

BAB IV

PENGUJIAN SISTEM

Pengujian sistem yang dilakukan penulis merupakan pengujian terhadap perangkat keras dan perangkat lunak dari sistem secara keseluruhan yang telah selesai dibuat untuk mengetahui komponen-komponen sistem apakah berjalan dengan baik.

4.1. Pengujian Arduino

4.1.1. Tujuan

Pengujian Arduino bertujuan untuk mengetahui apakah Arduino dapat melakukan proses *signature* dan *upload* program ke *minimum system* pada Arduino dengan baik.

4.1.2. Alat yang digunakan

Peralatan yang digunakan penulis untuk melakukan pengujian ini adalah sebagai berikut :

1. Arduino.
2. PC.
3. Arduino IDE (*Integrated Development Environment*).
4. Kabel USB.

4.1.3. Prosedur pengujian

Langkah-langkah pengujian untuk melakukan pengujian Arduino adalah sebagai berikut :

1. Sambungkan Arduino ke komputer menggunakan kabel USB.

2. Aktifkan PC dan jalankan program Arduino IDE.
3. Buat program pada Arduino IDE.
4. Setelah membuat program klik file pada jendela arduino IDE kemudian pilih *upload*.
5. Amati pada proses *upload*, pastikan proses berhasil dan sesuai dengan yang diharapkan.

4.1.4. Hasil pengujian

Pengujian arduino pada komputer untuk menulis program dan melakukan *upload* ke arduino berjalan dengan baik, seperti ditunjukkan pada gambar 4.1 berikut.



Gambar 4.1 Program Dan Hasil *Upload*

4.2. Pengujian Xbee

4.2.1. Tujuan

Pengujian komunikasi Xbee ini dilakukan untuk mengetahui komunikasi antar Xbee pro dengan Xbee. Pengujian komunikasi Xbee ini dilakukan dengan

mengkonfigurasi parameter yang telah dilakukan dengan baik dan dapat diterima serta ditampilkan pada layar komputer.

4.2.2. Alat yang digunakan

Peralatan yang digunakan penulis untuk melakukan pengujian ini adalah sebagai berikut :

1. Xbee Pro S2.
2. *Adapter* Xbee.
3. *Software* X-CTU.
4. Kabel USB.
5. Komputer.

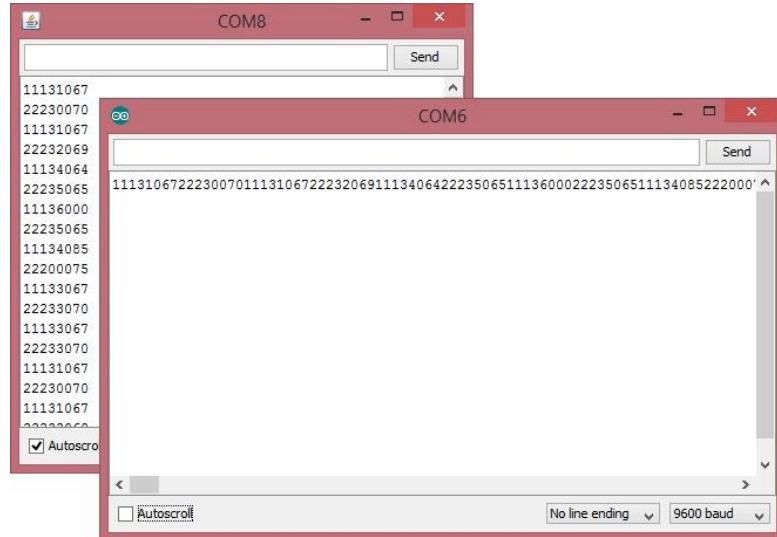
4.2.3. Prosedur pengujian

Langkah-langkah untuk melakukan pengujian komunikasi Xbee adalah sebagai berikut :

1. Tancapkan Xbee pada *adapter*.
2. Hubungkan *adapter* Xbee dengan komputer menggunakan kabel USB.
3. Jalankan program X-CTU pada komputer.
4. Konfigurasi Xbee.
5. Amati data yang ditampilkan untuk mengetahui hasil pengiriman data.

4.2.4. Hasil pengujian

Dari prosedur pengujian komunikasi data antar Xbee yang telah dilakukan dapat berjalan dengan baik, pada Gambar 4.2 adalah hasil pengujian komunikasi antar xbee



Gambar 4.2 Test Komunikasi Xbee

4.3. Pengujian jarak maksimal kemampuan pengiriman data Xbee Pro

4.3.1. Tujuan

Untuk mengetahui kemampuan jangkauan area Xbee pro S2 dalam melakukan penerimaan data dari Xbee S2

4.3.2. Alat yang digunakan

1. Arduino.
2. Xbee pro S2 dan Xbee S2 beserta *shield*.
3. Komputer.
4. Baterai 9V DC.
5. Kabel USB.

4.3.3. Prosedur pengujian

langkah-langkah untuk melakukan pengujian jarak komunikasi Xbee adalah sebagai berikut :

1. Pasang baterai pada papan arduino yang sudah terintegrasi dengan Xbee S2.

2. Sambungkan arduino yang sudah terintegrasi Xbee Pro S2 dengan komputer menggunakan kabel USB untuk *monitoring*.
3. Ukur jarak antar Xbee, dan carilah jangkauan penerimaan maksimal.

4.3.4. Hasil pengujian

dari prosedur pengujian komunikasi data pada Xbee yang telah dilakukan di luar ruangan (*Outdoor Area*) didapatkan hasil pengamatan jangkauan sebagai berikut :

Tabel 4.1 Hasil Pengamatan Jangkauan Xbee Dalam Kondisi Di Luar Ruangan (*Outdoor Area*)

No.	Jarak (meter)	Keterangan
1.	10	Ok
2.	20	Ok
3.	30	Ok
4.	40	Ok
5.	50	Ok
6.	60	Ok
7.	70	Ok
8.	80	Ok
9.	90	Ok
10.	100	Ok
11.	101	Gagal
12.	102	Gagal
13.	103	Gagal
14.	104	Gagal
15.	105	Gagal

4.4. Pengujian Komunikasi *Serial/USB*

4.4.1. Tujuan

Pengujian komunikasi *serial* ini dilakukan untuk mengetahui komunikasi antar Arduino dengan komputer berjalan dengan baik. Pengujian komunikasi

serial/USB dilakukan dengan memprogram karakter atau tulisan yang di tampilkan dan kemudian di cocokan dengan tampilan pada komputer.

4.4.2. Alat yang digunakan

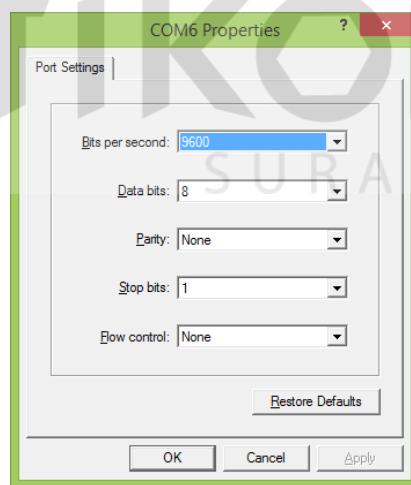
Peralatan yang digunakan penulis untuk melakukan pengujian ini adalah sebagai beriku :

1. Arduino Uno
2. Komputer dan kabel USB
3. *Hyper Terminal*

4.4.3. Prosedur pengujian

Langkah-langkah untuk melakukan pengujian komunikasi *serial / USB* adalah sebagai berikut :

1. Hubungkan Arduino dengan komputer
2. Buka program *hyper terminal* pada komputer, lalu atur konfigurasinya seperti gambar berikut.

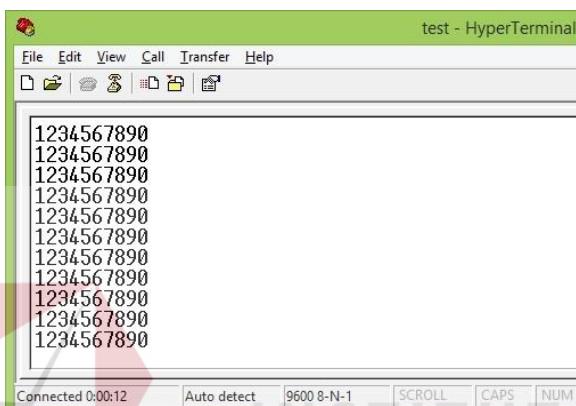


Gambar 4.3 Setting Port

3. Setelah setting port kemudian klik Ok, kemuadian amatilah jendela *hyper terminal* yang menampilkan hasil yang diterima melalui komunikasi *serial / USB*.

4.4.4. Hasil Pengujian

Pengujian komunikasi *serial* antar Arduino dengan komputer dapat dilihat pada jendela *hyper terminal*. Hasil *capture* pengujian komunikasi *serial* dapat dilihat pada Gambar 4.4 berikut :



The screenshot shows a window titled "test - HyperTerminal". The menu bar includes "File", "Edit", "View", "Call", "Transfer", and "Help". Below the menu is a toolbar with icons for copy, paste, cut, and others. The main window displays a series of identical text entries: "1234567890" repeated 12 times. At the bottom of the window, there is status information: "Connected 0:00:12", "Auto detect", "9600 8-N-1", and a row of function keys: SCROLL, CAPS, NUM.

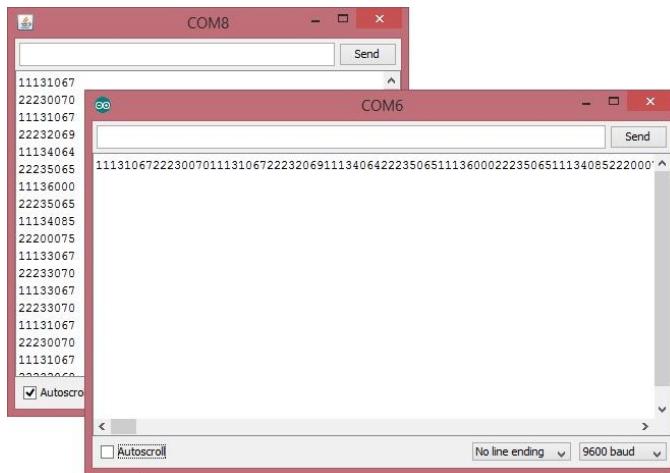
Gambar 4.4 Hasil Pengujian Komunikasi Serial

4.5. Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem ini bertujuan untuk mengetahui apakah sistem yang dirancang dapat berfungsi dengan baik sesuai dengan yang diharapkan. Berikut adalah pengujian dari keseluruhan sistem :

4.5.1. Pengujian data yang diterima oleh *Router* dari *Coordinator*

Pada pengujian ini dimulai dari *coordinator* yang mengirimkan data ke *router*, *coordinator* menggolah data yang diterima dari *end point1* dan *end point2* kemudian setelah data diolah oleh *coordinator* dilanjutkan ke *router*, namun penulis hanya menerangkan mulai dari *coordinator* sampai ke aplikasi yang menampilkan data yang di terima oleh *router*. Pada Gambar 4.5 merupakan hasil *capture* data yang dikirim *coordinator* ke *router user*.



Gambar 4.5 Com8 Monitor Coordinator , Com6 Monitor Router

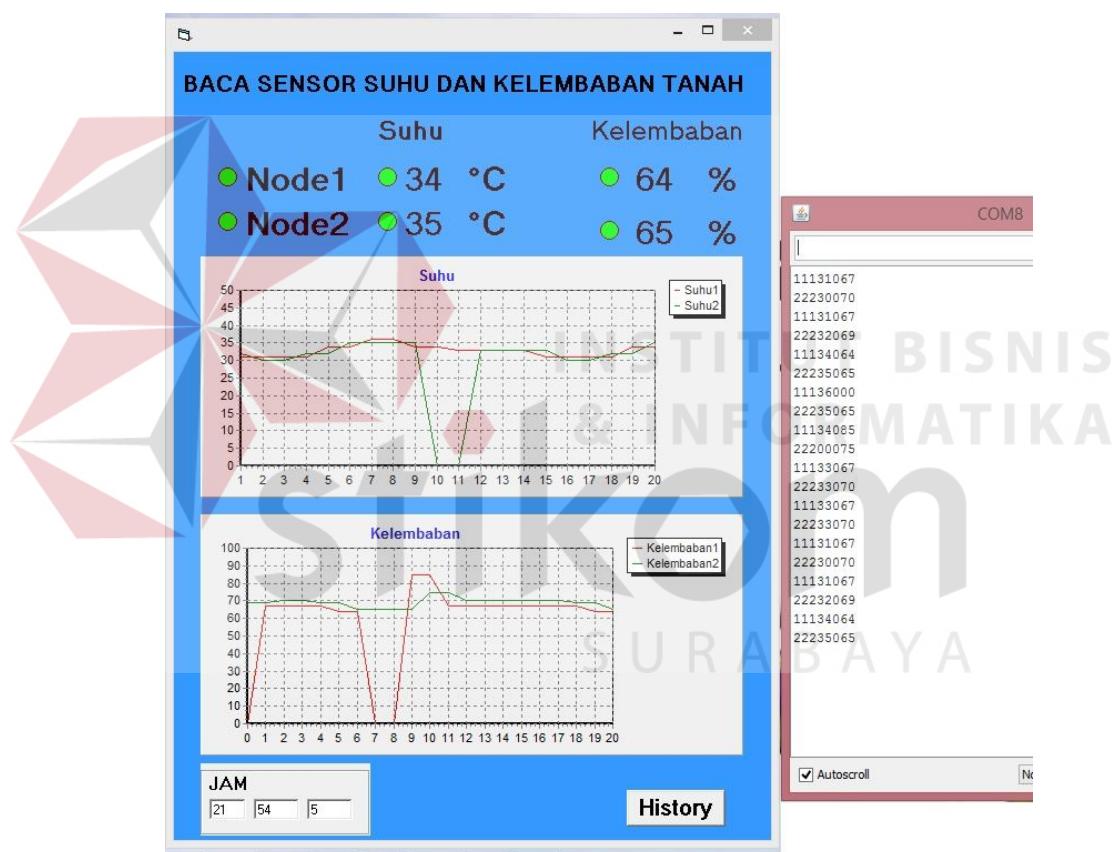
Tabel 4.2 Data Yang Diambil Dari Hasil Capture Pada Gambar 4.5

Data ke-	Data yang dikirim Coordinator	Data yang diterima Router	Data ke-	Data yang dikirim Coordinator	Data yang diterima Router
1	11131067	11131067	16	22230070	22230070
2	22230070	22230070	17	11131067	11131067
3	11131067	11131067	18	22232069	22232069
4	22232069	22232069	19	11134064	11134064
5	11134064	11134064	20	22235065	22235065
6	22235065	22235065	21	11136000	11136000
7	11136000	11136000	22	22235065	22235065
8	22235065	22235065	23	11134085	11134085
9	11134085	11134085	24	22200075	22200075
10	22200075	22200075	25	11133067	11133067
11	11133067	11133067	26	22233070	22233070
12	22233070	22233070	27	11133067	11133067
13	11133067	11133067	28	22233070	22233070
14	22233070	22233070	29	11131067	11131067
15	11131067	11131067	30	22230070	22230070

Dilihat dari beberapa data pengujian pada Tabel 4.2 menunjukkan data yang dikirim oleh *coordinator* ke *router user* berjalan dengan baik hal ini menunjukkan komunikasi antara *coordinator* dengan *router user* tidak ada data yang *loss*.

4.5.2. Pengujian data yang diterima oleh aplikasi visual basic dari *coordinator* melalui *router*

Pada pengujian ini data yang dikirim oleh *coordinator* dibandingkan dengan nilai yang akan ditampilkan pada aplikasi. Aplikasi ini berguna untuk menampilkan data yang terbaca oleh sensor dan sebagai pengingat untuk proses *monitoring*. Pada gambar 4.6 menampilkan nilai suhu dan kelembaban yang dikirim *coordinator* dan yang ditampilkan aplikasi.



Gambar 4.6 Data Yang Ditampilkan Aplikasi Pada VB 6.0 dan *Monitoring* Pada *Coordinator*

Pada pengujian pengiriman data antara *coordinator* dengan aplikasi pada VB 6.0 melalui *router* dan data yang ditampilkan aplikasi menunjukkan hasil yang sama. Maka dapat disimpulkan bahwa data yang diterima oleh aplikasi

menunjukkan sistem berjalan dengan baik dan data yang diterima sesuai dengan data yang dikirim oleh *coordinator* melalui *router*.

4.5.3. Pengujian Indikator Dan *Alarm* Pada Node Nilai Sensor

Pada pengujian ini menunjukkan bahwa aplikasi menunjukkan sebuah perubahan warna dan bunyi *alarm* atau *beep* sebagai pengingat kepada *user* apabila terdapat *error* pada sistem dan menunjukkan apabila data yang diterima tidak sesuai dengan data atau nilai yang telah ditetapkan. Untuk dapat menentukan batas minimum dan maksimum pada aplikasi *monitoring* ini mengacu pada bab 1 pendahuluan.

pada Gambar 4.7 menunjukkan bahwa *alarm* berbunyi (*beep*) dan warna indikator berubah menjadi warna merah jika nilai suhu maupun kelembaban tidak sesuai dengan nilai yang telah ditetapkan untuk lahan yang akan dipantau (*monitoring*). *Alarm* berbunyi dan indikator berubah warna menjadi warna merah yang semula berwarna hijau jika suhu yang diterima minimum 20° dan maksimum $> 35^{\circ}$, sedangkan untuk sensor kelembaban batas minimum 60 % RH dan maksimum 80 % RH. Tetapi jika salah satu sensor menampilkan nilai 0 menunjukkan bahwa terdapat *error* pada sensor. Berikut adalah cuplikan program yang menentukan warna indikator berubah sesuai batas minimum dan maksimum yang telah ditentukan.

```
If data1 < 20 Or data1 > 35 Then
    Timer2.Enabled = True
ElseIf data1 > 19 And data1 < 35 Then
    Timer2.Enabled = False
    Shape1.FillColor = vbGreen
End If

If data2 < 60 Or data2 > 80 Then
```

```

    Timer3.Enabled = True
    ElseIf data2 > 59 And data2 < 81 Then
        Timer3.Enabled = False
        Shape2.FillColor = vbGreen
    End If

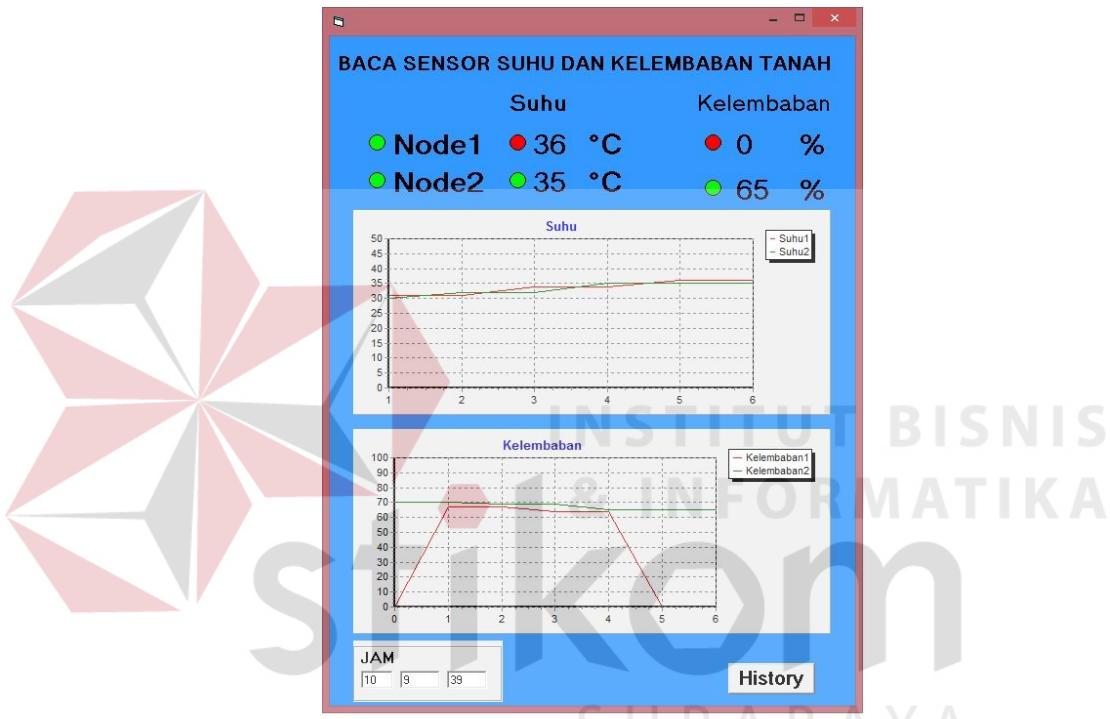
```

Berikut ini adalah cuplikan program untuk mengaktifkan *alarm*

```

Private Sub Timer6_Timer()
    Call AvirasBeep
End Sub

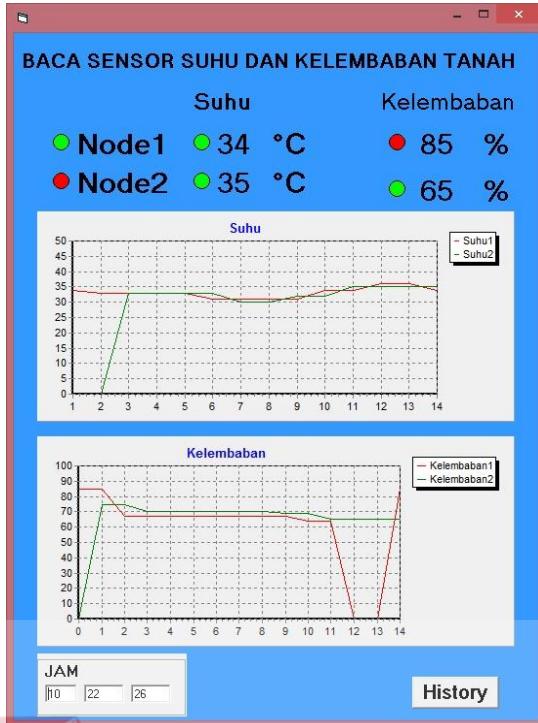
```



Gambar 4.7 Indikator Sensor Suhu Pada *Node1* Melebihi Batas Yang Ditetapkan,
Sensor Kelembaban Pada *Node1* Terjadi *Error*

4.5.4. Pengujian Indikator dan *Alarm* pada *Node*

Pada pengujian ini apabila data yang diterima melebihi batas yang telah ditentukan maka indikator *node* pada aplikasi yang semula berwarna hijau akan berubah menjadi berwarna merah dan *alarm* berbunyi



Gambar 4.8 Indikator pada *Node 2* Terjadi *Error*

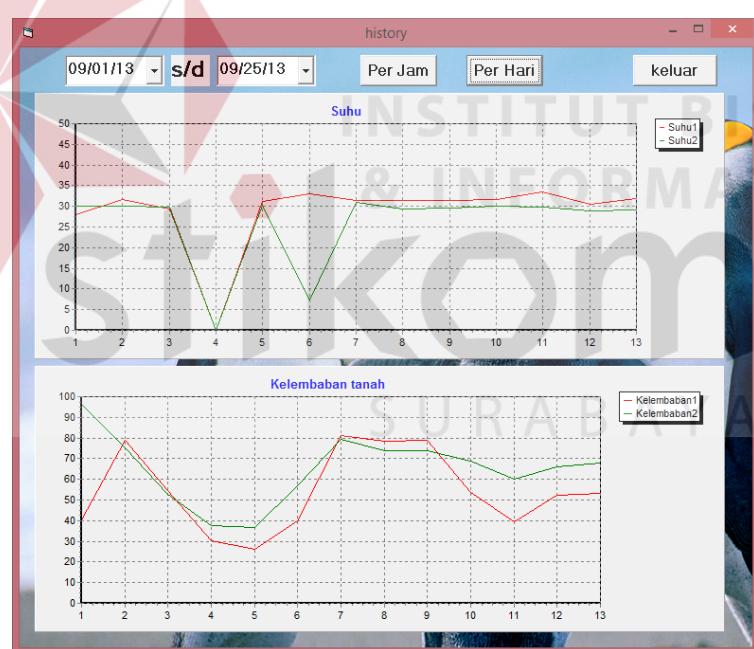
Pada Gambar 4.8 suhu dan kelembaban menunjukkan nilai yang diterima dan indikator pada suhu dan kelembaban berwarna hijau tetapi indikator pada *node2* berubah menjadi warna merah hal tersebut menunjukkan bahwa *node 2* koneksi terputus atau daya baterai habis dan tidak dapat mengirimkan data dalam waktu tertentu.

4.5.5. Pengujian pada form *history*

Pengujian *history* disini ada 2 macam yang pertama yaitu *history* untuk menampilkan rata-rata perhari, yang kedua yaitu untuk menampilkan rata-rata perjam. Kedua *history* ini dapat ditentukan oleh *user* sesuai tanggal bulan tahun dari data yang disimpan. Berikut adalah gambar hasil *history* dari rata-rata perjam dan perhari.



Gambar 4.9 History dari Rata-Rata Perjam



Gambar 4.10 History dari Rata-Rata Pertanggal/Perhari