

## **BAB IV**

### **HASIL PENGUJIAN DAN PENGAMATAN**

Dalam bab ini penulis akan menguraikan dan menjelaskan beberapa hasil pengujian dari hasil penelitian tugas akhir ini. Pengujian yang dilakukan meliputi pengujian perangkat keras (*hardware*), perangkat lunak (*software*) dan kinerja keseluruhan sistem pada tugas akhir ini.

#### **4.1 Pengujian Xbee**

Pengujian Xbee dilakukan dengan menggunakan program X-CTU. Program X-CTU merupakan open source yang digunakan untuk menkonfigurasi awal Xbee.

##### **4.1.1 Tujuan**

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

##### **4.1.2 Alat yang digunakan**

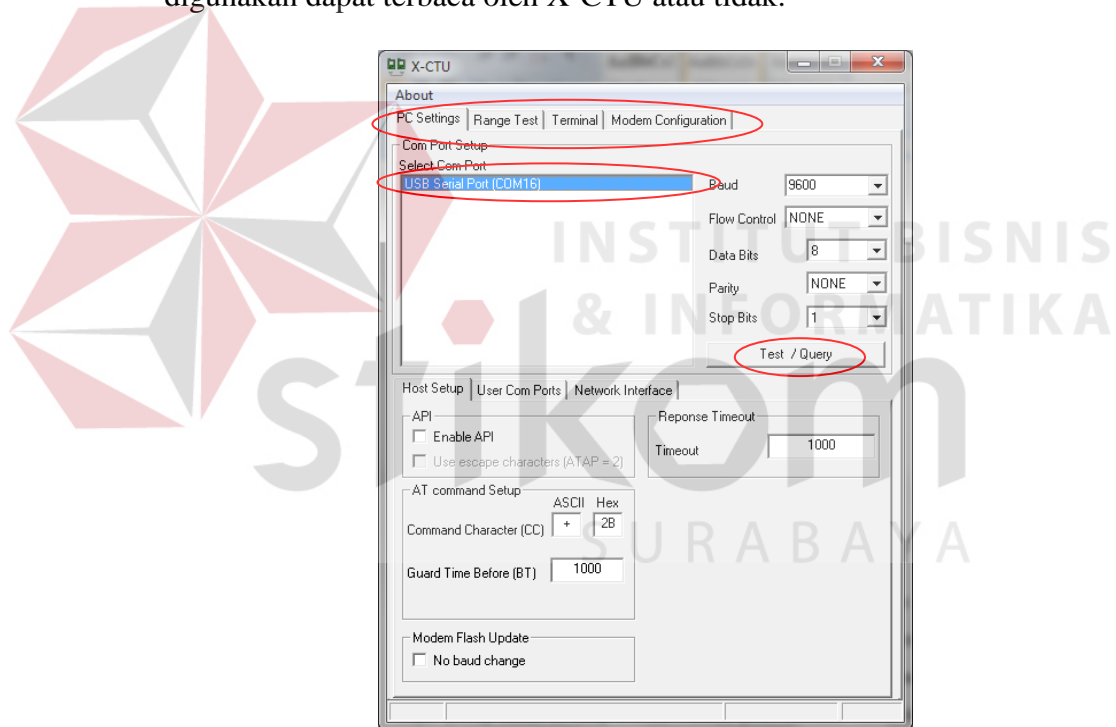
Untuk melakukan percobaan ini maka diperlukan beberapa alat sebagai berikut.

- a. Usb adapter
- b. Xbee adapter
- c. Xbee
- d. Komputer/ laptop
- e. *Software X-CTU*

### 4.1.3 Prosedur Pengujian

Prosedur pengujian alat :

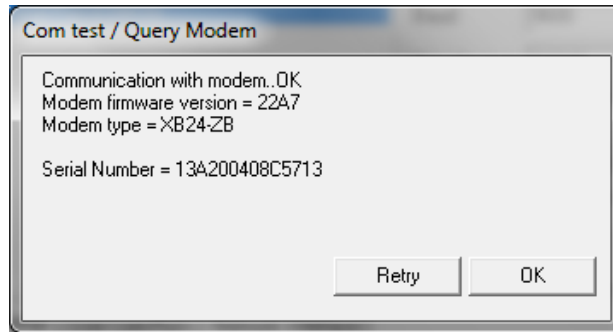
- Hubungkan xbee adapter dengan kabel usb adapter.
- Nyalakan komputer kemudian hubungkan kabel usb adaptornya ke komputer/laptop.
- Buka *software* X-CTU dan tekan tombol “ Test / Query” pada tab “PC Setting”.
- Maka akan muncul dialog yang dapat mengetahui apakah Xbee yang digunakan dapat terbaca oleh X-CTU atau tidak.



**Gambar 4.1** Tampilan *Software* X-CTU

### 4.1.4 Hasil Pengujian

Pada Gambar 4,2 tertulis “*Communication with Modem OK* ” hal ini menandakan bahwa Xbee yang digunakan dapat berkomunikasi dengan X-CTU. Dengan demikian maka Xbee dapat digunakan pada pengerjaan tugas akhir ini.



**Gambar 4.2** Xbee dalam keadaan normal

## **4.2 Pengujian Arduino**

Pengujian arduino dilakukan dengan memasukan skrip program sederhana pada arduino menggunakan aplikasi arduino IDE. Arduino yang baik dapat mengeksekusi program dengan baik.

### **4.2.1 Tujuan**

Pengujian ini dilakukan untuk mengetahui apakah arduino yang digunakan tidak mengalami kerusakan. Sehingga saat arduino digunakan pada sistem dapat membantu sistem berjalan dengan baik.

### **4.2.2 Alat yang digunakan**

Alat yang digunakan untuk melakukan pengujian antara lain :

- a. Kabel usb
- b. Arduino Uno
- c. Komputer/laptop
- d. *Software* Arduino IDE

### **4.2.3 Prosedur Pengujian**

- a. Hubungkan Arduino dengan kabel usb

b. Nyalakan komputer kemudian hubungkan kabel usb tadi dengan komputer.

c. Buka *software* Arduino IDE dan isi perintah dalam bahasa C. Sebagai contoh penulis memasukkan perintah sebagai berikut :

```
void setup()
{
    Serial.begin(9600);
    Serial.println("Cek Mulai :");
}
int i=0;
void loop()
{
    Serial.print("Data ke");
    Serial.println(i);
    delay(1000);
    i++;
}
```

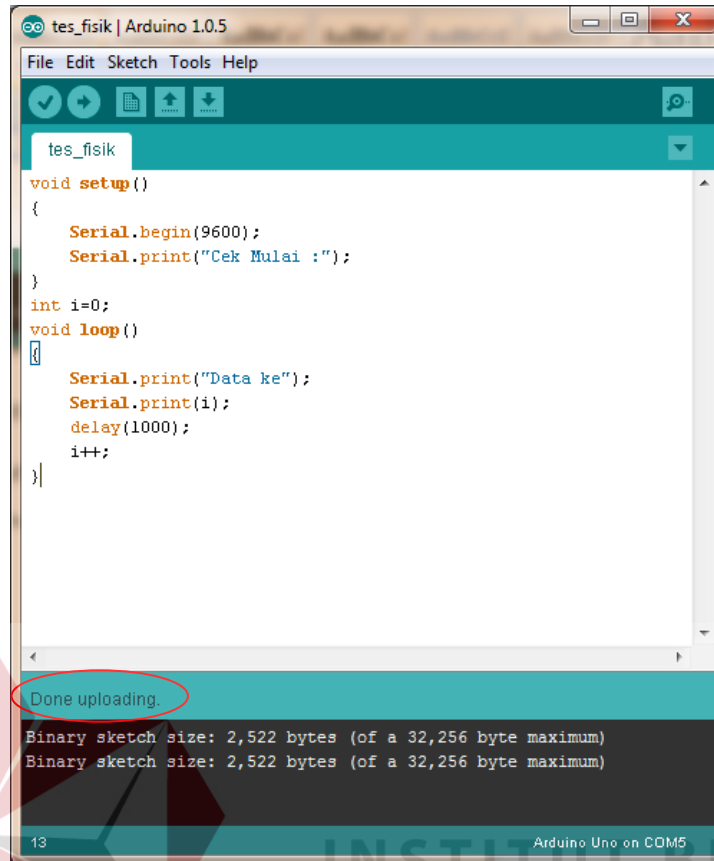
d. Apabila telah selesai untuk mengisi perintah, maka tekan “*Verify*” untuk mengecek apabila terdapat perintah yang salah dalam bahasa C. Dan tekan “*Upload*” untuk memasukkan perintah tersebut ke dalam Arduino Uno.

e. Setelah *program* telah berhasil dimasukkan, maka tekan *icon Serial monitor* pada kanan atas. Maka akan muncul tampilan serial monitor.

f. Setelah window serial monitor muncul, amati kiriman data serial oleh arduino.

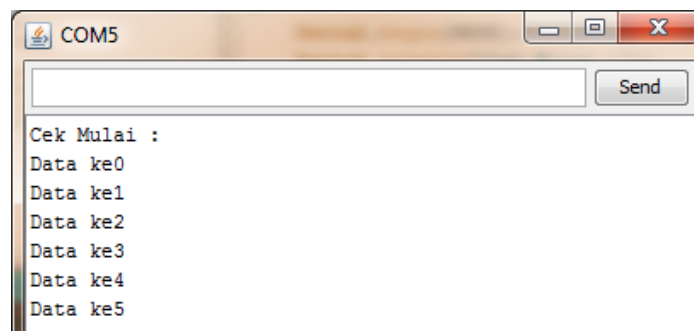
#### 4.2.4 Hasil Pengujian

Hasil dari pengujian pengisian program ke arduino dapat dilihat pada Gambar 4.4. Lingkaran merah menunjukkan bahwa arduino yang digunakan berhasil diisi dengan program yang telah ditulis dalam *software* arduino IDE.



**Gambar 4.3** *upload program berhasil*

Program yang dimasukkan kedalam arduino merupakan program untuk mengirimkan data menggunakan serial. Proses pengiriman ini apabila arduino masih dihubungkan dengan USB PC maka kita dapat menerima data yang dikirim menggunakan menu serial monitor pada *software* arduino IDE. Hasil dari serial monitor dapat dilihat pada Gambar 4.4.



**Gambar 4.4** *Program berhasil berjalan*

Gambar 4.4 menunjukkan bahwa data dikirimkan sesuai dengan perintah program yang telah diisi pada arduino. Dengan begitu arduino ini dapat bekerja dengan baik, dan dapat digunakan untuk sistem.

### **4.3 Pengujian tampilan prediksi**

Pengujian ini merupakan pengujian prediksi pada aplikasi visual basic dengan memberikan data tinggi bertambah secara konstan, bertambah berubah tapi beraturan dan bertambah berubah tidak beraturan.

#### **4.3.1 Tujuan**

Pengujian ini bertujuan untuk mengetahui waktu prediksi yang ditampilkan oleh aplikasi.

#### **4.3.2 Alat yang digunakan**

Alat yang digunakan untuk pengujian sistem ini antara lain:

- a. Arduino Uno
- b. Kabel USB
- c. Komputer/laptop
- d. *Software* Arduino IDE
- e. *Software* Visual Basic

#### **4.3.3 Prosedur pengujian :**

- a. Hubungkan Arduino dan komputer dengan menggunakan kabel USB.
- b. Aktifkan komputer dan buka program Arduino IDE.
- c. Upload skrip yang digunakan untuk pemberian data.
- d. Buka aplikasi monitoring dari visual basic.

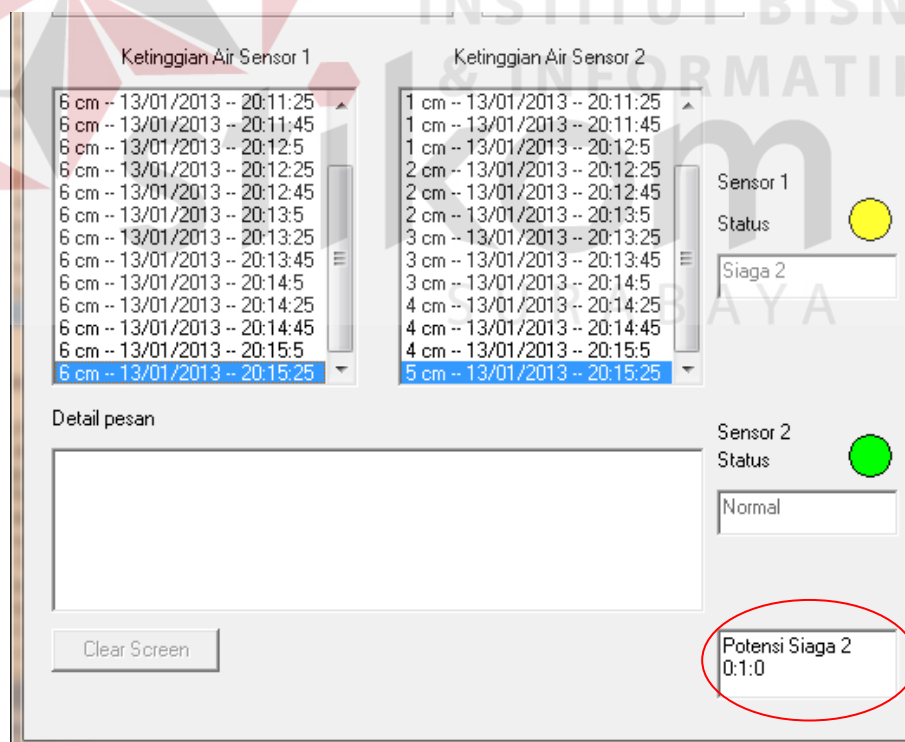
- e. Isi sampel ketinggian, pilih port serial yang terdeteksi dan tekan tombol “Connect”.
- f. Amati data, apakah terdapat data yang tidak diinginkan.

#### 4.3.4 Hasil Pengujian

Dalam aplikasi ini, waktu prediksi potensi siaga selalu diperbaharui setiap perubahan nilai tinggi berdasarkan data ketinggian yang masuk ke komputer.

Dalam percobaan ini menggunakan *interval* waktu pengiriman data setiap 20 detik sekali (sesuai waktu pengiriman *node coordinator* atau data diambil setiap 20 detik). Dalam pengujian, prediksi potensi siaga yang dihitung adalah prediksi waktu untuk mencapai 6 cm.

##### 1. Tinggi bertambah secara teratur



**Gambar 4.5** Percobaan pada pola teratur

Dalam Gambar 4.5 pada sensor 2, nilai tinggi bertambah setiap 3 kali pengiriman data. Jadi waktu yang diharapkan tiap perubahan 1cm adalah 1 menit.

Waktu prediksi potensi siaga 2 yang ditampilkan pada Gambar 4.5 merupakan waktu yang dihitung dari 5 cm menuju 6 cm dengan perhitungan sebagai berikut:

-Besarnya nilai perubahan tinggi = Tinggi sekarang – tinggi sebelumnya = 2 - 1 = 1 cm

-Waktu selang perubahan ketinggian 1 cm = 3 x interval pengiriman = 3 x 20 detik = 60 detik.

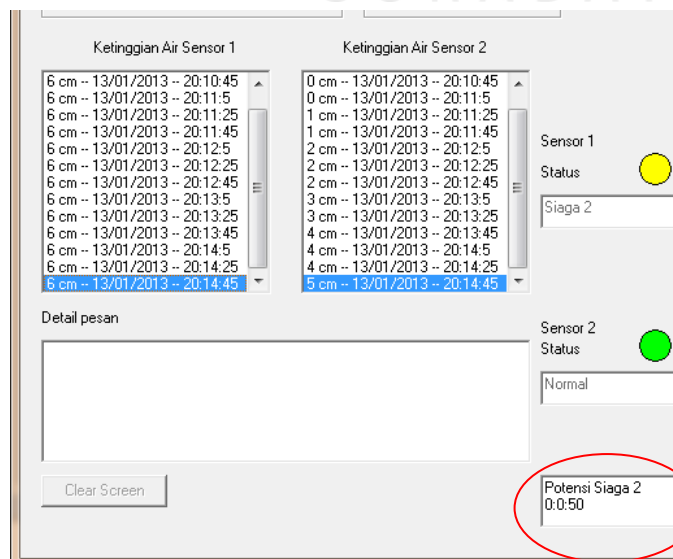
-Jadi kecepatan perubahan sebesar 1 cm/menit

-Karena nilai bertambah secara teratur, maka kecepatan rata-rata yaitu = 1 cm/menit

-Waktu prediksi perubahan tinggi air dari 5 cm ke 6 cm = (batas tinggi – tinggi sekarang)/kecepatan rata-rata = (6 cm – 5 cm)/1cm = 1 menit.

Dari perhitungan diatas, hasil hitung waktu prediksi sama dengan pola pengiriman yang diharapkan (3x pengiriman atau 1 cm per 1 menit).

## 2. Tinggi bertambah dan berubah secara teratur



Gambar 4.6 Percobaan pada pola berubah-ubah dan teratur



Berdasarkan Gambar 4.6 pada sensor 2, nilai tinggi bertambah setiap 3 kali pengiriman data (1 menit) kemudian nilai tinggi bertambah lagi 2 kali pengiriman data (40 detik) dan kembali berubah 3 kali pengiriman dan seterusnya.

Tabel 4.1 berikut adalah tabel data masuk seperti Gambar 4.6 dan hasil perhitungan kecepatan rata-rata perubahan ketinggian:

**Tabel 4.1** Tabel data masuk dengan pola berubah dan teratur

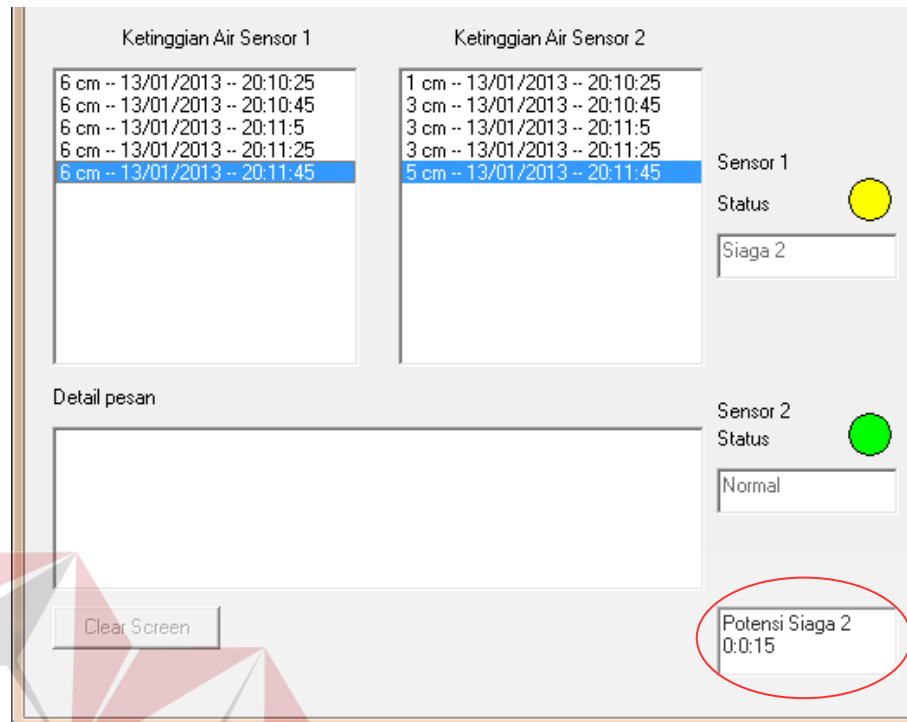
Tinggi (cm)	Tanggal	Waktu (detik)	Waktu	V (Cm/detik)	V <sub>rata-rata</sub> (Cm/detik)
			(n*20) (detik)		
0	13/01/2013	20:10:25	60	-	-
0	13/01/2013	20:10:45			
0	13/01/2013	20:11:05			
1	13/01/2013	20:11:25	40	0.016667	0.016667
1	13/01/2013	20:11:45			
2	13/01/2013	20:12:05	60	0.025	0.020834
2	13/01/2013	20:12:25			
2	13/01/2013	20:12:45			
3	13/01/2013	20:13:05	40	0.016667	0.019446
3	13/01/2013	20:13:25			
4	13/01/2013	20:13:45	60	0.025	0.020834
4	13/01/2013	20:14:05			
4	13/01/2013	20:14:25			
5	13/01/2013	20:14:45	?	0.016667	0.020001

Prediksi yang ditampilkan pada Gambar 4.6 merupakan hasil perhitungan untuk mengisi tanda tanya (?) pada tabel 4.1. Berikut adalah perhitungan dari waktu prediksi di atas:

$$\text{Waktu prediksi} = (\text{batas tinggi} - \text{tinggi sekarang}) / \text{kecepatan rata-rata} = (6 - 5) / 0.02 = 50 \text{ detik}$$

Jadi waktu prediksi potensi siaga 2 ketika tinggi 5cm ke 6 cm dengan kecepatan rata-rata di atas adalah 50 detik.

### 3. Tinggi bertambah tidak beraturan



**Gambar 4.7** Percobaan pada pola berubah tidak beraturan

Berdasarkan Gambar 4.7 pada sensor 2, nilai tinggi bertambah 2 cm (dari 1 cm ke 3 cm) dalam 20 detik kemudian naik 2 cm dalam 1 menit.

Tabel 4.2 berikut adalah tabel data masuk seperti Gambar 4.7 dan hasil perhitungan kecepatan rata-rata perubahan ketinggian:

**Tabel 4.2** Tabel data masuk dengan pola berubah tidak beraturan

Tinggi (cm)	Tanggal	Waktu (detik)	Waktu	V (Cm/detik)	V rata-rata (Cm/detik)
			$t = (n \cdot 20)$ (detik)		
1	13/01/2013	20:10:25	20	0	0
3	13/01/2013	20:10:45	60	0.1	0.1
3	13/01/2013	20:11:5			
3	13/01/2013	20:11:25			
5	13/01/2013	20:11:45	?	0.0334	0.0667

Prediksi yang ditampilkan pada Gambar 4.7 merupakan hasil perhitungan untuk mengisi tanda tanya (?) dari Tabel 4.2 dan berikut perhitungan dari prediksi di atas:

Waktu prediksi = (batas tinggi – tinggi sekarang) / kecepatan rata-rata = (6–5) / 0.0667 = 15 detik.

Jadi waktu prediksi potensi siaga 2 ketika tinggi 5cm ke 6 cm dengan kecepatan rata-rata di atas adalah 15 detik.

Dari percobaan di atas, ketika data tinggi berubah secara drastis atau berubah tidak beraturan, prediksi akan tetap dapat menghitung namun prediksi berubah (*update*) ketika 20 detik sesudahnya (ketika data masuk). Semakin cepat interval waktu pengiriman data, semakin cepat perhitungan prediksi yang dilakukan.

#### **4.4 Pengujian sistem**

Pengujian ini merupakan pengujian secara keseluruhan sistem. Pengujian ini menggunakan perangkat secara lengkap yaitu dengan menggabungkan perangkat xbee, arduino dan komputer.

##### **4.4.1 Tujuan**

Pengujian ini bertujuan untuk

1. Mengetahui bahwa aplikasi dapat menerima/membaca data yang masuk melalui xbee sesuai apa yang diharapkan.
2. Mengetahui apakah aplikasi visual basic dapat mengolah data masuk, menampilkan grafik, memberikan informasi keadaan *hardware*, melakukan perhitungan perkiraan dan memberikan keadaan status ketinggian air saat itu.

##### **4.4.2 Alat yang digunakan**

Alat yang digunakan untuk pengujian sistem ini antara lain:

- a. Xbee
- b. Arduino Uno
- c. Xbee Shield
- d. Kabel USB
- e. Komputer/laptop
- f. *Software* Arduino IDE
- g. *Software* Visual Basic

#### 4.4.3 Prosedur pengujian :

- a. Hubungkan Xbee dengan Xbee Shield.
- b. Hubungkan xbee shield yang telah terhubung dengan xbee tersebut dengan Arduino uno.
- c. Hubungkan Arduino dengan kabel usb dan hubungkan ke komputer.
- d. Buka *software* arduino IDE dan isikan perintah dalam bahasa C. Dalam tugas akhir ini penulis mengisikan perintah untuk mendengarkan atau menerima data khusus dari *node coordinator*. Berikut isi perintah yang diberikan oleh penulis.

```
String indata;
void setup()
{
    Serial.begin(9600);
    indata="";
}
void loop()
{
    while(Serial.available()>0)
    {
        char data_masuk=(char)Serial.read();
        indata+=data_masuk;
        if(data_masuk == '\n')
        {
            pisahData();
            indata="";
        }
    }
}
void pisahData()
```

```
{
  //masukkan data yg ditampung di indata
  if ((indata[0] == '%') && (indata[1] == 'C'))
  {
    // cetak atau kirim
    Serial.print(indata);
  }
}
```

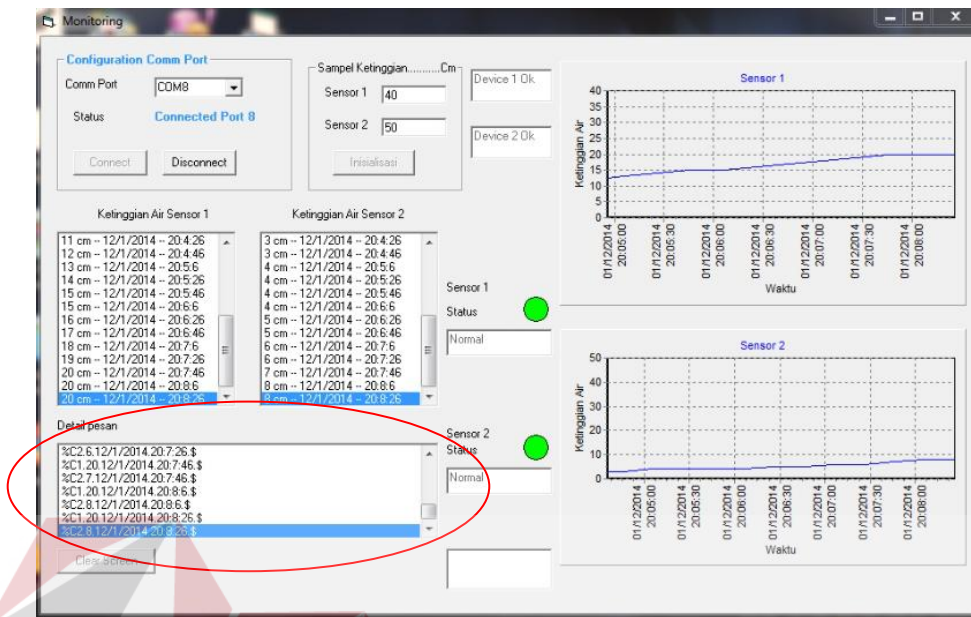
- e. Verify dan upload perintah tersebut dengan menekan tombol upload.
- f. Buka aplikasi monitoring dari visual basic.
- g. Isi sampel ketinggian, pilih port serial yang terdeteksi dan tekan tombol “Connect”.
- h. Amati data, apakah terdapat data yang tidak diinginkan.

#### 4.4.4 Hasil Pengujian

Data yang masuk nantinya pasti memiliki simbol “%C” dikarenakan data yang masuk berasal dari *node coordinator*. Jika data tersebut telah diterima oleh *node end device*, maka *node* ini berhak meneruskannya ke komputer dalam *software* visual basic. Sedangkan jika data tidak mempunyai simbol tersebut maka data akan dibuang.

Dalam aplikasi visual basic ini data yang masuk akan diproses dan ditampilkan untuk memberikan beberapa informasi.

## 1. Proses pembacaan data



**Gambar 4.8** Pengolahan data masuk

Dari gambar 4.8 di atas dapat disimpulkan aplikasi hanya dapat membaca data yang diharapkan yaitu data yang memiliki simbol “%” dan “C” saja dimana data tersebut merupakan data dari *coordinator*.

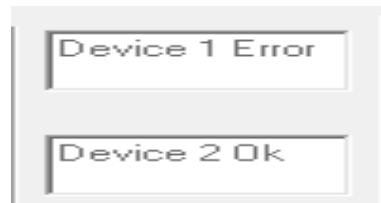
## 2. Tampilan *error checking*

Tampilan ini bertujuan untuk memberi informasi kepada *user* tentang baik tidaknya kondisi *node* sensor/router. Dalam tugas akhir ini *node coordinator* sebagai pemberi pesan bila terdapat *node* sensor/router yang rusak.

Pada dasarnya data masuk dibaca per karakter sampai karakter paling akhir yang telah ditentukan. Pada tugas akhir ini penulis menggunakan karakter paling akhir yaitu “s” (untuk pesan *error*) dan “\$” (untuk pesan tidak *error*).

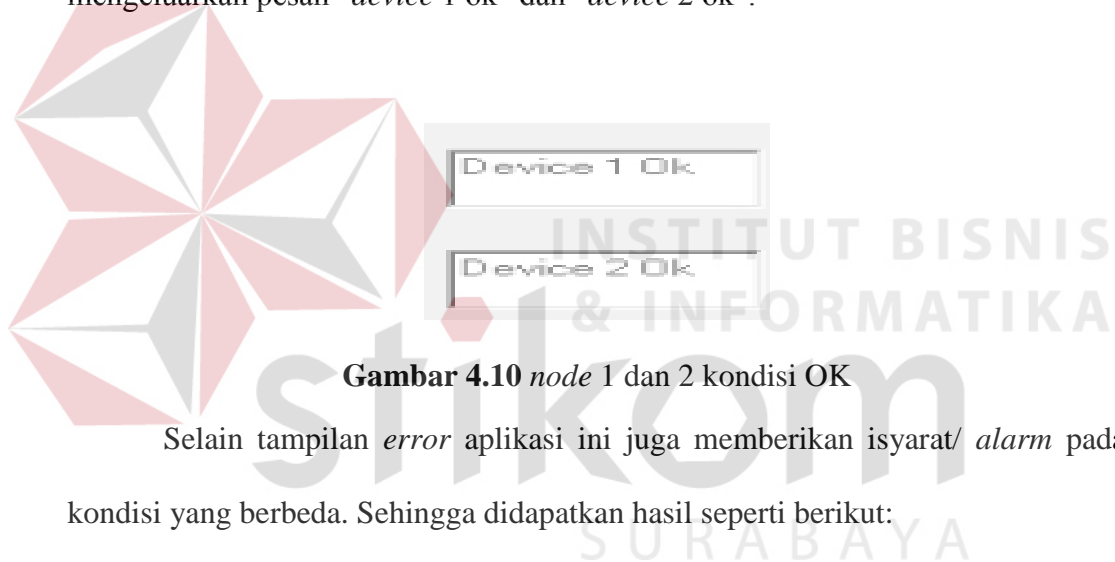
Pada *node coordinator* juga mengirimkan kode “C1” dan “C2” sebagai identitas *node* asal. Jika *node coordinator* mengirim pesan bahwa *node* sensor 1

atau 2 *error* maka pada program ini akan menampilkan informasi *error* tersebut seperti Gambar 4.9.



**Gambar 4.9** *device 1 error*

Dalam gambar yang ditunjukkan oleh Gambar 4.10 di bawah merupakan tampilan dimana semua koneksi terhubung dengan baik, maka aplikasi mengeluarkan pesan “*device 1 ok*” dan “*device 2 ok*”.



**Gambar 4.10** *node 1 dan 2 kondisi OK*

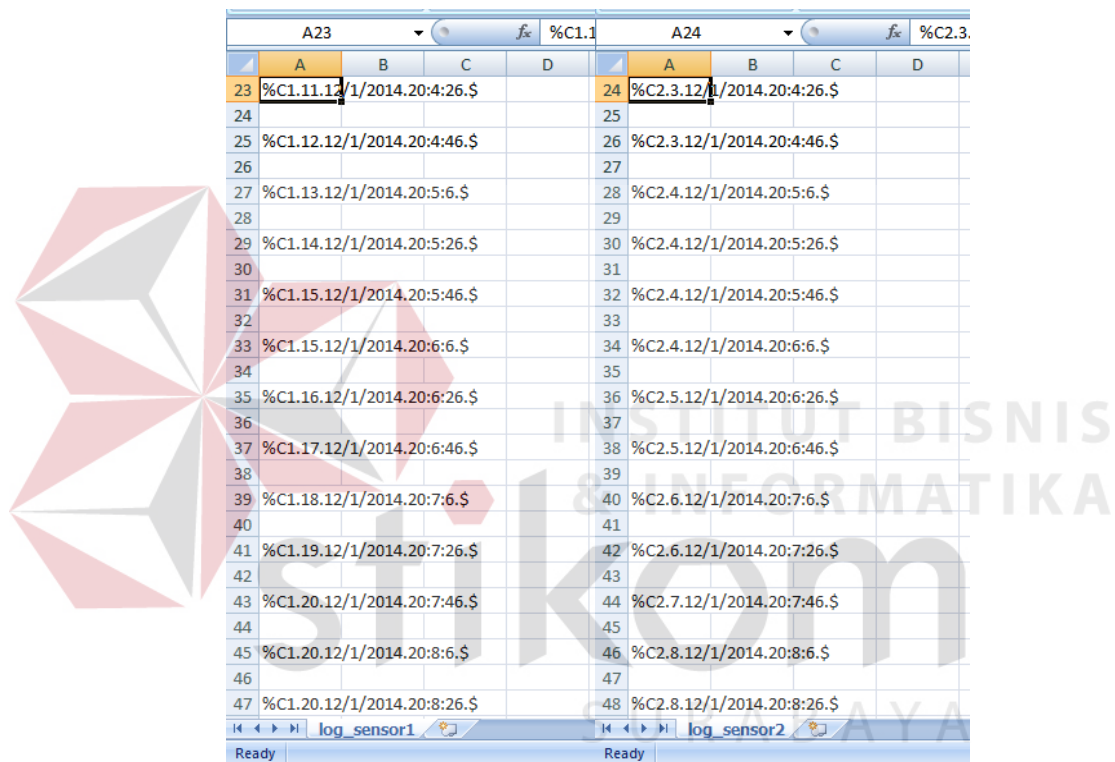
Selain tampilan *error* aplikasi ini juga memberikan isyarat/ *alarm* pada kondisi yang berbeda. Sehingga didapatkan hasil seperti berikut:

Tabel 4.3 Alarm untuk *device error*

Sensor 1	Sensor 2	Alarm
<i>Device 1 OK</i>	<i>Device 2 OK</i>	-
<i>Device 1 Error</i>	<i>Device 2 OK</i>	“ <i>Device one on damage</i> ”
<i>Device 1 OK</i>	<i>Device 2 Error</i>	“ <i>Device two on Damage</i> ”
<i>Device 1 Error</i>	<i>Device 2 Error</i>	“ <i>Both of device on damage</i> ”

### 3. Pengolahan data

Data yang masuk akan direkam dalam 2 bagian, yaitu data yang diterima sebagai data mentah yaitu data yang langsung diterima atau ditangkap sebelum diproses. Data mentah tersebut akan direkam dalam file bernama “log\_sensor1” dan “log\_sensor2”. Berikut isi dari data mentah (data yang belum diolah) yang direkam.



A23				A24			
A	B	C	D	A	B	C	D
23	%C1.11.12	/1/2014.20:4:26.\$		24	%C2.3.12	/1/2014.20:4:26.\$	
24				25			
25	%C1.12.12	/1/2014.20:4:46.\$		26	%C2.3.12	/1/2014.20:4:46.\$	
26				27			
27	%C1.13.12	/1/2014.20:5:6.\$		28	%C2.4.12	/1/2014.20:5:6.\$	
28				29			
29	%C1.14.12	/1/2014.20:5:26.\$		30	%C2.4.12	/1/2014.20:5:26.\$	
30				31			
31	%C1.15.12	/1/2014.20:5:46.\$		32	%C2.4.12	/1/2014.20:5:46.\$	
32				33			
33	%C1.15.12	/1/2014.20:6:6.\$		34	%C2.4.12	/1/2014.20:6:6.\$	
34				35			
35	%C1.16.12	/1/2014.20:6:26.\$		36	%C2.5.12	/1/2014.20:6:26.\$	
36				37			
37	%C1.17.12	/1/2014.20:6:46.\$		38	%C2.5.12	/1/2014.20:6:46.\$	
38				39			
39	%C1.18.12	/1/2014.20:7:6.\$		40	%C2.6.12	/1/2014.20:7:6.\$	
40				41			
41	%C1.19.12	/1/2014.20:7:26.\$		42	%C2.6.12	/1/2014.20:7:26.\$	
42				43			
43	%C1.20.12	/1/2014.20:7:46.\$		44	%C2.7.12	/1/2014.20:7:46.\$	
44				45			
45	%C1.20.12	/1/2014.20:8:6.\$		46	%C2.8.12	/1/2014.20:8:6.\$	
46				47			
47	%C1.20.12	/1/2014.20:8:26.\$		48	%C2.8.12	/1/2014.20:8:26.\$	

**Gambar 4.11** Isi dari “log\_sensor1.csv” dan “log\_sensor2.csv”

Selanjutnya, pada proses ini merupakan proses data yang masuk dipisah menjadi data tinggi, tanggal dan waktu. Maka pada proses sebelumnya data harus berisikan format seperti ini “%C1.20.12/01/2014.20:08:26.\$”. Ketika proses sebelumnya data memiliki karakter akhir “s” seperti data ini “%C1.errors”, maka data tersebut tidak akan masuk dalam proses ini.



Dengan format diatas proses pembacaan dimulai dengan pembacaan tanda titik (.) sampai tanda titik (.) lagi. Dengan data seperti ini "%C1.20.12/01/2014.20:08:26.\$" maka aplikasi akan membaca bahwa C1 adalah id *node*, data tinggi berisi 20, tanggal = 12/01/2014 dan waktu pengukuran = 20:08:26.

Setelah data mengetahui nilai tinggi, tanggal dan waktu, dalam aplikasi *visual basic* ini data akan direkam dalam file Ms.Excel berformat csv dengan nama "data\_ketinggian1.csv" (untuk id C1/ dari sensor 1) dan "data\_ketinggian2" (untuk id C2/ dari sensor 2). Berikut hasil isi file csv yang telah direkam oleh *software visual basic*.

	A	B	C	D
7	6	12/1/2014	20:02:46	
8	7	12/1/2014	20:03:06	
9	8	12/1/2014	20:03:26	
10	9	12/1/2014	20:03:46	
11	10	12/1/2014	20:04:06	
12	11	12/1/2014	20:04:26	
13	12	12/1/2014	20:04:46	
14	13	12/1/2014	20:05:06	
15	14	12/1/2014	20:05:26	
16	15	12/1/2014	20:05:46	
17	15	12/1/2014	20:06:06	
18	16	12/1/2014	20:06:26	
19	17	12/1/2014	20:06:46	
20	18	12/1/2014	20:07:06	
21	19	12/1/2014	20:07:26	
22	20	12/1/2014	20:07:46	
23	20	12/1/2014	20:08:06	
24	20	12/1/2014	20:08:26	
25	20	12/1/2014	20:08:45	
26	21	12/1/2014	20:09:05	
27	20	12/1/2014	20:09:25	
28	21	12/1/2014	20:09:45	
29	23	12/1/2014	20:10:05	
30	24	12/1/2014	20:10:25	
31	26	12/1/2014	20:10:45	

	A	B	C	D
7	1	12/1/2014	20:02:47	
8	2	12/1/2014	20:03:07	
9	2	12/1/2014	20:03:27	
10	2	12/1/2014	20:03:47	
11	2	12/1/2014	20:04:06	
12	3	12/1/2014	20:04:26	
13	3	12/1/2014	20:04:46	
14	4	12/1/2014	20:05:06	
15	4	12/1/2014	20:05:26	
16	4	12/1/2014	20:05:46	
17	4	12/1/2014	20:06:06	
18	5	12/1/2014	20:06:26	
19	5	12/1/2014	20:06:46	
20	6	12/1/2014	20:07:06	
21	6	12/1/2014	20:07:26	
22	7	12/1/2014	20:07:46	
23	8	12/1/2014	20:08:06	
24	8	12/1/2014	20:08:26	
25	9	12/1/2014	20:08:46	
26	9	12/1/2014	20:09:06	
27	9	12/1/2014	20:09:26	
28	10	12/1/2014	20:09:46	
29	10	12/1/2014	20:10:06	
30	11	12/1/2014	20:10:26	
31	12	12/1/2014	20:10:46	

**Gambar 4.12** Isi dari "data\_ketinggian1.csv" dan "data\_ketinggian2.csv"

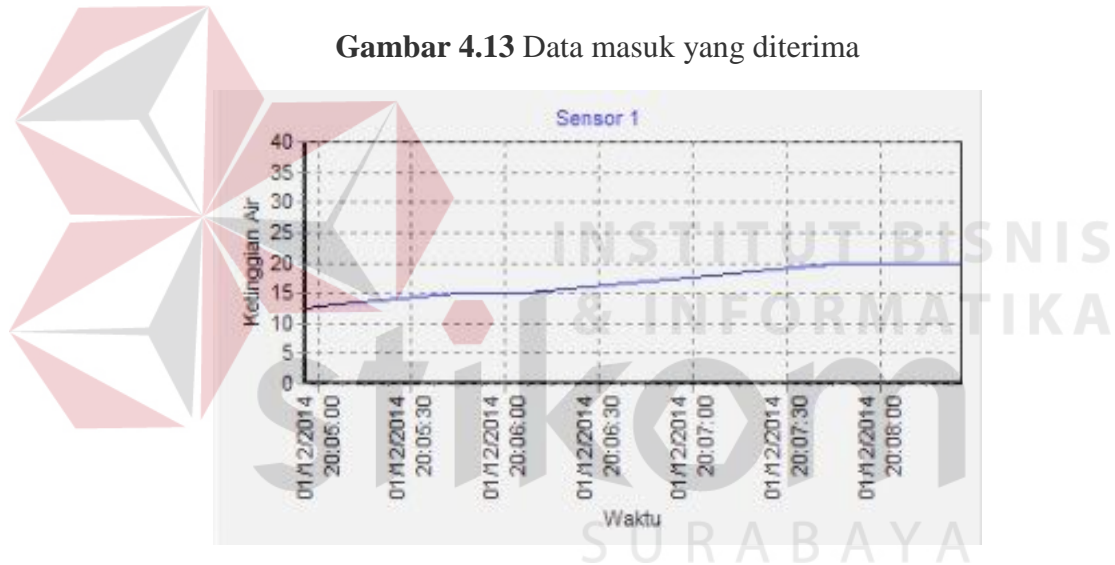
Pada Gambar 4.9 data ketinggian ditampilkan pada kolom 1, tanggal pengukuran ditampilkan pada kolom 2 dan waktu pengukuran ditampilkan pada kolom 3.

#### 4. Menampilkan grafik

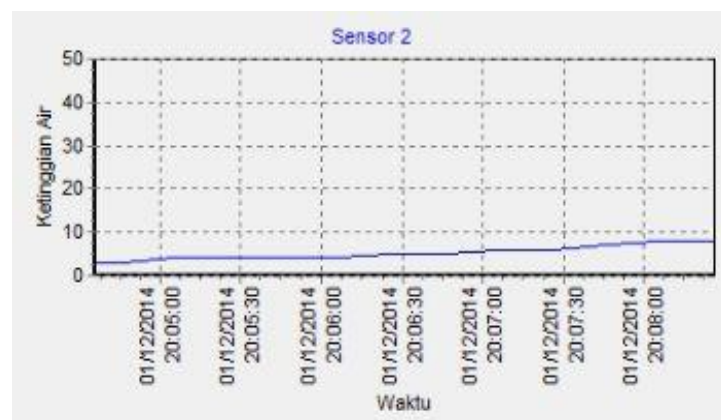
Data yang masuk juga akan ditampilkan dalam grafik. Berikut grafik yang ditunjukkan dari data yang masuk pada Gambar 4.13.

Ketinggian Air Sensor 1	Ketinggian Air Sensor 2
11 cm - 12/1/2014 - 20:4:26	3 cm - 12/1/2014 - 20:4:26
12 cm - 12/1/2014 - 20:4:46	3 cm - 12/1/2014 - 20:4:46
13 cm - 12/1/2014 - 20:5:6	4 cm - 12/1/2014 - 20:5:6
14 cm - 12/1/2014 - 20:5:26	4 cm - 12/1/2014 - 20:5:26
15 cm - 12/1/2014 - 20:5:46	4 cm - 12/1/2014 - 20:5:46
15 cm - 12/1/2014 - 20:6:6	4 cm - 12/1/2014 - 20:6:6
16 cm - 12/1/2014 - 20:6:26	5 cm - 12/1/2014 - 20:6:26
17 cm - 12/1/2014 - 20:6:46	5 cm - 12/1/2014 - 20:6:46
18 cm - 12/1/2014 - 20:7:6	6 cm - 12/1/2014 - 20:7:6
19 cm - 12/1/2014 - 20:7:26	6 cm - 12/1/2014 - 20:7:26
20 cm - 12/1/2014 - 20:7:46	7 cm - 12/1/2014 - 20:7:46
20 cm - 12/1/2014 - 20:8:6	8 cm - 12/1/2014 - 20:8:6
20 cm - 12/1/2014 - 20:8:26	8 cm - 12/1/2014 - 20:8:26

Gambar 4.13 Data masuk yang diterima



Gambar 4.14 Grafik sensor 1



Gambar 4.15 Grafik sensor 2

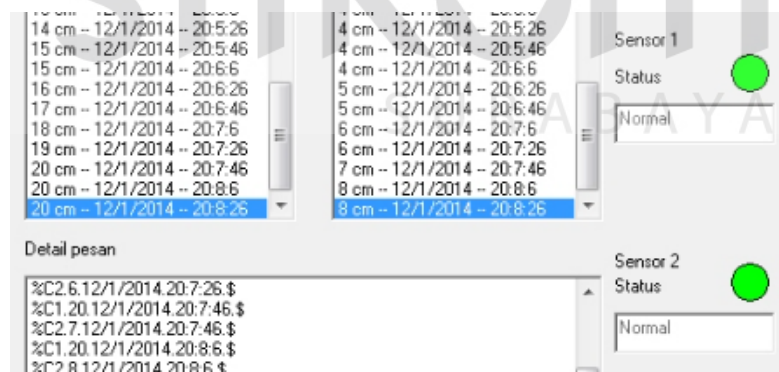
Pada Gambar 4.14 dan Gambar 4.15 menunjukkan data yang ditampilkan pada jam 20:05 sampai 20:08 terakhir. Pembacaan tersebut mempresentasikan sumbu x sebagai waktu dan sumbu y sebagai waktu pengukuran.

## 5. Tampilan indikator warna

Tingkat siaga ini bertujuan untuk mengetahui status ketinggian. Tingkat siaga berubah setiap tinggi mencapai atau melebihi batas normal.

**Tabel 4.3** Ketentuan siaga

Batas tinggi	Status	Warna indikator	Isyarat	
			Sensor 1	Sensor 2
Tinggi < 60% Batas tinggi	Normal	Hijau	Tidak ada	Tidak ada
60% Batas tinggi > Tinggi < 80% Batas tinggi	Siaga 2	Kuning	Perhitungan sensor 2 kapan mencapai siaga 2	Tidak ada
Tinggi >= 80% Batas tinggi	Siaga 1	Merah	Alarm 1x + Perhitungan sensor 2 kapan mencapai siaga 1	Alarm 3x + Perhitungan kapan mencapai meluap



**Gambar 4.16** indikator saat normal

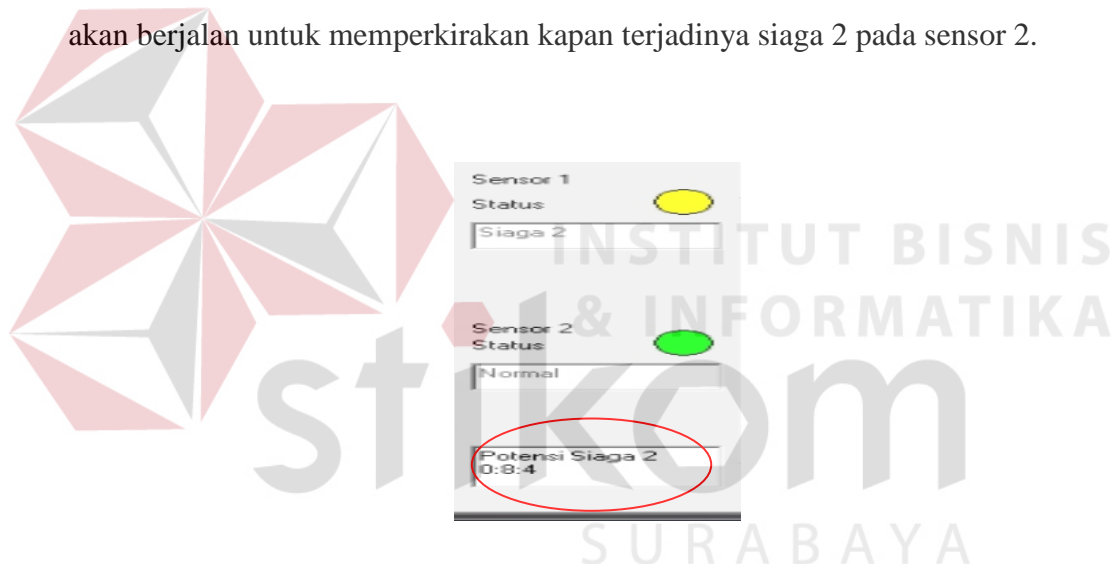
Dalam tugas akhir ini penulis menggunakan status normal bila nilai dibawah 60 % dari tinggi sampel, status siaga 2 bila nilai tinggi yang diterima berkisar antara diatas 60 % dari tinggi sampel dan status siaga 1 untuk nilai tinggi diatas 80 % dari sampel tinggi.

Pada gambar 4.16 menunjukkan indikator pada sensor 1 berwarna hijau dan status normal dikarenakan tinggi yang terakhir diterima bernilai 20. Dimana nilai 20 merupakan nilai dibawah 24 (60 % dari tinggi sampel dari sensor 1 yang bernilai 40). Berikut ketentuan nilai batas yang ditentukan.

## 6. Waktu perkiraan perubahan status

Proses ini merupakan proses pemberian waktu perkiraan untuk sensor 2. Proses ini berjalan ketika pada sensor 1 mencapai status siaga 2 dan siaga 1.

Ketika sensor 1 mencapai siaga 2, maka proses perkiraan untuk sensor 2 akan berjalan untuk memperkirakan kapan terjadinya siaga 2 pada sensor 2.



**Gambar 4.17** Tampilan perkiraan waktu siaga pada sensor 2

Pada Gambar 4.17 merupakan tampilan perkiraan kapan terjadinya siaga 2 pada *node* sensor 2. Perkiraan waktu ini terjadi karena pada sensor 1 mencapai status siaga 2. Perkiraan di atas didapat dari rumusan berikut.

Pada tugas akhir ini, penulis menggunakan interval waktu 20 detik dalam setiap pengiriman. Maka banyaknya waktu untuk mencapai ketinggian air saat itu adalah sebagaimana yang dirumuskan pada rumus (1), perbedaan/perubahan ketinggian air dengan menggunakan rumus (2). Setelah mengetahui perbedaan

ketinggian saat ini, maka kecepatan perubahan tinggi air dapat dihitung dengan rumus (3). Untuk mengetahui kecepatan rata-rata perubahan tinggi air dengan menggunakan rumus (4) dan untuk mengetahui waktu perkiraan dengan menggunakan rumus (5).

Dari perumusan diatas maka akan diketahui berapa banyak waktu air akan mencapai batas siaga 2, siaga 1 dan mencapai batas maksimum/meluap. Waktu yang telah diketahui akan ditampilkan pada aplikasi dan dikonversi dalam jam, menit dan detik.

