Gambar 5.6 Program untuk *start*

Perintah diatas merupakan program untuk :

1. Memulai *timer* saat tombol START ditekan untuk menyimpan data.
2. Menampilkan waktu pada “TEXTBOX11” ketika tombol START ditekan.

5.4.4 Save

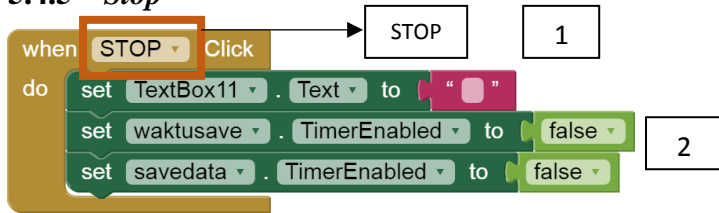


Gambar 5.7 Program untuk save

Perintah diatas merupakan program untuk :

1. Membuat *file* di *folder* “TA” dengan nama *file* “IncuA.txt”.
2. Menambahkan data yang tampil pada tampilan ke dalam *file* setiap *timer* berjalan 30 detik. *Setting* 30 detik diatur di komponen *timer*.

5.4.5 Stop

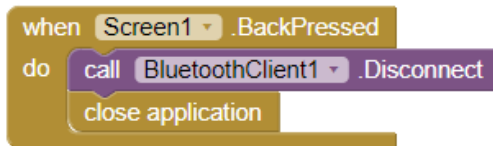


Gambar 5.8 Program untuk *stop*

Perintah diatas merupakan program untuk :

1. Menghapus tampilan “TEXTBOX11” ketika tombol STOP ditekan
2. Menghentikan *timer* penyimpanan sehingga data akan berhenti disimpan.

5.4.6 Back



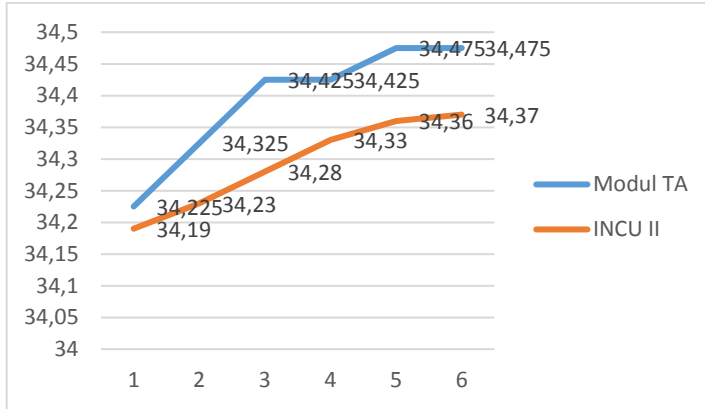
Gambar 5.9 Program untuk *back*

Perintah diatas merupakan program untuk memutus koneksi *bluetooth* dan menutup aplikasi saat tombol BACK pada *android* ditekan.

5.5 Hasil Analisis Data

5.5.1 Analisis Suhu pada *Setting* 35°C

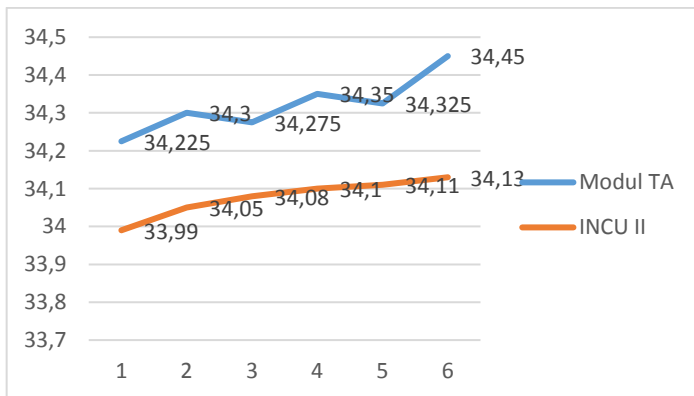
5.5.1.1 Hasil Analisis pada T1



Gambar 5.10 Grafik Pengukuran T1 pada *Setting* 35°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* -0.2867418%

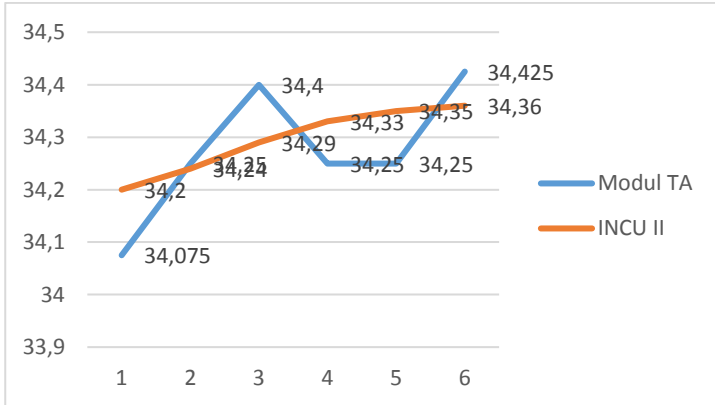
5.5.1.2 Hasil Analisis pada T2



Gambar 5.11 Grafik Pengukuran T2 pada *Setting* 35°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* -0.7165216%.

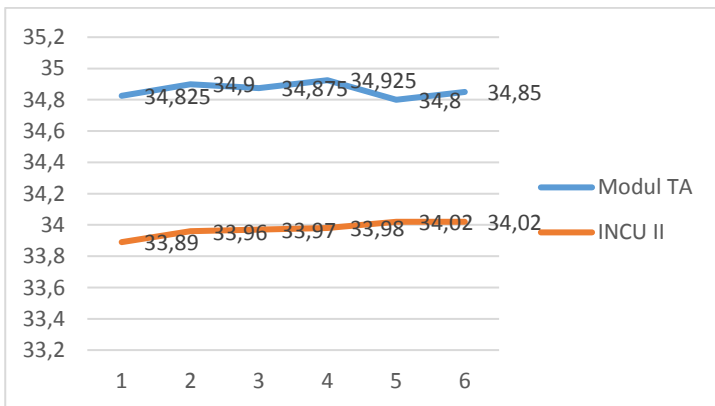
5.5.1.3 Hasil Analisis pada T3



Gambar 5.12 Grafik Pengukuran T3 pada *Setting* 35°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* 0.0583175%.

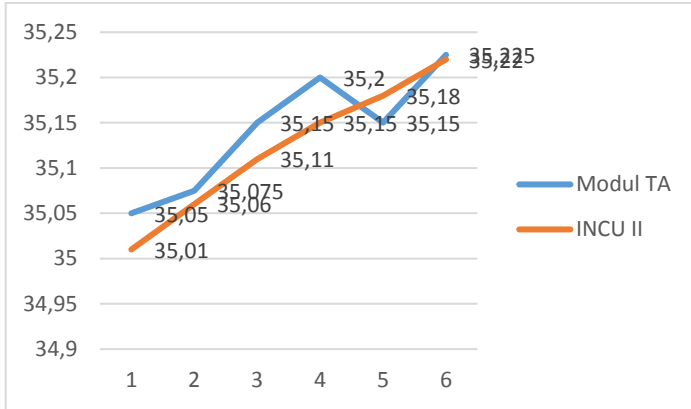
5.5.1.4 Hasil Analisis pada T4



Gambar 5.13 Grafik Pengukuran T4 pada *Setting* 35°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* -2.6172488%.

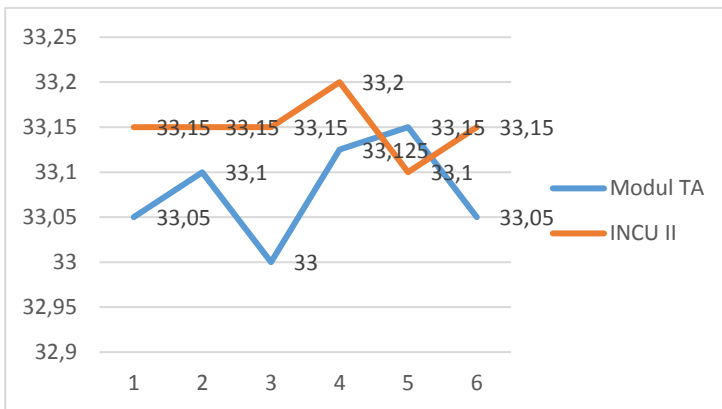
5.5.1.5 Hasil Analisis pada T5



Gambar 5.14 Grafik Pengukuran T5 pada *Setting* 35°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* -0.0569449%.

5.5.1.6 Hasil Analisis pada TM

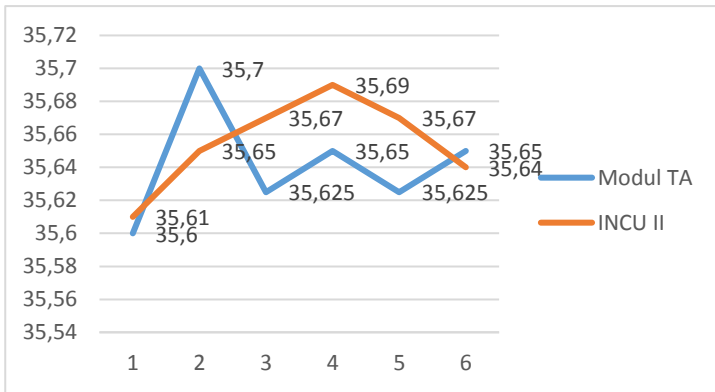


Gambar 5.15 Grafik Pengukuran TM pada *Setting* 35°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* 0.2136752%.

5.5.2 Analisis Suhu pada *Setting* 36°C

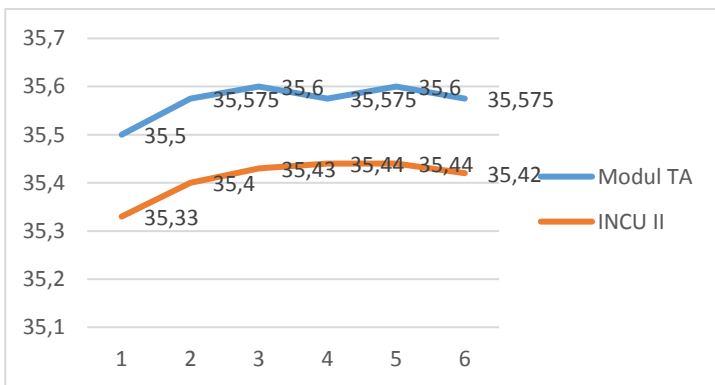
5.5.2.1 Hasil Analisis pada T1



Gambar 5.16 Grafik Pengukuran T1 pada *Setting* 36°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* 0.0373954%.

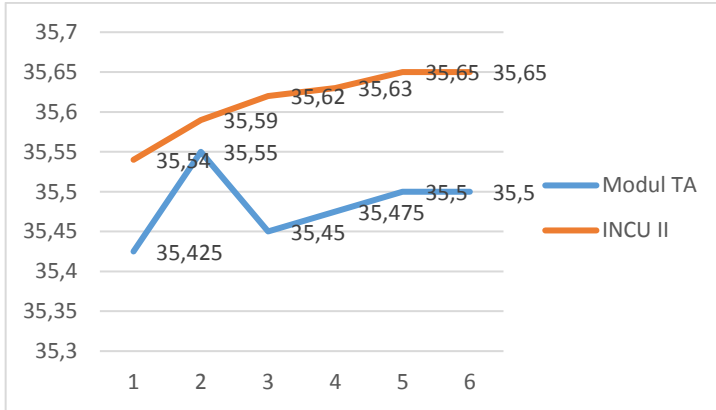
5.5.2.2 Hasil Analisis pada T2



Gambar 5.17 Grafik Pengukuran T2 pada *Setting* 36°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* -0.4542031%.

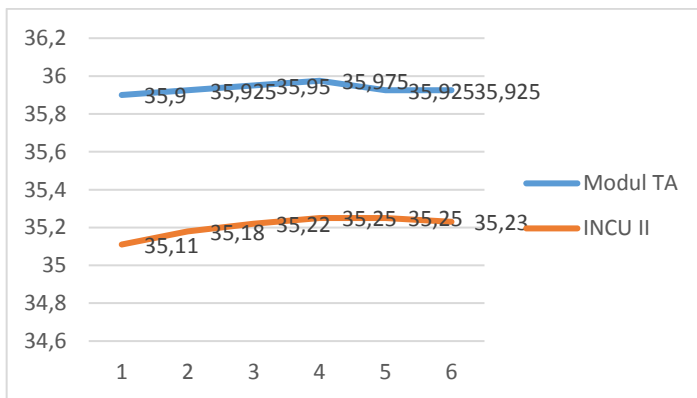
5.5.2.3 Hasil Analisis pada T3



Gambar 5.18 Grafik Pengukuran T3 pada *Setting* 36°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* 0.3650318%.

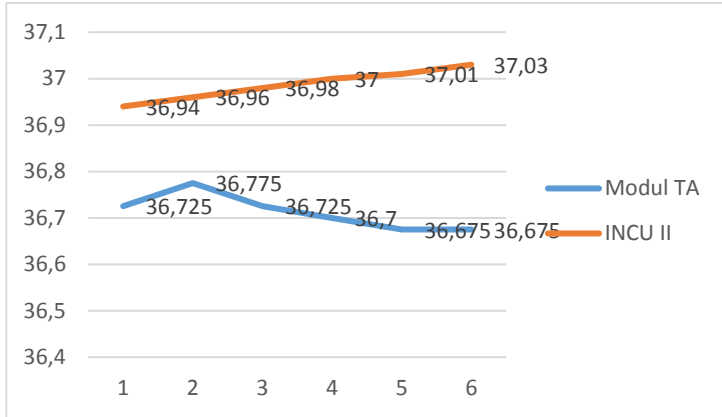
5.5.2.4 Hasil Analisis pada T4



Gambar 5.19 Grafik Pengukuran T4 pada *Setting* 36°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* -2.064003%.

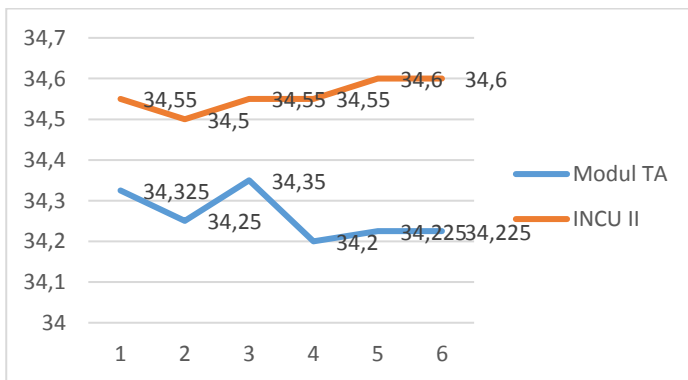
5.5.2.5 Hasil Analisis pada T5



Gambar 5.20 Grafik Pengukuran T5 pada *Setting* 36°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* 0.7412581%.

5.5.2.6 Hasil Analisis pada TM

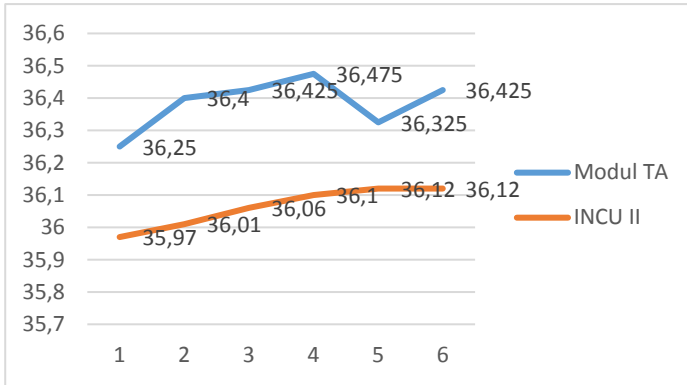


Gambar 5.21 Grafik Pengukuran TM pada *Setting* 36°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* 0.8560405%.

5.5.3 Analisis Suhu pada *Setting* 37°C

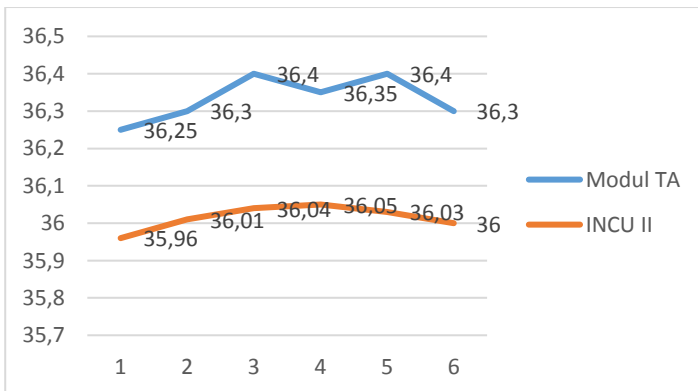
5.5.3.1 Hasil Analisis pada T1



Gambar 5.22 Grafik Pengukuran T1 pada *Setting* 37°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* -0.8873278%

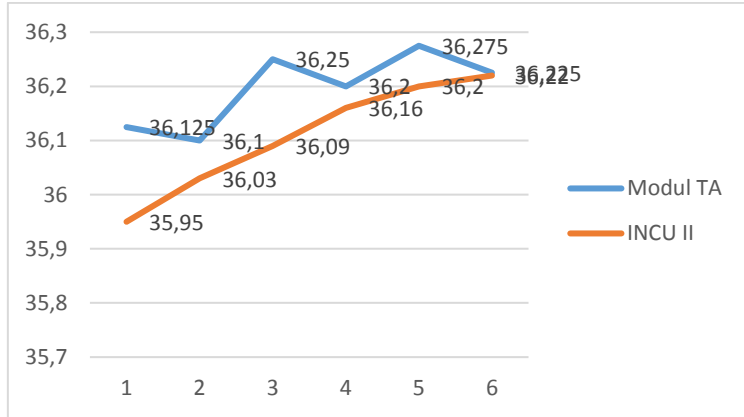
5.5.3.2 Hasil Analisis pada T2



Gambar 5.23 Grafik Pengukuran T2 pada *Setting* 37°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* -0.883891%

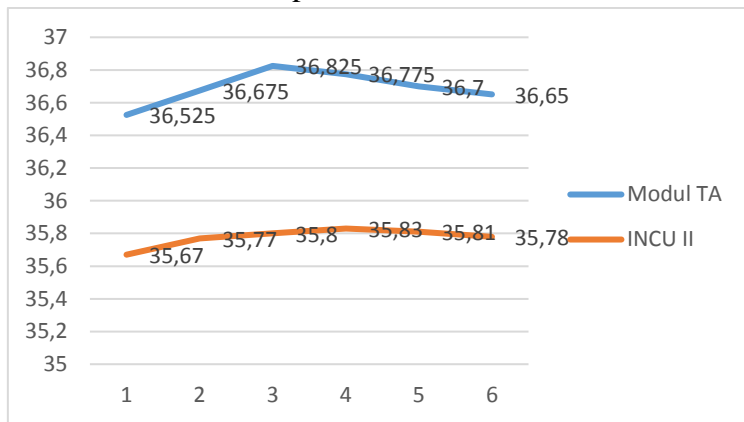
5.5.3.3 Hasil Analisis pada T3



Gambar 5.24 Grafik Pengukuran T3 pada *Setting* 37°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* -0.2423263%

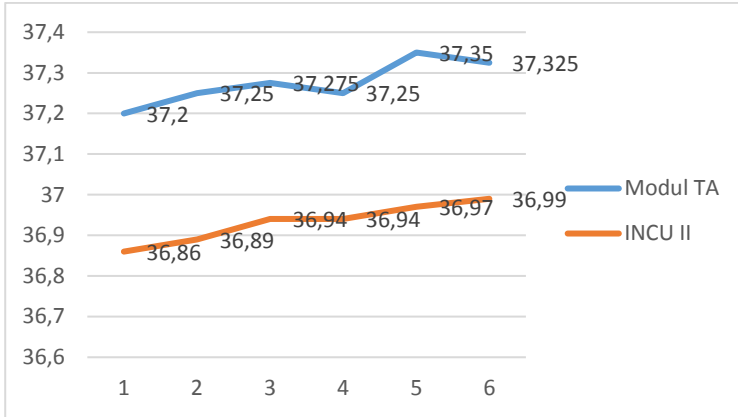
5.5.3.4 Hasil Analisis pada T4



Gambar 5.25 Grafik Pengukuran T4 pada *Setting* 37°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* -2.5575328%

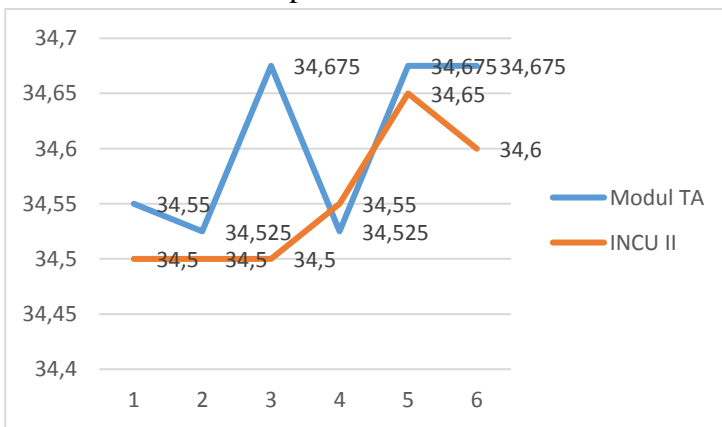
5.5.3.5 Hasil Analisis pada T5



Gambar 5.26 Grafik Pengukuran T5 pada *Setting* 37°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* -0.9296448%.

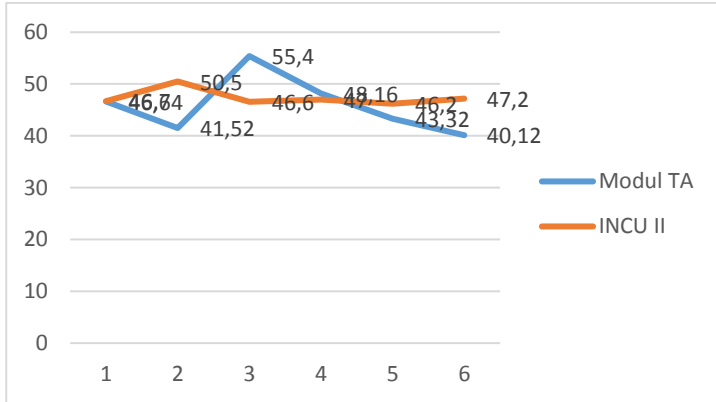
5.5.3.6 Hasil Analisis pada TM



Gambar 5.27 Grafik Pengukuran TM pada *Setting* 37°C

Sensor LM35 mendapatkan hasil pembacaan suhu yang mendekati INCU dengan *error* -0.1567776%.

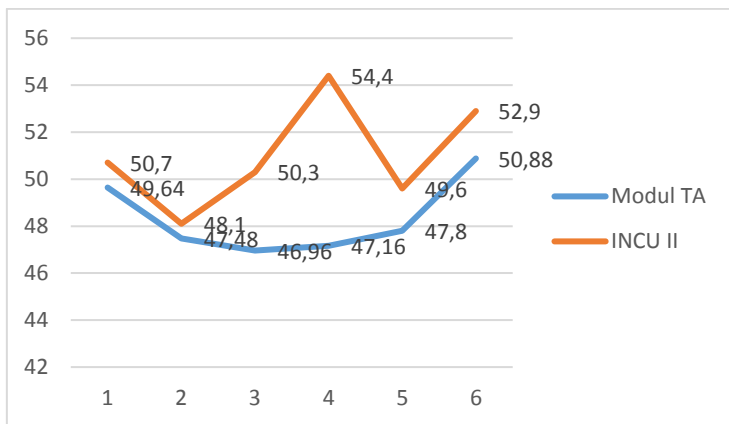
5.5.4 Analisis Kebisingan pada *Setting* 35°C



Gambar 5.28 Grafik Pengukuran Kebisingan pada *Setting* 35°C

Sensor kebisingan mendapatkan hasil yang mendekati INCU II dengan *error* 3.1808586%.

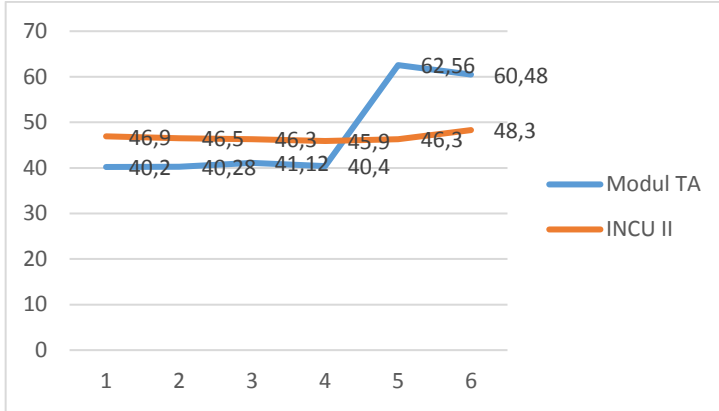
5.5.5 Analisis Kebisingan pada *Setting* 36°C



Gambar 5.29 Grafik Pengukuran Kebisingan pada *Setting* 36°C

Sensor kebisingan mendapatkan hasil yang mendekati INCU II dengan *error* 5.254902%.

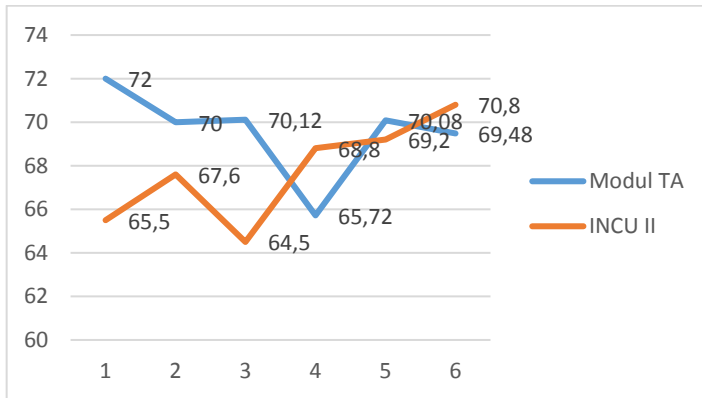
5.5.6 Analisis Kebisingan pada *Setting* 37°C



Gambar 5.30 Grafik Pengukuran Kebisingan pada *Setting* 37°C

Sensor kebisingan mendapatkan hasil yang mendekati INCU II dengan *error* -1.7273376%.

5.5.7 Analisis Kebisingan saat Diberi Perlakuan



Gambar 5.31 Grafik Pengukuran Kebisingan saat Diberi Perlakuan

Sensor kebisingan mendapatkan hasil yang mendekati INCU II dengan *error* -2.7066929%.

5.6 Kinerja Sistem Keseluruhan

Dari penelitian ini dapat diketahui bahwa modul ini dapat digunakan untuk melakukan pengukuran parameter pada *baby incubator* dilengkapi tampilan *android* dan penyimpanan data. Modul ini terdiri dari sensor LM35 yang digunakan untuk mendeteksi suhu di dalam ruang *baby incubator* dan *Mic Condenser* yang digunakan mendeteksi kebisingan di dalam ruang *baby incubator*. *Output* dari sensor LM35 merupakan tegangan analog. *Output* dari LM35 masuk ke rangkaian *Differential Amplifier* untuk *offset* tegangan sebesar 0.25 V lalu *output* dari *Differential Amplifier* masuk pada pin *Non-Inverting Amplifier* LM358 untuk dikuatkan nilai tegangannya. Penguatan dilakukan sebanyak 10,1 kali. Keluaran dari penguat *Non-Inverting* kemudian masuk ke pin ADC arduino untuk dilakukan kalkulasi pada program untuk mengubah nilai tegangan menjadi suhu untuk kemudian ditampilkan pada *display*. Untuk parameter kebisingan yang menggunakan *Mic Condenser*, *output* dari *Mic Condenser* berupa tegangan analog masuk ke rangkaian *Inverting Amplifier* LM358 untuk dikuatkan nilai

tegangannya. Penguatan dilakukan sebanyak 2200 kali. Penulis menggunakan rangkaian *Inverting Amplifier* dikarenakan tegangan yang dikuatkan merupakan tegangan bolak-balik. Arduino hanya perlu membaca tegangan positif, oleh karena itu tegangan keluaran negatif amplifier tidak dipergunakan. Keluaran dari penguat *Inverting Amplifier* kemudian masuk ke pin ADC arduino untuk dilakukan kalkulasi pada program untuk mengubah nilai tegangan menjadi level kebisingan dengan satuan dB untuk kemudian ditampilkan pada *display*.

Modul ini juga dilengkapi pengiriman data melalui *bluetooth*. Arduino melakukan konfigurasi agar semua data parameter dapat dikirimkan. Semua data parameter akan diolah dalam bentuk *list* kemudian dikirim melalui serial data.

Data parameter yang dikirimkan oleh *bluetooth* dapat ditampilkan pada aplikasi *android* dengan menyambungkan antara *bluetooth* HC-05 dan *bluetooth* yang ada pada *android*. Pada aplikasi dilakukan pemilihan perangkat *bluetooth* dengan cara menekan tombol daftar *bluetooth* dan memilih perangkat HC-05. Data semua parameter akan secara *real time* ditampilkan pada kolom sesuai dengan nama parameter. Untuk mulai menyimpan

data *user* dapat menekan tombol *start*. Data akan disimpan setiap 30 detik. Ketika *user* merasa data yang didapat sudah cukup maka *user* dapat menekan tombol *stop* untuk menyimpan data. Data akan tersimpan di penyimpanan internal pada *folder* TA dengan nama *file* “*incuA.txt*”.

Pemrograman pada aplikasi *android* dibuat menggunakan MIT App Inventor. Data *list* yang dikirim dari serial data *bluetooth* akan dipisah dan ditampilkan pada kolom parameter. Pada program digunakan 2 *timer*.

Timer pertama setiap satu detik untuk menampilkan data parameter dan *timer* setiap 30 detik untuk melakukan penyimpanan.

Modul kemudian dibandingkan dengan alat pembanding INCU II dan didapatkan hasil pada parameter suhu nilai *error* suhu tekecil sebesar 0.0373954% pada T1 suhu *setting* 36°C dan *error* terbesar sebesar 2.6172488% pada T4 suhu *setting* 35°C. Sedangkan pada parameter kebisingan didapat nilai *error* tekecil sebesar - 1.7273376% pada *setting* 37°C dan *error* terbesar sebesar 5.254902% pada *setting* 36°C.

Berdasarkan hasil penelitian ini, ada beberapa keterbatasan yang dialami penulis. Keterbatasan ini dapat lebih diperhatikan untuk penelitian selanjutnya.

Keterbatasan dalam penelitian ini yaitu penyimpanan hanya dapat dilakukan melalui aplikasi pada *android*. Pengolahan data hasil pengukuran masih dilakukan secara manual dengan cara mengimpor data penyimpanan ke excel. Tidak ada tampilan grafik untuk mengetahui saat keadaan stabil.