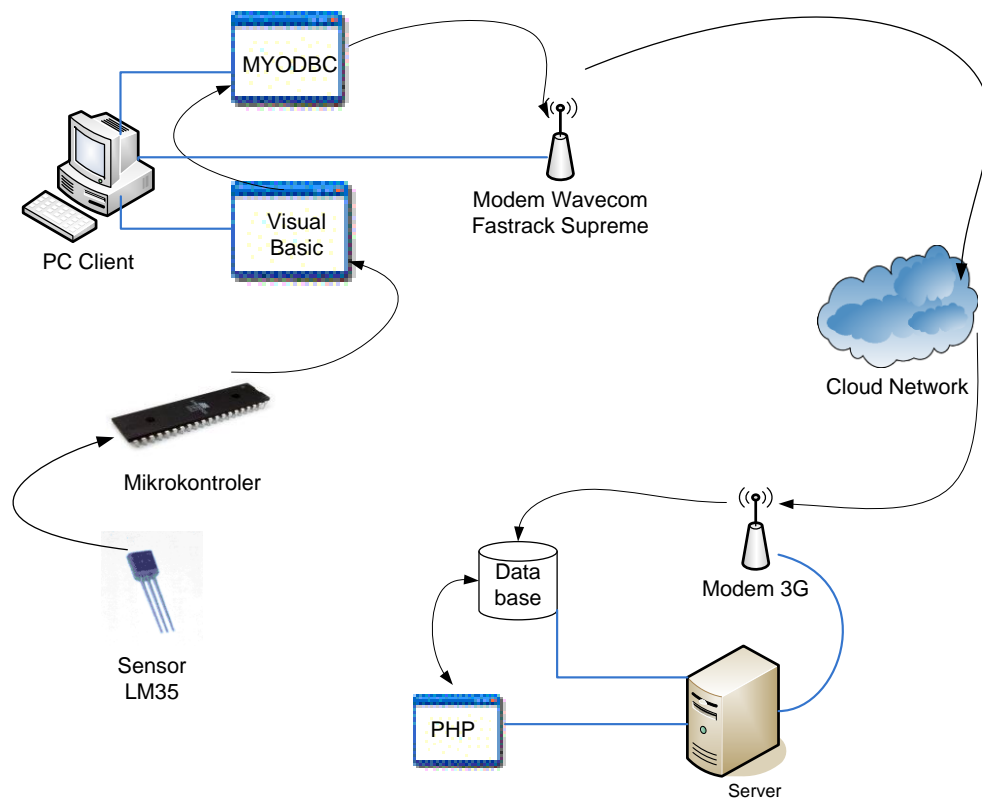


IV. HASIL DAN PEMBAHASAN

Setelah merencanakan suatu alat dan membuatnya sampai terbentuk sistem rangkaian lengkap, selanjutnya dilakukan pengecekan dan pengukuran terhadap alat tersebut. Pengujian alat dilakukan untuk mengetahui apakah peralatan yang ada pada perangkat keras berjalan sesuai yang diharapkan atau belum. Pengujian perangkat keras dan pengujian perangkat lunak dengan menjalankan sistem secara keseluruhan yang bertujuan untuk mengetahui apakah alat dan program yang telah dibuat menghasilkan tampilan sesuai yang dimaksudkan atau belum. Pengujian terhadap sistem ini dilakukan pada beberapa lokasi yang berbeda. Hal tersebut penting dilakukan mengingat kualitas sinyal dan kesibukan jaringan di tiap wilayah pasti berbeda. Hal tersebut sangat mempengaruhi *Quality of Service* (QOS) pada sistem ini.

Implementasi perancangan pada tugas akhir ini dapat digambarkan sebagai berikut.



Gambar 37. Implementasi perancangan sistem

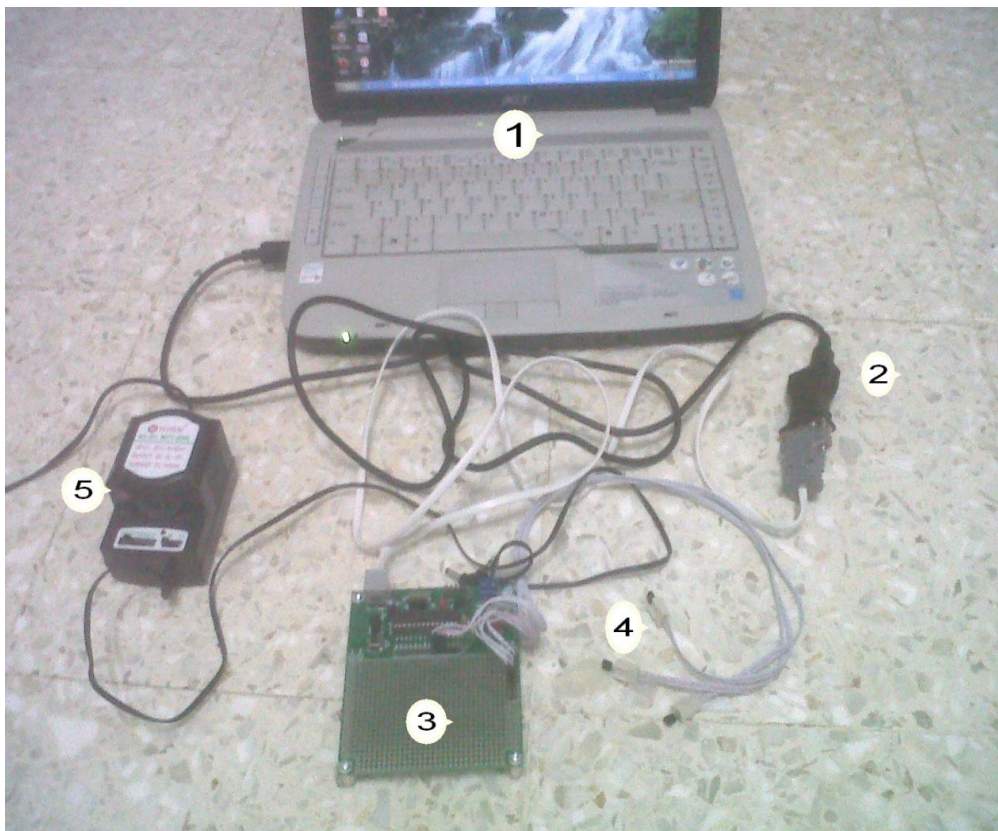
Pada pengujian ini menggunakan koneksi yang berbasis teknologi GPRS. Sehingga hasil pengamatan sangat bergantung pada kondisi trafik kesibukan dari *Internet Service Provider* (ISP) yang digunakan.

A. Hasil Pengujian Perangkat Keras

Pengujian dilakukan dengan cara pemeriksaan rangkaian serta menguji komponen penunjangnya secara keseluruhan. Pengujian ini dilakukan untuk mengetahui peralatan yang ada pada *hardware* yang dibuat (baik buruknya kondisi alat dan kinerjanya).

1. Sistem *Monitoring* Suhu Ruangan

Hasil dari perancangan perangkat keras sistem *monitoring* ini ditunjukkan pada gambar di bawah ini.



- Keterangan :
1. *Notebook*
 2. *Prolific Serial to USB Connector*
 3. *DT proto 28 pin AVR*
 4. *Sensor suhu LM 35*
 5. *Power Supply*

Gambar 38. Foto rangkaian sistem *monitoring* suhu


```

:konfigurasi PinC
ldi r16,0b00000000
out ddrC,r16           ;Pin C sebagai input

```

Gambar 40. Konfigurasi *input* pin C

Komunikasi *half duplex* pada mikrokontroler bekerja pada *baudrate* 9600 dengan 1 bit stop dan 8 bit karakter. Berikut *listing* program komunikasi serial pada mikrokontroler.

```

:konfigurasi serial interface
ldi r16,0b00001000     ;Transmit enable
out UCSRB,r16
ldi r16,0b10000110     ;UBRRH digunakan,asinkron,tanpa paritas
out UCSRC,R16          ;1bit stop,8 bit karakter
ldi r16,0x0C           ;setting baudrate 9600
out UBRRL,r16          ;Asinkron kecepatan normal
ldi r16,0x00           ;UBRR=12=0x000C
out UBRRH,r16

```

Gambar 41. Konfigurasi serial pada ATmega8

Pengeksekusian ADC untuk masing-masing sensor dilakukan secara bergantian dan berurutan. Data hasil eksekusi ADC disimpan dalam 3 register yang berbeda untuk kemudian dipanggil pada saat pengiriman data.

```

:Analog to Digital Converter
konversi_ADC0 :
  ldi r16,0b11100000     ;Set ADC dengan AREF,ADLAR 1,
  out ADMUX,r16          ;dan input ADC0
  ldi r16,0b11000100     ;ADC start,ADSC=1 konv. mulai
  out ADCSRA,r16         ;prescaler 16
Pantau_ADIF0:
  sbis ADCSRA,ADIF       ;pantau ADIF
  rjmp Pantau_ADIF0
  in r20,ADCH            ;masukkan hasil konv ke r20
  cbi ADCSRA,ADSC        ;kembalikan ADSC ke 0
  sbi ADCSRA,ADIF        ;kembalikan ADIF ke 0
  rjmp konversi_ADC1

```

Gambar 42. Konfigurasi ADC

Proses pengiriman data diawali dengan mengirim nilai penanda awal 255 kepada aplikasi *Visual Basic*. Selanjutnya data hasil konversi ADC yang telah disimpan pada register dikirim secara berurutan.

```

;Pengiriman dan penerimaan data
kirim:
    sbis UCSRA,UDRE           ;pantau UDRE (siap terima data baru)
    rjmp kirim
    ldi r17,0xff
    out UDR,r17
    rcall delay_250ms
    rjmp kirim0

kirim0:
    sbis UCSRA,UDRE           ;pantau UDRE (siap terima data baru)
    rjmp kirim0
    out UDR,r20
    rcall delay_250ms
    rjmp kirim1

kirim1:
    sbis UCSRA,UDRE           ;pantau UDRE (siap terima data baru)
    rjmp kirim1
    out UDR,r21
    rcall delay_250ms
    rjmp kirim2

kirim2:
    sbis UCSRA,UDRE           ;pantau UDRE (siap terima data baru)
    rjmp kirim2
    out UDR,r22
    rcall delay_250ms
    rjmp konversi_ADC0       ;kirim lagi data ADC

```

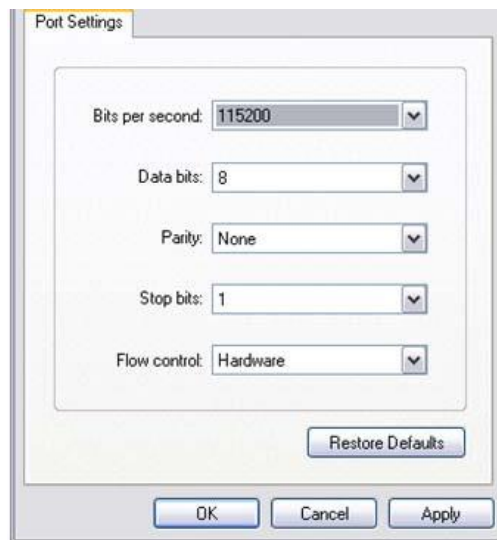
Gambar 43. Listing program pengiriman data

3. Modem GPRS Wavecom Fastrack Supreme 10

Modem yang digunakan pada penelitian ini adalah Wavecom Fastrack Supreme dan dihubungkan dengan menggunakan koneksi *serial*. Namun pada pengujian ini, digunakan perangkat keras tambahan yaitu *serial to usb converter* sehingga *modem* dapat bekerja menggunakan *port* USB.

Agar *modem* dapat bekerja harus dilakukan konfigurasi terlebih dahulu. Adapun langkah yang harus dilakukan adalah:

1. Menghubungkan PC/komputer dengan *modem* Wavecom Fastrack Supreme 10 dengan koneksi kabel *serial* RS232 lengkap dengan *antena*, *power supply*, dan *SIM card*.
2. Melakukan *setting port* pada *Windows Hyper Terminal*, dan memastikan menggunakan parameter *boudrate* 115200 seperti gambar berikut.



Gambar 44. *Port setting* pada *hyper terminal*

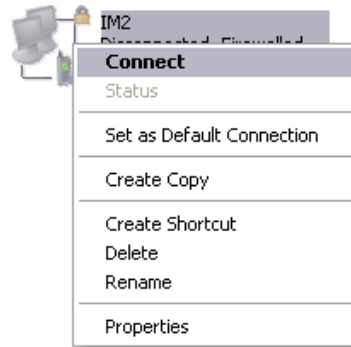
3. Menggunakan fungsi *AT command* untuk mengkonfigurasi modem tersebut. Berikut ada beberapa contoh *AT command* yang digunakan.

Tabel 4. *AT Command*

AT Command	Description	Response
ATZ	Memeriksa koneksi PC dengan Modem	OK
AT&V	Memperlihatkan seluruh parameter	'Parameter setting'
AT+CGSN	Memperlihatkan IMEI	'IMEI number'
AT+CSQ	Memperlihatkan Kualitas sinyal	'besarnya sinyal'

4. Memasukan konfigurasi *Access Point Name* (APN) menggunakan perintah *AT command*. Langkah ini bergantung pada jenis APN yang digunakan. Berikut adalah contoh perintah untuk beberapa APN.
 - a. Telkomsel = `at+cgdcont=1,ip,"telkomsel"`
 - b. ProXL = `at+cgdcont=1,ip,www.xlgprs.net`

- c. M3 = at+cgdcont=1,ip,www.indosat-m3.net
 - d. Mentari = at+cgdcont=1,ip,"satelindogprs.com"
 - e. Three = at+cgdcont=1,ip,"3gprs"
5. Membuat sebuah konfigurasi *modem dial up* pada komputer yang berfungsi untuk mengaktifkan *modem* seperti gambar berikut.



Gambar 45. Menu *dial up*

6. Pada penelitian ini menggunakan layanan GPRS ProXL dan Telkomsel *Flash* sehingga harus menggunakan pengaturan *dial up* seperti di bawah ini.

a. ProXL

Connection Name : XL GPRS
Data Bearer : GPRS
Access Point Name : www.xlgprs.net
User name : xlgprs
Password : proxl

b. Telkomsel *Flash*

Connection Name : Telkomsel *Flash*

Data Bearer : GPRS
Access Point Name : telkomsel
User name : wap
Password : wap123

B. Hasil Pengujian pada Komputer *Client*

1. Pengujian perangkat lunak Visual Basic 6

Pada sistem *monitoring* ini digunakan perangkat lunak Visual Basic 6 sebagai media antar muka yang dapat menampilkan hasil konversi data 8 bit yang diterima dari *micro controller* ATmega8. Terdapat 3 buah sensor suhu LM 35 yang diimplementasikan pada sistem ini.

Langkah awal yang dilakukan pada aplikasi ini adalah mendeklarasikan koneksi serial yang terhubung dengan peralatan sensor. Untuk itu digunakan *fungsi* berikut:

```

Dim buffer As String
Dim Data, Tg1, Tg2, Tg3, Tx1, Tx2, Tx3, i As Integer

'Deklarasi Koneksi
Dim rc As New ADODB.Recordset
Dim cnn As New ADODB.Connection
Dim Cmd As New ADODB.Command

Private Sub Command3_Click()
  cnn.Close
  Set cnn = Nothing
  'rc.Close
  Unload Me

Private Sub Form_Load()
  Dim msql As String
  Dim KoneksiData As String
  MSComm1.CommPort = 13
  MSComm1.Settings = "9600,N,8,1"
  MSComm1.InputLen = 1
  MSComm1.RThreshold = 8
  MSComm1.PortOpen = True
  Timer1.Enabled = True
  
```

End Sub

Setelah dapat berkomunikasi, maka akan ditampilkan hasil pembacaan data yang diterima ke dalam tampilan dengan fungsi sebagai berikut:

```

Sub Writefile()
Dim msq1 As String
Label7.Caption = Time
Label4.Caption = Date
'Mengisi Record ke Tabel
cnm.BeginTrans
msq1 = " INSERT INTO data_suhu(suhu1, suhu2, suhu3, tanggal, waktu)" & _
" VALUES (" & Label1.Caption & "," & _
" " & Label5.Caption & "," & "" & Label6.Caption & "," & _
" " & Label4.Caption & "," & "" & Label7.Caption & ")"

```

Berikut adalah jendela tampilan sistem *monitoring* suhu ruangan.



Gambar 46. Jendela tampilan Visual Basic

Pada gambar di atas terdapat 3 buah kolom yang berisi data suhu, kolom tanggal dan kolom waktu. Data suhu didapat dari pengolahan data 8 bit yang dikirimkan *micro controller*. Setelah mendeklarasikan pembacaan dari masukan sensor, aplikasi ini akan meng-*input file* data ke dalam *database* yang terletak pada komputer *server* dengan perantara aplikasi MyOdbc versi 3.51. Berikut adalah perintah pada *Visual Basic* untuk menggunakan aplikasi MyOdbc sebagai perantara koneksi terhadap *database* MySQL.

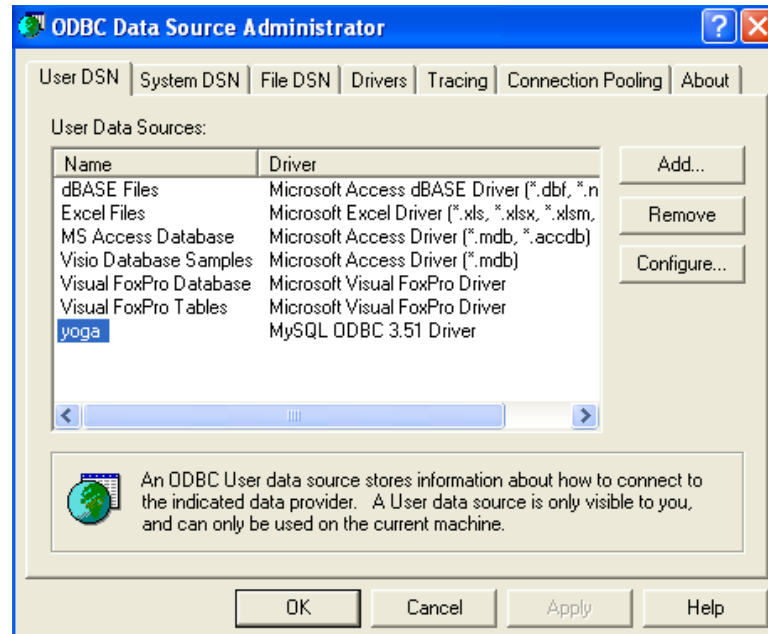
```
cnn.CursorLocation = adUseClient
```

```
cnn.Open "provider=MSDASQL.1;Persist Security Info=False;Data source=yoga"
```

2. Implementasi MyOdbc 3.51

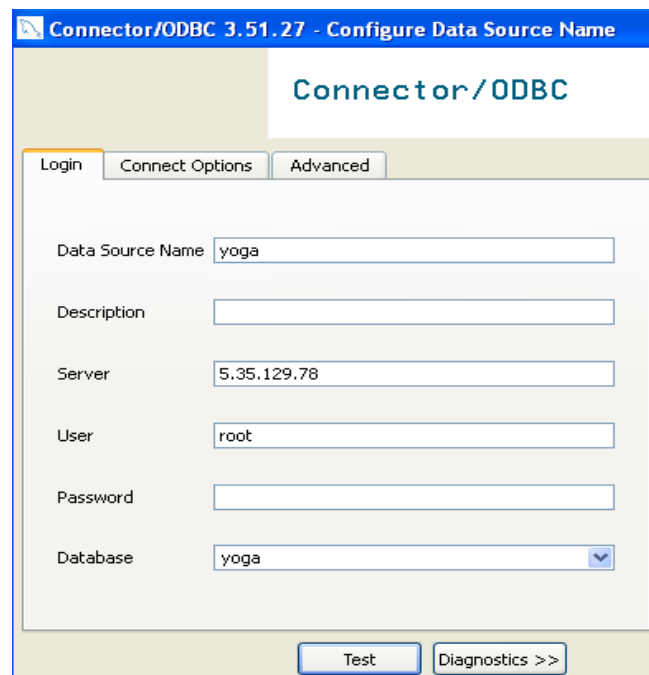
Untuk dapat menggunakan *database* MySQL, dibutuhkan sebuah perangkat lunak yang menjembatani aplikasi *Visual Basic* dan MySQL. Perangkat lunak yang digunakan adalah MyOdbc 3.51 *driver*.

Langkah selanjutnya adalah dengan membuat sebuah *user* DSN agar MySQL dapat berintegrasi dengan Visual Basic. Pada tugas akhir ini telah dibuat sebuah *user* DSN seperti pada gambar berikut:



Gambar 47. ODBC *data source administrator*

Odbc akan mengirimkan *file database* yang terletak pada PC *server*. Terdapat beberapa konfigurasi untuk dapat terhubung dengan *database*. Berikut adalah konfigurasi yang digunakan pada penelitian ini.



Gambar 48. Menu konfigurasi MyOdbc 3.51

Data Source Name adalah *link* komunikasi yang akan digunakan oleh *Visual Basic* untuk dapat meng-*input* data. Sedangkan untuk koneksi dengan *database MySQL* terdapat pada kolom menu *server* yang diisi sesuai dengan IP komputer *server* yang dituju. Menu *user* dan *password* harus diisi apabila *database* diproteksi dengan *password*.

C. Hasil pengujian pada Komputer Server

Telah dilakukan pengujian pada komputer *server* meliputi beberapa aspek, diantaranya adalah *web interface* untuk pemantauan sensor dan kehandalan komunikasi sistem antara *client-server* yang menggunakan koneksi internet berbasis GPRS.

1. Implementasi *database MySQL* pada sistem *monitoring*

Langkah pertama yang dilakukan sebelum membuat *database* adalah mengaktifkan *apache* dan *MYSQL-D-NT* agar *database* yang di buat dapat terhubung pada *server*. Kemudian membuat *database* baru dengan format *.sql*. Sesuai dengan kebutuhan yang sudah dijelaskan pada bab sebelumnya, *database* yang dibuat terdiri atas 2 tabel yaitu tabel *data_suhu* dan tabel *user*. Pada penelitian ini *database* yang dibuat yaitu “*yoga.sql*” seperti pada gambar di bawah ini.

Database yoga - table data_suhu running on localhost

[Browse] [Select] [Insert] [Empty] [Drop]

	Field	Type	Attributes	Null	Default	Extra	Action					
<input type="checkbox"/>	suhu1	int(20)		Yes	NULL		Change	Drop	Primary	Index	Unique	Fulltext
<input type="checkbox"/>	suhu2	int(20)		Yes	NULL		Change	Drop	Primary	Index	Unique	Fulltext
<input type="checkbox"/>	suhu3	int(20)		Yes	NULL		Change	Drop	Primary	Index	Unique	Fulltext
<input type="checkbox"/>	tanggal	varchar(12)		Yes	NULL		Change	Drop	Primary	Index	Unique	Fulltext
<input type="checkbox"/>	waktu	varchar(12)		Yes	NULL		Change	Drop	Primary	Index	Unique	Fulltext

↑ With selected: Or

Gambar 49. Tabel data_suhu

Database yoga - table user running on localhost

[Browse] [Select] [Insert] [Empty] [Drop]

	Field	Type	Attributes	Null	Default	Extra	Action					
<input type="checkbox"/>	nama	varchar(20)		Yes	NULL		Change	Drop	Primary	Index	Unique	Fulltext
<input type="checkbox"/>	nm_lengkap	varchar(35)		Yes	NULL		Change	Drop	Primary	Index	Unique	Fulltext
<input type="checkbox"/>	pswr	varchar(20)		Yes	NULL		Change	Drop	Primary	Index	Unique	Fulltext
<input type="checkbox"/>	level	varchar(25)		Yes	NULL		Change	Drop	Primary	Index	Unique	Fulltext
<input type="checkbox"/>	id	int(3)		Yes	NULL	auto_increment	Change	Drop	Primary	Index	Unique	Fulltext

↑ With selected: Or

Gambar 50. Tabel user

2. Pengujian Website

Website dibuat agar admin atau *user* dapat mengetahui secara langsung kondisi dari pembacaan sensor tersebut. Dengan adanya *website ini*, admin dan *user* dapat mengakses di mana dan kapan saja sehingga dapat memudahkan memonitor kondisi ruangan. perancangan ini data yang ditampilkan diurut berdasarkan waktu update data terakhir. Berikut adalah cuplikan *script* untuk fungsi *query* tersebut.

```
$query="select * from data_suhu order by waktu DESC ";
$hasil=mysql_query($query,$koneksi);
```

Pengujian *website* ini dilakukan di *localhost*, sehingga *website* ini dapat diakses pada alamat <http://localhost/yoga/>.

Suhu 1 (°C)	Suhu 2 (°C)	Suhu 3 (°C)	Tanggal / Waktu
29	29	29	11/10/2010 / 9:16:34 PM
29	28	29	11/10/2010 / 9:16:29 PM
29	28	29	11/10/2010 / 9:16:24 PM
29	28	29	11/10/2010 / 9:16:19 PM
29	29	29	11/10/2010 / 9:16:14 PM
29	29	30	11/10/2010 / 8:25:41 PM
30	29	30	11/10/2010 / 8:25:36 PM
30	29	30	11/10/2010 / 8:25:31 PM

Gambar 51. Tampilan *Home Website* sistem *monitoring*

Gambar di atas merupakan tampilan utama *website* yang dapat diakses setelah admin/user melakukan proses otentifikasi identitas. Admin/user dapat langsung melihat hasil pemantauan sensor berupa tabel yang berisi pembacaan 3 buah sensor, tanggal dan waktu saat data dikirimkan sensor. Menu *home* akan menampilkan data terbaru pada kolom teratas, sehingga akan memudahkan admin/user untuk memantau pembacaan sensor. *Delay refresh* pada aplikasi PHP harus disesuaikan terhadap *delay* pembacaan sensor, sehingga data yang akan ditampilkan dapat lebih akurat. Pada pengujian ini digunakan *delay refresh* setiap 5000ms atau 5 detik. Berikut adalah *script* untuk menggunakan fungsi tersebut.

```
<script type=text/javascript>
// The time out value is set to be 10,000 milli-seconds (or 10 seconds)
setTimeout(' document.location=document.location' ,5000);
```

Untuk meng-*edit* tabel data suhu, admin dapat menggunakan menu *edit* yang memungkinkan *administrator* menghapus data satu per satu atau keseluruhan.

Berikut adalah *listing program* dalam PHP untuk menu edit suhu.

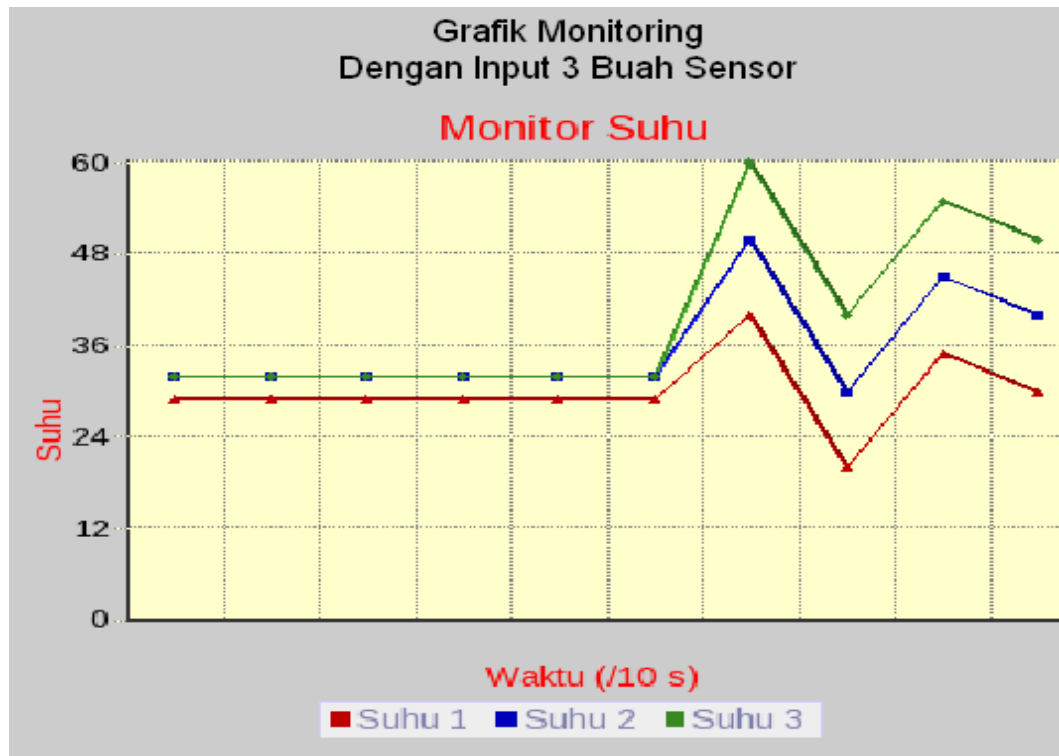
```
<script language="javascript">
function hapus_semua(){
del = window.confirm("Yakin dihapus Semua Datanya ??")
if (del){
return true;
}else{
return false;
}
return false;
}
</script>
```

```
<script language="javascript">
function hapus(){
del = window.confirm("Yakin dihapus?")
if (del){
return true;
}else{
return false;
}
return false;
}
</script>
```


Gambar 52. Menu *edit*

Admin dapat melakukan penghapusan data satu per satu dengan meng-klik simbol silang di sisi kanan tabel dan dapat melakukan penghapusan seluruh data dengan menggunakan *link* di sisi atas tabel data. Fungsi *edit* tersebut hanya dapat digunakan oleh *administrator*.

Selanjutnya terdapat fungsi dari pemrograman PHP ini untuk menampilkan data pemantauan sensor dalam bentuk grafik garis. Implementasi grafik ini menggunakan bahasa pemrograman *Java* oleh sebab tidak adanya aplikasi grafik garis pada pemrograman PHP. User dapat menampilkan fungsi ini dengan meng-klik *link* grafik pada halaman *web*. Fungsi grafik langsung menampilkan setiap *update* data pada *database* MySQL. Berikut adalah tampilan menu fungsi grafik pada halaman *web* yang dibuat.



Gambar 53. Tampilan grafik pemantauan sensor

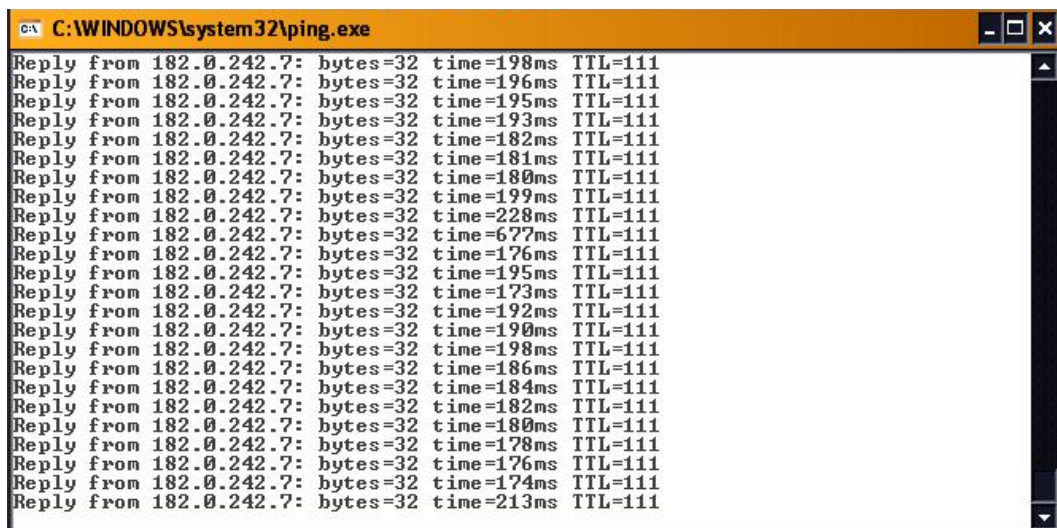
Pada fungsi grafik garis di atas, terdapat sumbu X dan Y dimana sumbu X adalah interval waktu pembacaan sensor, dan sumbu Y merupakan nilai suhu dalam satuan derajat *Celcius* yang akan menyesuaikan otomatis. Suhu 1, 2 dan 3 dilambangkan dengan garis grafik dengan warna yang berbeda agar memudahkan *administrator* untuk membedakan tiap-tiap masukan. Fungsi grafik ini dapat diakses dan dilihat oleh *user* ataupun *administrator*.

D. Hasil Pengujian *Transfer Data* Berbasis GPRS

Setelah antar muka *website* selesai, maka langkah selanjutnya adalah pengujian pengiriman data yang menggunakan layanan GPRS. Adapun jenis layanan GPRS yang digunakan adalah Telkomsel GPRS. Untuk dapat menggunakan layanan tersebut perlu dilakukan pengaturan pada *modem* sebagai berikut:

APN : www.xlgprs.net
 User : xlgprs
 Password : proxl
 Dial no. : *99***1#

Setelah dilakukan pengaturan seperti di atas, maka *client-server* akan mendapatkan IP publik dan dapat saling berkomunikasi. Untuk memastikan hal tersebut, maka dapat digunakan perintah ping seperti gambar berikut.



```

C:\WINDOWS\system32\ping.exe
Reply from 182.0.242.7: bytes=32 time=198ms TTL=111
Reply from 182.0.242.7: bytes=32 time=196ms TTL=111
Reply from 182.0.242.7: bytes=32 time=195ms TTL=111
Reply from 182.0.242.7: bytes=32 time=193ms TTL=111
Reply from 182.0.242.7: bytes=32 time=182ms TTL=111
Reply from 182.0.242.7: bytes=32 time=181ms TTL=111
Reply from 182.0.242.7: bytes=32 time=180ms TTL=111
Reply from 182.0.242.7: bytes=32 time=199ms TTL=111
Reply from 182.0.242.7: bytes=32 time=228ms TTL=111
Reply from 182.0.242.7: bytes=32 time=677ms TTL=111
Reply from 182.0.242.7: bytes=32 time=176ms TTL=111
Reply from 182.0.242.7: bytes=32 time=195ms TTL=111
Reply from 182.0.242.7: bytes=32 time=173ms TTL=111
Reply from 182.0.242.7: bytes=32 time=192ms TTL=111
Reply from 182.0.242.7: bytes=32 time=190ms TTL=111
Reply from 182.0.242.7: bytes=32 time=198ms TTL=111
Reply from 182.0.242.7: bytes=32 time=186ms TTL=111
Reply from 182.0.242.7: bytes=32 time=184ms TTL=111
Reply from 182.0.242.7: bytes=32 time=182ms TTL=111
Reply from 182.0.242.7: bytes=32 time=180ms TTL=111
Reply from 182.0.242.7: bytes=32 time=178ms TTL=111
Reply from 182.0.242.7: bytes=32 time=176ms TTL=111
Reply from 182.0.242.7: bytes=32 time=174ms TTL=111
Reply from 182.0.242.7: bytes=32 time=213ms TTL=111
  
```

Gambar 54. Pengecekan koneksi *client-server*

Pengujian kemampuan transfer data ini dilakukan pada saat malam hari, yaitu pada pukul 00.15 – 02.15 WIB. Hal tersebut bertujuan untuk mendapatkan kualitas layanan yang lebih baik karena trafik penggunaan akses internet pada waktu tersebut tidak begitu padat.

Pengujian dilakukan berdasarkan interval waktu pembacaan sensor terhadap waktu yang diperlukan untuk mengirim *database* menuju *server*. Besar *database* yang dikirimkan adalah sama yaitu 216 bit data yang berisi informasi tiga buah

sensor suhu, tanggal, dan waktu. Berikut adalah tabel data hasil pengamatan yang telah dilakukan.

Tabel 5. Pengamatan dengan interval masukan 2 detik

No	Interval Masukan dari Sensor (detik)	Waktu Input Sensor pada Database PC Client	Waktu Input pada Database PC Server	Waktu yang Dibutuhkan Selama Transfer Data (detik)
1	2	01.57.17	01.57.22	5
2	2	01.57.19	01.57.22	3
3	2	01.57.21	01.57.22	1
4	2	01.57.23	01.57.29	6
5	2	01.57.25	01.57.29	4
6	2	01.57.27	01.57.29	2
7	2	01.57.29	01.57.33	4
8	2	01.57.31	01.57.35	4
9	2	01.57.33	01.57.38	5
10	2	01.57.35	01.57.38	3

Rata-rata waktu pengiriman data : 3.7 detik

Tabel 6. Pengamatan dengan interval masukan 5 detik

No	Interval Masukan dari Sensor (detik)	Waktu Input Sensor pada Database PC Client	Waktu Input pada Database PC Server	Waktu yang Dibutuhkan Selama Transfer Data (detik)
1	5	01.52.54	01.52.58	4
2	5	01.52.59	01.53.05	6
3	5	01.53.04	01.52.06	2
4	5	01.53.09	01.53.12	3
5	5	01.53.14	01.53.18	4
6	5	01.53.19	01.53.24	5
7	5	01.53.24	01.53.26	2
8	5	01.53.29	01.53.33	4
9	5	01.53.34	01.53.38	4
10	5	01.53.39	01.53.42	3

Rata-rata waktu pengiriman data : 3.7 detik

Tabel 7. Pengamatan dengan interval masukan 10 detik

No	Interval Masukan dari Sensor (detik)	Waktu Input Sensor pada Database PC Client	Waktu Input pada Database PC Server	Waktu yang Dibutuhkan Selama Transfer Data (detik)
1	10	01.48.27	01.48.30	3
2	10	01.48.37	01.48.40	3
3	10	01.48.47	01.48.51	4
4	10	01.48.57	01.49.06	9
5	10	01.49.07	01.49.10	3
6	10	01.49.17	01.49.21	4
7	10	01.49.27	01.49.20	3
8	10	01.49.37	01.49.39	2
9	10	01.49.47	01.49.49	2
10	10	01.49.57	01.50.00	3

Rata-rata waktu pengiriman data : 3.6 detik

Tabel 8. Pengamatan dengan interval masukan 30 detik

No	Interval Masukan dari Sensor (detik)	Waktu Input Sensor pada Database PC Client	Waktu Input pada Database PC Server	Waktu yang Dibutuhkan Selama Transfer Data (detik)
1	30	01.05.11	01.05.14	3
2	30	01.05.41	01.05.44	3
3	30	01.06.11	01.06.14	3
4	30	01.06.41	01.06.45	4
5	30	01.07.11	01.07.15	4
6	30	01.07.41	01.07.44	3
7	30	01.08.11	01.08.13	2
8	30	01.08.41	01.08.44	3
9	30	01.09.11	01.09.17	6
10	30	01.09.41	01.09.46	5

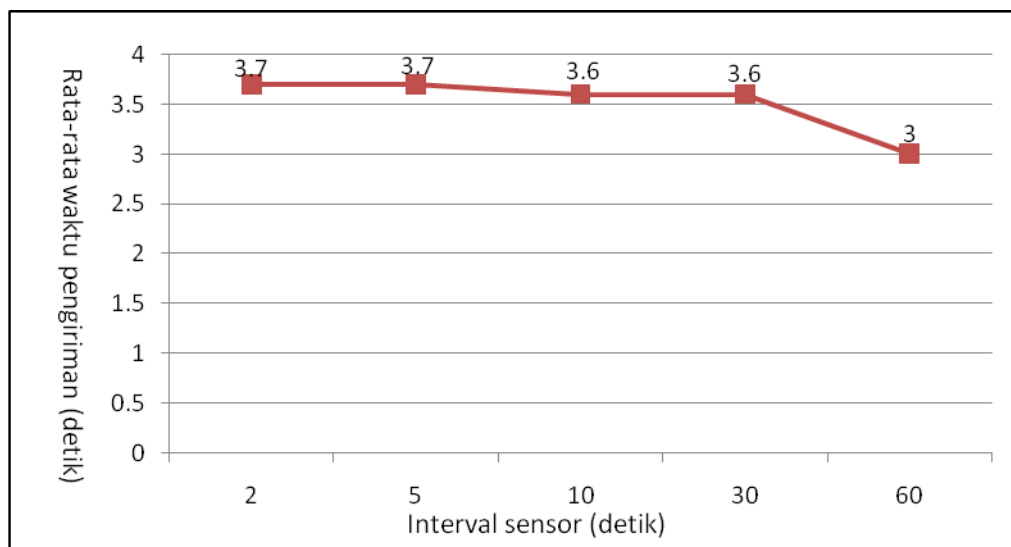
Rata-rata waktu pengiriman data : 3.6 detik

Tabel 9. Pengamatan dengan interval masukan 60 detik

No	Interval Masukan dari Sensor (detik)	Waktu Input Sensor pada Database PC Client	Waktu Input pada Database PC Server	Waktu yang Dibutuhkan Selama Transfer Data (detik)
1	60	02.12.10	02.12.13	3
2	60	02.13.10	02.13.15	5
3	60	02.14.10	02.14.12	2
4	60	02.15.10	02.15.13	3
5	60	02.16.10	02.16.15	5
6	60	02.17.10	02.17.12	2
7	60	02.18.10	02.18.13	3
8	60	02.19.10	01.19.13	3
9	60	02.20.10	02.20.12	2
10	60	02.21.10	02.21.12	2

Rata-rata waktu pengiriman data : 3 detik

Berdasarkan tabel pengamatan di atas, dapat diketahui kebutuhan waktu *transfer* data yang dibutuhkan dalam sistem *monitoring* berbasis GPRS. Waktu yang dibutuhkan berkisar antara 3-4 detik setiap kali pengiriman. Data rata-rata tersebut telah diolah dan ditampilkan pada grafik berikut ini.

Gambar 55. Grafik rata-rata waktu *transfer* data

Kecepatan transfer data tiap saat tidak selalu tetap. Hal ini dikarenakan kesibukan trafik jaringan yang berubah-ubah dan sangat memperngaruhi pengujian ini. Pengujian dilakukan pada saat malam hari dengan tujuan untuk mendapatkan hasil koneksi yang baik. Selain itu, jenis layanan yang digunakan juga sangat berpengaruh, dimana layanan yang digunakan adalah layanan berbasis GPRS dengan kecepatan maksimum 14,4 Kbps.