

BAB III

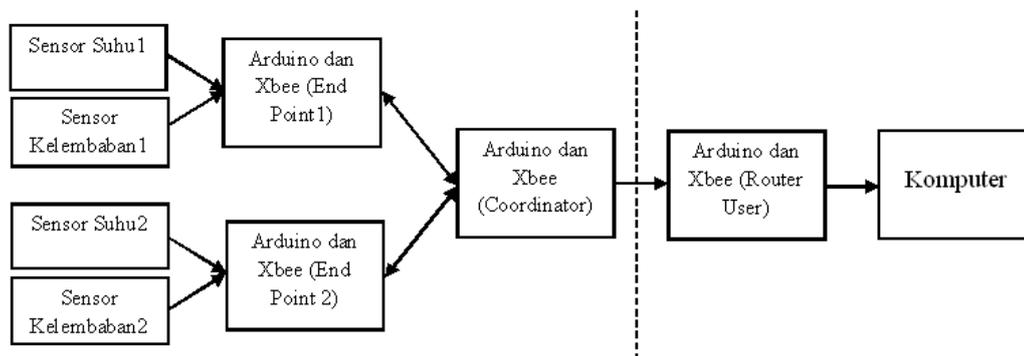
METODE PENELITIAN

Metode penelitian yang digunakan pada pembuatan perangkat keras dan perangkat lunak yaitu dengan studi kepustakaan. Dengan cara ini penulis berusaha untuk mendapatkan dan mengumpulkan data-data, informasi, konsep-konsep yang bersifat teoritis dari buku, bahan-bahan kuliah dan internet yang berkaitan dengan permasalahan.

Dari data-data yang diperoleh maka dilakukan perencanaan rangkaian perangkat keras (*hardware*). Dalam perangkat keras ini, penulis akan melakukan pengujian perangkat keras dengan program-program yang telah dibuat. Kemudian dilanjutkan pembuatan perangkat lunak untuk *monitoring*. Terakhir adalah penggabungan perangkat keras dengan kerja perangkat lunak (*software*) yang telah selesai dibuat.

Pada bab ini dibahas mengenai masalah yang timbul dalam perencanaan dan perancangan perangkat keras (*hardware*) maupun pembuatan perangkat lunak (*Software*). Dari kedua bagian tersebut akan dipadukan / diintegrasikan agar dapat bekerja sama menjalankan sistem dengan baik.

Dalam perancangan sistem *wireless sensor network* untuk pemantauan suhu dan kelembaban tanah pada lahan tanaman jarak, digunakan blok diagram seperti pada Gambar 3.1.



Gambar 3.1 Blok Diagram Sebuah Sistem *Wireless Sensor Network* Untuk

Monitoring Suhu dan Kelembaban Tanah Pada Lahan Tanaman Jarak

Pada tugas akhir ini hanya membahas bagian *router user* dan komputer, Peran *router* disini yaitu menerima data dari *coordinator* melalui xbee kemudian diproses oleh arduino setelah data diproses data dikirim ke komputer. Sedangkan peranan dari komputer adalah mengolah data yang diterima dari *router* kemudian ditampilkan pada layar monitor komputer menggunakan Visual basic 6.0, komunikasi yang digunakan antara *router* dengan komputer menggunakan komunikasi USB *serial*.

3.1. Perancangan Perangkat Keras

3.1.1. Koneksi Arduino dengan Komputer

Perancangan perangkat keras (*hardware*) ini membahas tentang koneksi arduino dan dengan komputer, arduino dapat berkomunikasi dengan komputer melalui USB. Tetapi komputer harus menginstal *driver* pada arduino terlebih

dahulu, untuk mendapatkan *driver* arduino dan *Software* IDE arduino bisa diunduh di www.arduino.cc/.

3.1.2. Koneksi Xbee dengan Komputer

Pada perancangan ini menggunakan Xbee Pro *Series 2*, konfigurasi xbee memiliki peranan masing-masing yang berfungsi supaya antar xbee dapat berkomunikasi dengan baik pada alamat yang ditujukan dan sesuai alamat yang kita inginkan. Sebelum mengkonfigurasi xbee harus sudah terinstal *driver adapter* xbee dan *Software* bernama X-CTU.



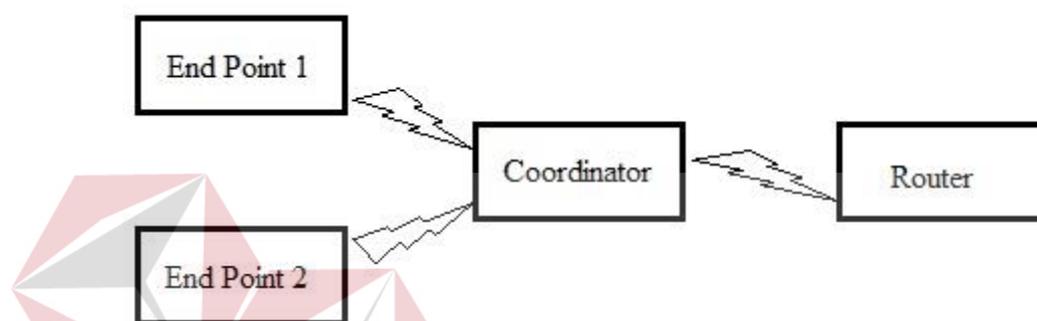
Gambar 3.2 Xbee dan Adapter

3.2. Perancangan Perangkat Lunak

Selain *hardware* yang diperlukan pada perancangan dan pembuatan sistem *monitoring* suhu dan kelembaban tanah pada lahan tanaman jarak dengan memanfaatkan *wireless sensor network* sebagai komunikasi data antar *node* dan usb untuk komunikasi dengan komputer, juga diperlukan *Software* / program pada Arduino dan komputer untuk dapat saling mendukung. Perancangan perangkat lunak meliputi algoritma dan program baik pada Arduino maupun komputer.

3.2.1. Program Xbee sebagai Router

Sebelum menentukan program masing-masing xbee harus menentukan topologi jaringan, sehingga lebih mudah untuk proses konfigurasi xbee pada peranan masing-masing dalam *routing* atau pengalamatan yang ditujukan. Untuk pilihan topologi menggunakan topologi *tree*, karena dibutuhkan untuk komunikasi *point to point* dan *point to multipoint*.

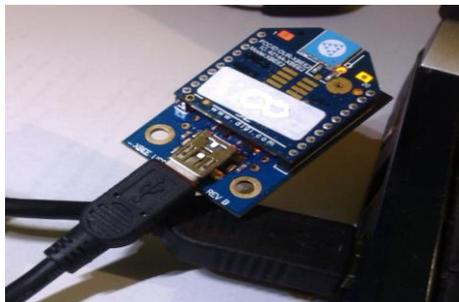


Gambar 3.3 Topologi

Pada Gambar 3.3 *node End point 1* maupun *node End point 2* dapat menerima dan mengirim data tetapi hanya komunikasi *point to point*. Sedangkan pada *node coordinator* dapat menerima sekaligus mengirim data dan berkomunikasi *point to multipoint* pada *node* yang terhubung langsung yang disebut dengan *broadcast*. Untuk komunikasi antara *node coordinator* dengan *node router*, *node coordinator* dapat mengirim data dan tidak dapat menerima data dari *node router*. Begitu juga untuk *node router* tidak dapat mengirim data pada *node coordinator* dan hanya dapat menerima data dari *node coordinator* saja, perlu diketahui sekali lagi bahwa tugas akhir ini hanya membahas pada sisi *router user* saja dan konfigurasinya sebagai berikut:

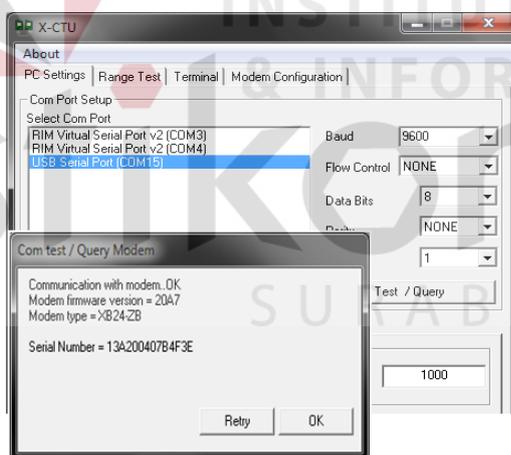
Konfigurasi Xbee sebagai *Router*

Untuk mengkonfigurasi xbee, pasang xbee sebagai *router* pada *adapter* dan hubungkan kabel USB pada komputer.



Gambar 3.4 Xbee Beserta *Adapter*

Kemudian buka *Software X-CTU* yang sudah diinstal sebelumnya, dan pada tab *PC Setting port* pada *adapter* adalah *port com15*. Lalu klik pada *Test/Query* dan tunggu akan muncul seperti pada Gambar 3.13 dan klik *ok*.

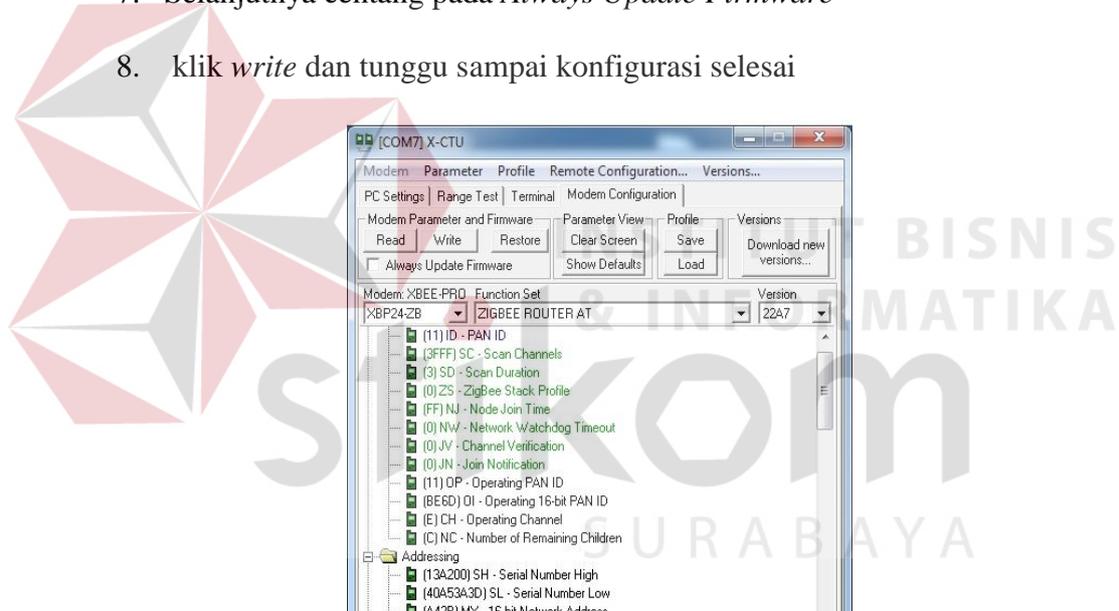


Gambar 3.4 *Test Query* Xbee

Pada Gambar 3.4 menunjukkan informasi *serial number* dan konfigurasi *firmware* yang sudah terisi sebelumnya. Langkah selanjutnya adalah sebagai berikut:

1. Pilih tab *Modem Configurasi*
2. Klik *Read*

3. Pilih model *modem* xbee yaitu XB24-ZB
4. Pilih *Function Set* atau *firmware* xbee sebagai *router* yaitu ZIGBEE ROUTER AT
5. Isi PAN ID dengan nilai 11, maksudnya parameter xbee tersebut hanya berkomunikasi dengan xbee lainnya yang memiliki PAN ID 11 saja.
6. Kemudian tentukan *destination* untuk tujuan transfer data yang diinginkan, tetapi *router* ini untuk komunikasi *point to point* sehingga DH=FFFF DL=0
7. Selanjutnya centang pada *Always Update Firmware*
8. klik *write* dan tunggu sampai konfigurasi selesai

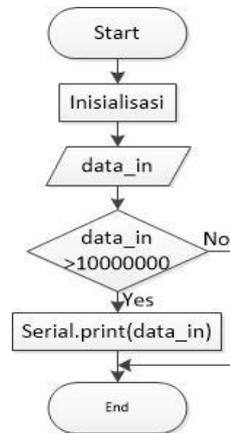


Gambar 3.5 Konfigurasi Xbee

3.2.1. Program Arduino Sebagai Router

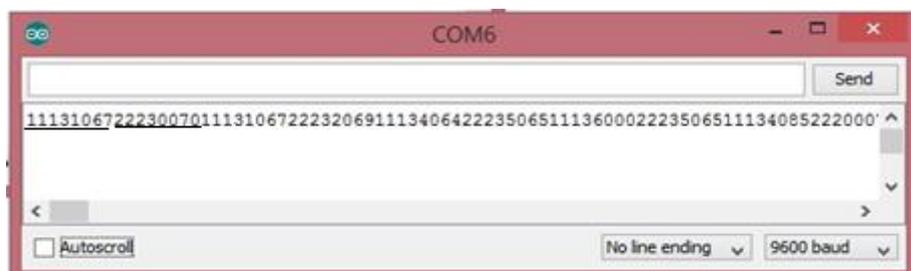
Cara kerja *router* yaitu menerima data dari *coordinator* setelah data diterima, data langsung di proses oleh arduino jika data sesuai yang di tentukan oleh program maka data kemudian dikirim ke komputer. pengiriman data antara

router dengan komputer menggunakan kabel usb. Gambar 3.6 merupakan *flowchart* program pada arduino pada sisi *router user*.



Gambar 3.6 *Flowchart* Pengolahan Data Pada *Router*

Dari penjelasan Gambar 3.6 langkah pertama program menginisialisasi data “data_in” bertipe data *unsigned long*, setelah proses inisialisasi selesai data diterima kemudian data diseleksi apakah data tersebut >10000000 jika benar data akan dikirim langsung ke komputer, jika data yang diterima *router* < 10000000 data di abaikan karena data tersebut bukan dari *coordinator* yang mana data yang dikirim oleh *coordinator* sebesar 8 digit. Pada tugas akhir *end point* hanya menggunakan 2 *node* maka *router user* hanya menerima data yang nilainya lebih dari 10000000 dan 20000000.



Gambar 3.7 Data Yang Dikirim Ke Komputer

Pada Gambar 3.7 merupakan hasil *capture* data yang akan dikirim ke komputer. Data yang dikirim ke komputer sebanyak 8 digit misal pada hasil *capture* Gambar 3.7 yaitu 1113106722230070111310672. . . artinya data yang dikirim adalah 11131067, 22230070 kemudian 11131067 dan seterusnya.

3.2.2. Program Antarmuka pada Komputer

perancangan antarmuka pada komputer untuk *monitoring* ini menggunakan visual basic 6.0. Visual basic mempunyai program dengan aplikasi *grafichal user interface* (GUI) yang dapat berkomunikasi langsung dengan komputer. Perancangan program visual basic ini terdiri dari tampilan aplikasi, diagram alir, properti yang digunakan yang terakhir adalah program

a. Tampilan perancangan aplikasi

1. Tampilan aplikasi *monitoring*



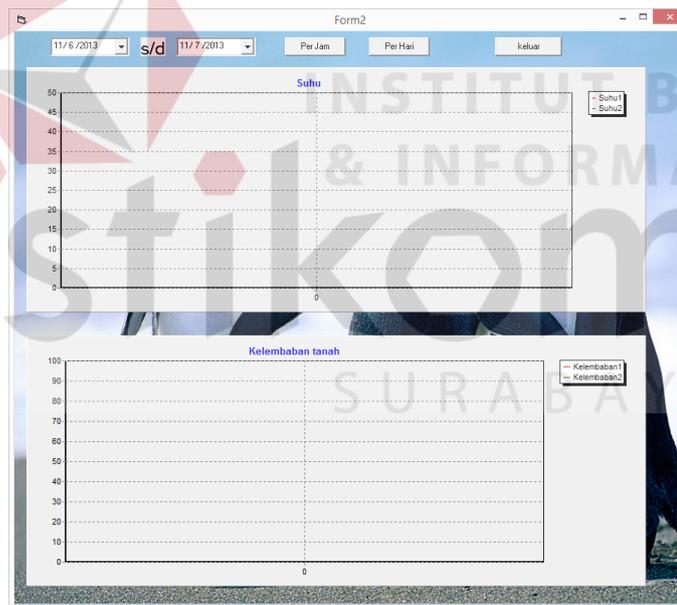
Gambar 3.8 Aplikasi *Monitoring*

Tampilan aplikasi *monitoring* pada Gambar 3.8 tersebut diatas dibuat menggunakan visual basic berikut penjelasan dari tampilan aplikasi.

Tabel 3.1 Keterangan Pada Aplikasi *Monitoring*

No.	Keterangan
1	Sebagai pemberitahuan apabila warna merah <i>node</i> 1 atau 2 tidak mengirimkn data
2	Sebagai pemberitahuan apabila berwarna merah melebihi atau kurang dari batas yang telah ditentukan
3	menampilkan angka pembacaan dari <i>node</i> 1 dan <i>node</i> 2
4	Menampilkan grafik dari pembacaan <i>node</i> 1 berwarna merah dan <i>node</i> 2 berwarna hijau
5	Menampilkan waktu saat ini
6	Untuk menampilkan <i>history</i>

2. Tampilan aplikasi pada *history*

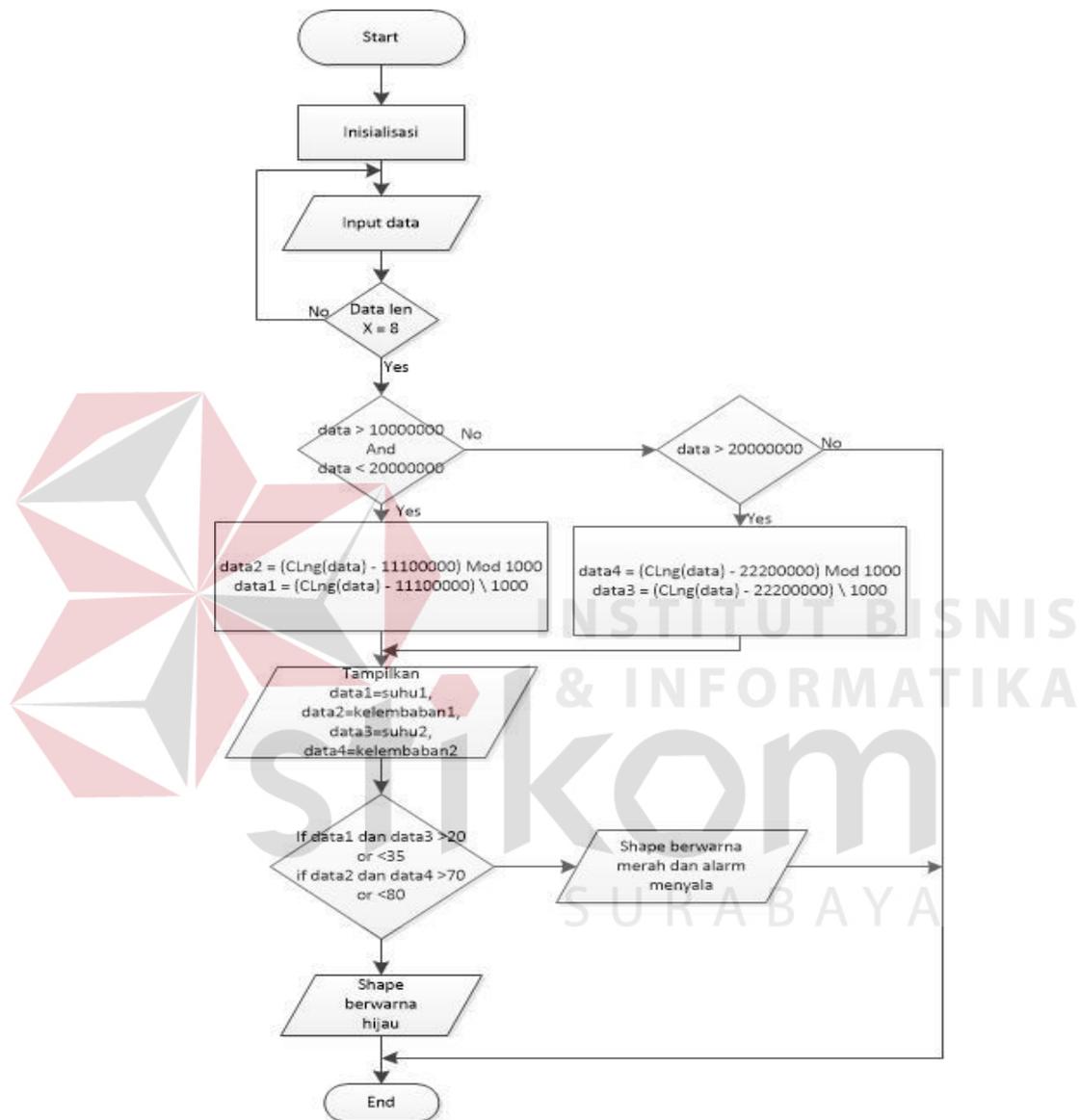


Gambar 3.9 Tampilan Aplikasi *History*

Tampilan aplikasi pada *history* ini untuk menampilkan hasil rata-rata *record* dari aplikasi utama. Untuk dapat menampilkan grafik *history* pada aplikasi tersebut langkah pertama yaitu menentukan tanggal, bulan, dan tahun kemudian klik

tombol per jam atau per hari untuk melihat tampilan rata-rata perjam atau perhari.
Tombol keluar untuk keluar pada tampilan *history*.

b. Diagram alir pemrograman visual basic



Gambar 3.10 Diagram Alir Program Aplikasi Untuk *Monitoring* Pada Vb 6.0

Pada aplikasi utama ini berfungsi untuk menampilkan data yang diterima oleh *router*. Proses ini dimulai ketika *user* mengaktifkan aplikasi, proses pertama yang dilakukan oleh program aplikasi tersebut adalah membuka *port serial* / *USB* dan menginisialisasi. inisialisasi “data, log, loghari” bertipe data *string*, “data1,

data2, data3, data4” bertipe data *Long*. “i, x, a, b, c, d, z, y” bertipe data *Integer*. Setelah proses inialisasi selesai data diproses oleh program visual basic berjumlah 8 digit jika data masih kurang dari 8 digit maka akan menunggu hingga data sebesar 8 digit jika memenuhi syarat data diproses ke langkah selanjutnya. Penjelasan Gambar 3.10 pada $data2 = (CLng(data) - 11100000) \text{ Mod } 1000$, $data1 = (CLng(data) - 11100000) \setminus 1000$ maksud dari CLng adalah data yang diterima di *convert* dahulu ke dalam tipe data *long* agar dapat menampung nilai bulat dengan nilai yang lebih panjang. Berikut adalah contoh penghitungan jika data yang diterima adalah 11130067 :

$$((11130067) - 11100000) \text{ Mod } 1000$$

$$((11130067) - 11100000) \setminus 1000$$

Maka :

$$(30067) \text{ Mod } 1000 = 30 \text{ sisa } 67$$

$$(30067) \setminus 1000 = 30,067$$

Maksud mod pada operator aritmatika dan logika adalah merupakan hasil dari sisa pembagian sedangkan operator \setminus adalah hasil bulat dari pembagian. Setelah melakukan perhitungan dan mengetahui fungsi operator dapat diketahui hasil penghitungan dari mod (*modulus*) adalah 67 sedangkan penghitungan dari operator / adalah 30. 3 angka didepan menunjukkan data yang diterima hasil pengiriman dari *Node1/end point1* misal data yang diterima 11130067 maka data tersebut adalah hasil pengiriman dari *Node1* yang dikirim melalui *coordinator*. Berikut adalah keterangan hasil proses data jika data yang diterima adalah 11130067 :

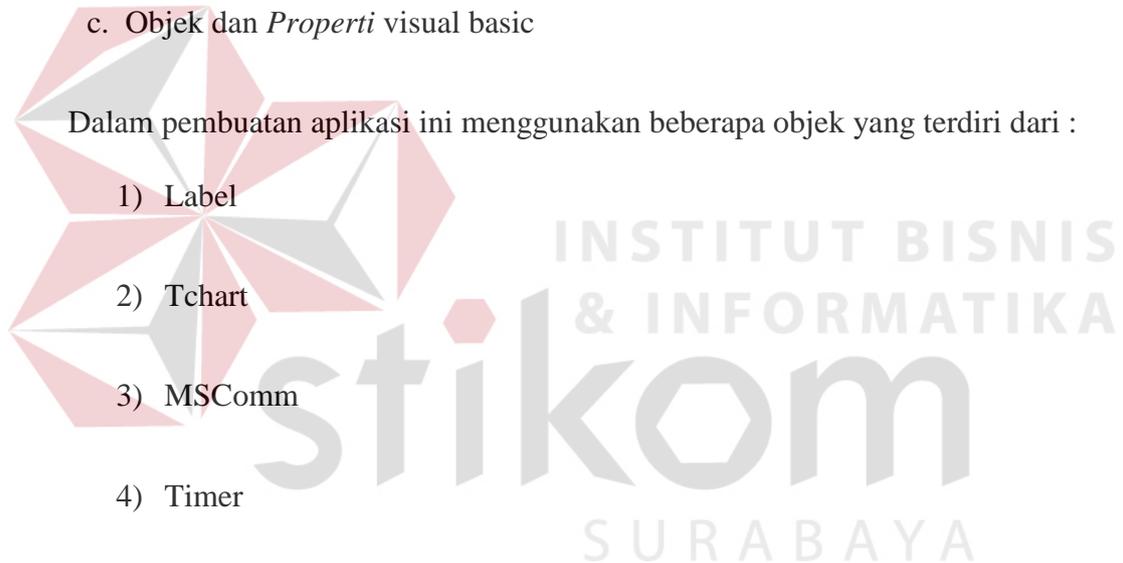
- a) Angka 111 pada 3 digit pertama menunjukkan bahwa data tersebut adalah pengiriman dari *Node1*.
- b) Angka 30 pada 2 digit selanjutnya menunjukkan nilai pembacaan dari sensor suhu adalah 30 °C.
- c) Angka 067 pada 3 digit terakhir menunjukkan nilai pembacaan dari sensor kelembaban tanah adalah 67 %.

Setelah data di pisah-pisahkan data di tampilkan pada aplikasi penampil visual basic 6.0.

c. Objek dan *Properti* visual basic

Dalam pembuatan aplikasi ini menggunakan beberapa objek yang terdiri dari :

- 1) Label
- 2) Tchart
- 3) MSComm
- 4) Timer
- 5) Shape
- 6) Frame
- 7) Text
- 8) Command Button
- 9) DTPicker



MSComm (Microsoft Communication Control) digunakan untuk melakukan komunikasi pertukaran data dengan arduino melalui *port serial*. Beberapa properti yang dapat di kontrol pada MSComm ini adalah sebagai berikut :

- 1) *CommPort*, yaitu nomor *port* yang dimiliki oleh *serial port* yang akan berkomunikasi
- 2) *Handshaking*, yaitu metode pertukaran data yang dipakai. Terdapat empat metode *Handshaking* yaitu: 0 (*comNone*), 1 (*comXOnXoff*), 2 (*comRTS*), 3 (*comRTSXOnXoff*).
- 3) *RThreshold*, bernilai *integer* (bilangan bulat), berfungsi untuk mengaktifkan *OnComm* event (even terjadi pada kontrol *MsComm* jika terjadi perubahan ukuran buffer data). *RThreshold* untuk mendeteksi perubahan buffer pada data masuk (*receive*).
- 4) *SThreshold*, seperti *RThreshold* hanya fungsinya untuk mengaktifkan *OnComm* event data yang keluar dari komputer.
- 5) *Settings*, properti untuk mengeset bit rate, parity, data bits dan stop bit.
`MSComm1.Setting = "9600,n,8,1"`
- 6) *Input*, merupakan properti yang menampung data yang masuk dari luar.
- 7) *Output*, merupakan properti yang dapat diset sebagai data yang akan dikirimkan keluar.

d. Pembuatan Program Visual Basic

Pembuatan program visual basic didasarkan pada diagram alir dan tampilan yang sudah direncanakan (dapat dilihat pada gambar 3.16). Berikut ini adalah cuplikan listing program aplikasi untuk *monitoring* :

1) Prosedur untuk mengaktifkan *port serial*

```
Private Sub Form_Load()
    MSComm1.PortOpen = True
End Sub
```

“MSComm1.PortOpen = True” digunakan untuk mengaktifkan atau membuka *port serial* yang telah dipilih pada properti Mscomm

2) Prosedur untuk menonaktifkan *port serial*

```
Private Sub Form_Unload(Cancel As Integer)
    MSComm1.PortOpen = False
End Sub
```

“MSComm1.PortOpen = False” digunakan untuk menonaktifkan atau menutup kembali *port serial* yang digunakan

3) Prosedur untuk menerima data

```
Private Sub MSComm1_OnComm()
    If MSComm1.CommEvent = 2 Then
        data = MSComm1.Input
        x = Len(data)
        If x = 8
```

Cuplikan pada program diatas digunakan untuk merima data dari *router user* dengan panjang 8 digit

4) Prosedur untuk menampilkan kedalam grafik

```
Call TChart1.Series(0).AddXY(i, CDBl(data1), "",
    clTeeColor)
```

Grafik digunakan untuk menampilkan perubahan data yang diterima dan digunakan untuk menampilkan grafik dari aplikasi pada *form history*.

5) Prosedur untuk menyimpan dan menambahkan data yang diterima

```
Open log For Append As #1
Write #1, Day(Now), Month(Now), Year(Now), Hour(Now),
Minute(Now), Second(Now), data1, data2, data3, data4
```

“Append” digunakan untuk menyimpan data yang di terima dan juga dapat menambahkan data pada *log*.

6) Prosedur mengaktifkan *beep*

```
Private Sub Timer6_Timer()
    Call AvirasBeep
End Sub
```

Beep digunakan sebagai *alarm* untuk pemberitahuan atau peringatan kepada *user* apabila *error* pada setiap *node* dan jika masing-masing sensor mengirim data yang melebihi batas minimum dan maksimum yang telah ditentukan.

7) Prosedur mengaktifkan tombol *history*

```
Private Sub Command1_Click()
    Form2.Show
End Sub
```

Tombol *history* digunakan untuk menampilkan aplikasi pada *form histor*

