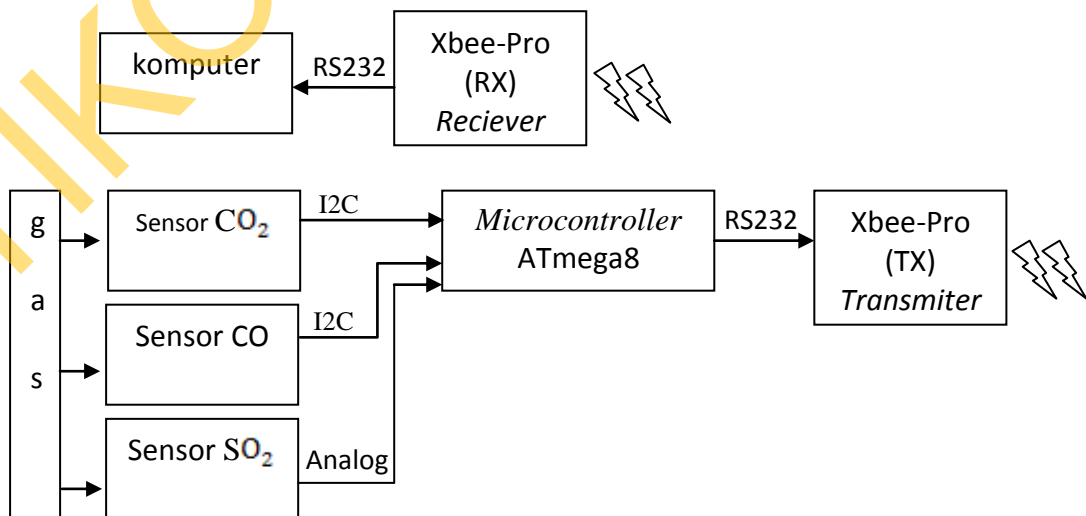


BAB III

METODE PENELITIAN

Metode penelitian yang digunakan meliputi studi kepustakaan dan penelitian laboratorium. Studi kepustakaan dilakukan untuk mencari teori atau informasi dari buku, jurnal, *datasheet*, dan artikel-artikel yang berkaitan dengan permasalahan. Antara lain, *carbon monoxide sensor*, *carbon dioxide sensor*, *sulfur Dioxide sensor*, *microcontroller ATmega8*, modul komunikasi *wireless XBee-PRO*, LCD (*Liquid Crystal Display*), komunikasi *serial asynchronous* (UART), dan komunikasi *serial synchronous* (I2C).

Dari teori atau informasi dari studi kepustakaan yang diperoleh maka dilakukan penelitian laboratorium, yaitu perancangan perangkat keras dan perangkat lunak seperti pada Gambar 3.1. Selain itu, setelah sistem jadi, sistem harus dikalibrasi dengan menggunakan metode yang digunakan oleh Gao Daqi. Bila sistem telah terkalibrasi dengan baik, baru kemudian sistem akan diuji secara keseluruhan dan diimplementasikan.



Gambar 3.1 Diagram Blok Sistem Keseluruhan

Dari blok diagram pada Gambar 3.1 dapat dilihat diagram blok dari system keseluruhan, di mana *input* sistem ini adalah gas odor. *Array sensor* akan mendeteksi apakah gas tersebut mengandung polutan seperti CO, CO₂ dan SO₂. Hasil yang didapatkan array sensor akan dikirimkan ke *microcontroller* menggunakan komunikasi I2C untuk sensor CO dan CO₂ dan analog untuk sensor SO₂ untuk kemudian diproses oleh *microcontroller* Atmega8. Data sensor yang telah diproses akan dikirimkan melalui komunikasi RS232 ke XBee-Pro *transmitter* (Tx). Xbee-Pro Tx mengirimkan informasi ini ke Xbee-Pro *receiver* (Rx) yang terhubung pada komputer secara *wireless*. Di komputer, informasi tersebut akan di sajikan dan disimpan dengan aplikasi yang telah dibuat menggunakan Microsoft Visual Basic 6.0.

3.1 Perancangan Perangkat Keras

3.1.1 Perancangan Wadah Sensor Gas

Alat yang penulis rancang terdiri atas kotak plastik yang digunakan untuk menaruh semua komponen yang digunakan untuk menjalankan sensor. Berikut spesifikasi bahan dan ukuran alat.

Ukuran

Ukuran kotak : 15 cm x 15 cm x 5 cm.

Struktur Material

Bahan Material yang digunakan :

Bagian Rangka

1. Fiber plastik.
2. Mur dan Baut.

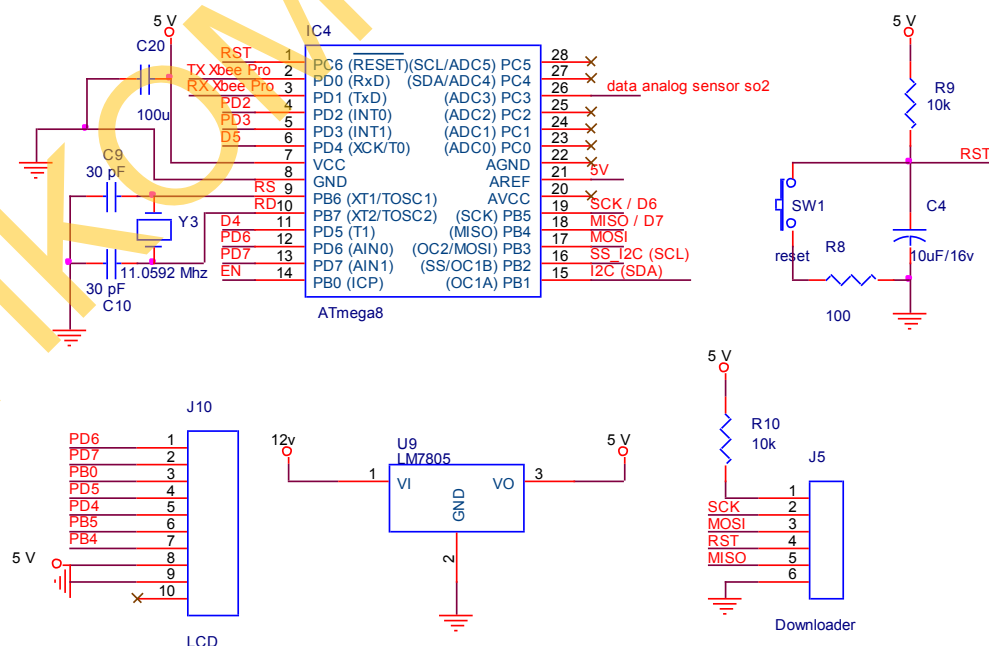
3.1.2 Perancangan Minimum Sistem

Rangkaian minimum sistem dibuat untuk mendukung kerja dari *microchip* ATmega8 dimana *microchip* tidak bisa berdiri sendiri melainkan harus ada rangkaian dan komponen pendukung seperti halnya rangkaian catu daya, kristal dan lain sebagainya yang biasanya disebut minimum sistem.

Microchip berfungsi sebagai otak dalam mengolah semua instruksi baik *input* maupun *output* seperti halnya pemroses data *input* dari *array sensor* yang kemudian menghasilkan *output* yang berupa data yang digunakan sebagai informasi pencemaran udara.

Minimum sistem yang dirancang dalam Tugas Akhir kali ini menggunakan minimum sistem *microcontroller* ATmega8. *Microcontroller* ATmega8 digunakan karena pada perancangan Tugas Akhir ini tidak memerlukan banyak pin I/O.

3.1.2.1 Minimum Sistem Atmega 8.



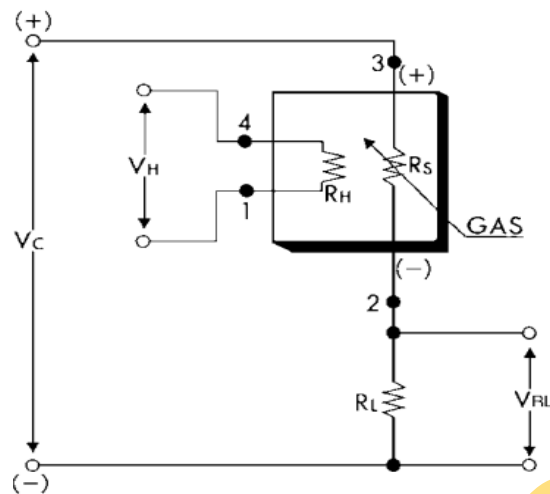
Skematik minimum sistem ATmega8 pada Gambar 3.2 terdiri dari *microchip* ATmega8, rangkaian *oscillator*, rangkaian reset, rangkaian catu daya, rangkaian konektor untuk *downloader* program, serta pin-pin konektor untuk komunikasi I2C, komunikasi serial, dan LCD. Tabel 3.1 merupakan daftar pin I/O yang digunakan dalam Tugas Akhir ini.

Tabel 3.1 Pin I/O Minimum Sistem

Pin I/O	Fungsi
Vcc	Power 5v
Gnd	Ground
PD0/RxD	Sebagai pin Tx Xbee-Pro
PD1/TxD	Sebagai pin Rx Xbee-Pro
PB2 (SS/OC1B)	Terhubung secara paralel pada jalur SCL (<i>clock</i>) dari sensor CO dan SO ₂
PB1(OC1A)	Terhubung secara paralel pada jalur SDA (data) dari sensor CO dan SO ₂
Reset	Mereset program
PD (bit 6 dan 7) (bit 5 dan 4)	Untuk RS dan RD (LCD) Untuk D4 dan D5
PB (bit 0, 5, 4)	Untuk EN , D6 , D7 (LCD)
PC3 (ADC3)	Untuk data analog dari sensor SO ₂

3.1.3 Perancangan Sensor TGS-2602

Untuk sensor SO₂ digunakan sensor TGS dari *Figaro* yaitu TGS 2602. Sesuai dengan datasheet, karakteristik gas yang dapat terdeteksi yaitu gas-gas yang mengandung *sulfur dioxide* SO₂ dan *hydrogen sulfide*. Dan untuk mendapatkan hasil output yang lebih optimal, sensor dirancang sesuai dengan rangkaian yang terdapat pada *datasheet* seperti terlihat pada Gambar 3.2.



Gambar 3.3 Rangkaian Sensor TGS.

Agar bisa bekerja dengan baik sensor ini membutuhkan dua tegangan masukan. *Heater voltage* (V_H) digunakan sebagai tegangan *heater* dan *circuit voltage* (V_C) merupakan catu daya rangkaian, keduanya diberikan tegangan sebesar 5 volt DC. Nilai resistor beban (R_L) dapat dipilih atau diatur untuk mengoptimalkan nilai *alarm threshold*, menjaga *power dissipation* (P_S) semikonduktor di bawah batas 15mW. *Power dissipation* (P_S) akan menjadi sangat tinggi ketika nilai R_S adalah sama dengan nilai R_L . Nilai *power dissipation* (P_S) dan hambatan sensor (R_S) dapat dihitung dengan rumus sebagai berikut :

Rumus untuk menghitung nilai *Power Dissipation* (P_S) :

$$P_S = \frac{(V_C - V_{RL})^2}{R_S} \quad (3.1)$$

Rumus untuk menghitung nilai hambatan sensor (R_S) :

$$R_S = \frac{V_C - V_{RL}}{V_{RL}} \times R_L \quad (3.2)$$

3.1.4 Modul Xbee-Pro TX & RX

XBee merupakan suatu modul yang didesain untuk memenuhi standar zigbee/ IEEE 802.15.4 yang biasa digunakan untuk aplikasi jaringan sensor yang berbiaya dan berdaya rendah. Modul ini membutuhkan daya yang rendah dan menyediakan transfer data yang handal antara dua *device*. Modul ini mempunyai dimensi fisik kecil sehingga praktis dalam penempatan. Modul ini beroperasi pada rentang frekuensi 2.4 GHz.

3.1.4.1 Driver Modul Xbee-Pro

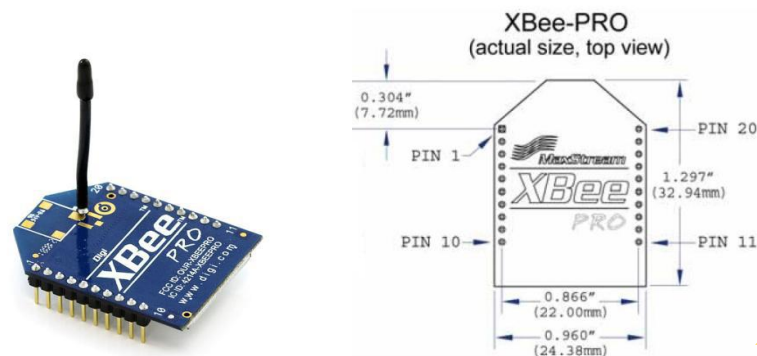
Untuk mengakses modul Xbee-Pro, diperlukan sebuah *driver* untuk modul Xbee-Pro yang mana modul Xbee-Pro ini hanya memiliki tegangan catu daya rendah yaitu antara 2,8 volt sampai 3,4 volt, sehingga diperlukan regulator tegangan sebesar 3.3 volt, namun untuk data *interface* dapat dihubungkan secara langsung ke *microcontroller*. Berikut adalah hubungan antara modul pin Xbee-Pro dan ATmega8 ditunjukkan pada Tabel 3.12.

Tabel 3.12. Hubungan antara modul pin pada Xbee-Pro & *microcontroller* ATmega8

ATmega8		Xbee-Pro	
Pin	Nama	Pin	Nama
10	VCC	1	VCC
15	Tx(PD1)	2	Dout (Tx)
14	Rx (PD0)	3	Din (Rx)
11	GND	10	GND

3.1.4.2 Konfigurasi Pin Xbee-Pro

Gambar 3.3 adalah gambar & konfigurasi pin Xbee-Pro dan penjelasannya, sedangkan untuk penjelasan fungsi-fungsi setiap pin pada Xbee-Pro dapat dilihat pada Tabel 3.13, dan untuk spesifikasi Xbee-Pro dapat dilihat pada Tabel 3.14.



Gambar 3.4 Modul Xbee-Pro & Dimensi Xbee-Pro

Tabel 3.13 Konfigurasi pin Xbee-Pro

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	CD / DOUT_EN / DO8	Output	Carrier Detect, TX_enable or Digital Output 8
5	RESET	Input	Module Reset
6	PWM0 / RSSI	Output	PWM Output 0 or RX Signal Strength Indicator
7	[reserved]	-	Do not connect
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4 / RF_TX	Either	Analog Input 4, Digital I/O 4 or Transmission Indicator
12	DIO7 / CTS	Either	Digital I/O 7 or Clear-to-Send Flow Control
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	AD5 / DIO5 / Associate	Either	Analog Input 5, Digital I/O 5 or Associated Indicator
16	AD6 / DIO6 / RTS	Either	Analog Input 6, Digital I/O 6 or Request-to-Send Flow Control
17	AD3 / DIO3 / COORD_SEL	Either	Analog Input 3, Digital I/O 3 or Coordinator Select
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

Tabel 3.14 Spesifikasi Xbee-Pro

Specification	XBee-PRO
Performance	
Indoor/Urban Range	up to 300 ft. (100 m)
Outdoor RF line-of-sight Range	up to 4000 ft. (1200 m)
Transmit Power Output	60 mW (18 dBm), 100 mW EIRP
RF Data Rate	250,000 bps
Receiver Sensitivity	-100 dBm (1% PER)
Power Requirements	
Supply Voltage	2.8 – 3.4 V
Transmit Current (typical)	270 mA (@ 3.3 V)
Receive Current (typical)	55 mA (@ 3.3 V)
Power-down Current	< 10 μ A
General	
Operating Frequency	ISM 2.4 GHz
Dimensions	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)
Antenna Options	U.FL Connector, Chip Antenna or Wire Antenna
Networking & Security	
Supported Network Topologies	Point-to-Point, Point-to-Multipoint, Peer-to-Peer and Mesh
Number of Channels	16 Direct Sequence Channels (software selectable)
Network Layers	PAN ID and Source/Destination Addressing
Agency Approvals	
FCC Part 15.247	pending
Industry Canada (IC)	pending
Europe	pending

3.2 Perancangan Perangkat Lunak

Modul sensor MG-811 dan MQ-7 memiliki antarmuka UART TTL dan I2C yang dapat digunakan untuk menerima perintah atau mengirim data, dan untuk sensor TGS-2602 menggunakan pin ADC sebagai jalur data antara sensor dan *microcontroller*.

3.2.1 Antarmuka UART TTL

Antarmuka UART TTL selain digunakan untuk komunikasi serial juga digunakan untuk mengubah alamat I2C pada pengerjaan Tugas Akhir kali ini. Karena penulis menggunakan dua modul sensor maka alamat dari tiap sensor harus dibedakan untuk bisa berkomunikasi, berikut parameter yang digunakan untuk mengubah alamat I2C :

1. *Baudrate* 38400 bps
2. 8 bit data
3. 1 bit stop
4. Tanpa bit *parity*
5. Tanpa *flow control*

untuk mengubah alamat I2C pin TX *microcontroller* harus terhubung ke pin RX modul gas sensor dan pin RX *microcontroller* terhubung dengan pin TX dari modul gas sensor. Berikut adalah *source code* yang dituliskan ke *microcontroller* untuk mengubah alamat I2C.

```
putchar(0x53); // perintah SET I2C ADDRESS
putchar(0xAA); // parameter
putchar(0x55); // parameter
putchar(newAddress); // alamat baru 0xE2
```

Contoh kode program dengan antarmuka UART untuk mengganti alamat I2C dari 0xE0 menjadi 0xE2. Data alamat I2C disimpan di EEPROM sehingga tidak akan hilang saat *power off*. Alamat I2C disajikan dalam Tabel 3.15.

Tabel 3.15 Alamat I2C

Alamat I2c	
Alamat tulis I2C	Alamat baca I2C
0xE0	0xE1
0xE2	0xE3
0xE4	0xE5
0xE6	0xE7
0xE8	0xE9
0xEA	0xEB
0xEC	0xED
0xEE	0xEF

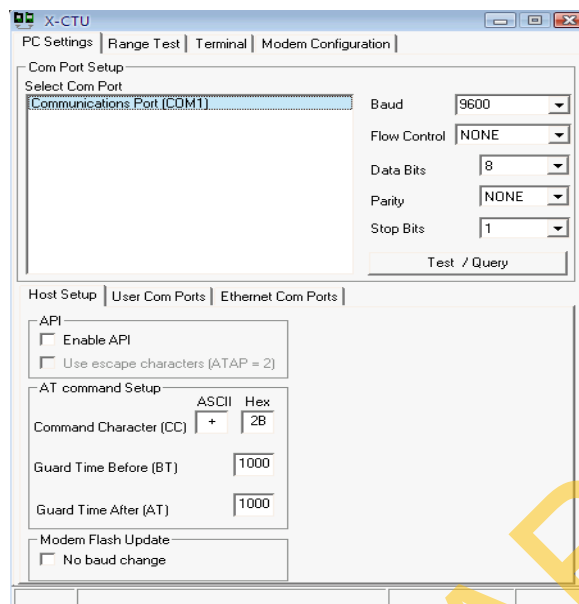
3.2.2 Antarmuka I2C Sensor MG-811 dan MQ-7 dan ADC Sensor TGS

2602

Modul sensor MG-811 dan MQ-7 memiliki antarmuka I2C. Pada antarmuka I2C ini, modul MG-811 dan MQ-7 bertindak sebagai *slave* dengan alamat *default* adalah 0xE0 dan dapat diganti menggunakan perintah yang dijelaskan menggunakan jalur komunikasi UART. Pin I2C adalah pin SDA dan SCL yang pada ATmega8 hanya ada dua pin saja, dengan adanya 2 sensor yang digunakan, untuk itu pin SDA dan SCL di *parallel* menjadi 2 pin dengan membedakan alamat I2C dari setiap sensor-sensor sehingga dapat digunakan untuk 2 sensor tersebut. Untuk pin ADC digunakan untuk data analog dari sensor So2 yaitu TGS 2602, karena pin adc dan pin I2C berada di port C maka itu menyebabkan konflik jika di gunakan secara bersamaan, oleh karena itu pin I2C dipindah ke port B menggunakan pin PB1 untuk SDA dan PB2 untuk SCL.

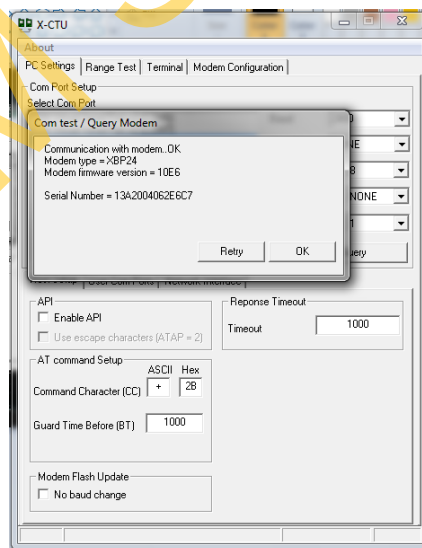
3.2.3 Konfigurasi Parameter Xbee-Pro Tx & Rx

Untuk melakukan konfigurasi parameter modul Xbee-Pro dapat melalui sebuah *software* bawaan Xbee-PRO yaitu X-CTU. Agar Xbee- PRO dapat melakukan komunikasi *point to point* atau komunikasi secara dua titik yaitu pengirim dan penerima saja adalah dengan melakukan konfigurasi alamat dari masing-masing modul. Untuk masuk ke mode konfigurasi pada XCTU, Xbee-PRO harus dihubungkan dengan komputer melalui kabel USB. Setelah terhubung dan *communications port* muncul, buka software XCTU yang merupakan bawaan dari XBee-PRO dengan terlebih dahulu mengatur *baudrate default* sebesar 9600, kemudian klik Test/Query. Seperti ditunjukkan pada Gambar 3.5.



Gambar 3.5 Tampilan untuk setting konfigurasi parameter pada X-CTU

Jika Xbee-PRO berhasil terhubung dengan *software* X-CTU, maka akan keluar jendela baru yang menunjukkan keterangan tipe, *firmware* dan *Serial Number* Xbee-PRO. Seperti ditunjukkan pada Gambar 3.6.



Gambar 3.6 Informasi Xbee-Pro setelah berhasil terhubung dengan XCTU

Setelah Xbee-Pro terhubung dengan *software* XCTU, dilanjutkan melakukan setting konfigurasi parameter Xbee-Pro di terminal XCTU agar Xbee-

Pro di sisi Tx & Rx dapat berkomunikasi dengan baik. Terdapat beberapa perintah untuk mensetting Xbee-Pro. Berikut perintah yang diperlukan untuk mengatur parameter Xbee-Pro dengan menggunakan *software* X-CTU yang di dapat dari *datasheet Software X-CTU* adalah sebagai berikut.

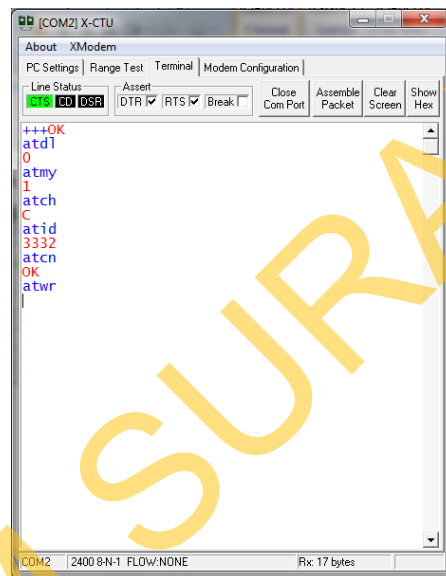
- 1 “+++” merupakan perintah untuk memastikan Xbee-Pro siap diatur atau tidak dan juga untuk mengawali setting parameter pada Xbee-Pro.
- 2 “AT” (AT Command) merupakan perintah awalan penulisan perintah pada Xbee-Pro.
- 3 “DL” (Destination Address Low) merupakan perintah untuk mengatur alamat yang akan dituju oleh Xbee-Pro.
- 4 “MY” (Source Address) merupakan perintah untuk mengatur alamat dari Xbee-Pro (alamat diri sendiri), nilai dari “DL” dan “MY” tidak boleh sama.
- 5 “CH” (Chanel) merupakan perintah set/read dari Xbee-Pro dimana nilai awalnya adalah C dan nilainya harus sama untuk Rx dan Tx.
- 6 “ID” (Networking {Addressing}) merupakan perintah pengalamatan PAN (Personal Area Network) dimana nilainya harus sama untuk satu jaringan.
- 7 “WR”(Write) merupakan perintah penulisan pada Xbee-Pro, apakah Xbee-Pro siap untuk mengirimkan data.
- 8 “CN” (Exit Command Mode) merupakan perintah keluar dari ATCommand

Agar 2 buah XBee-Pro dapat saling berkomunikasi, maka XBee tersebut harus mempunyai kesamaan sebagai berikut:

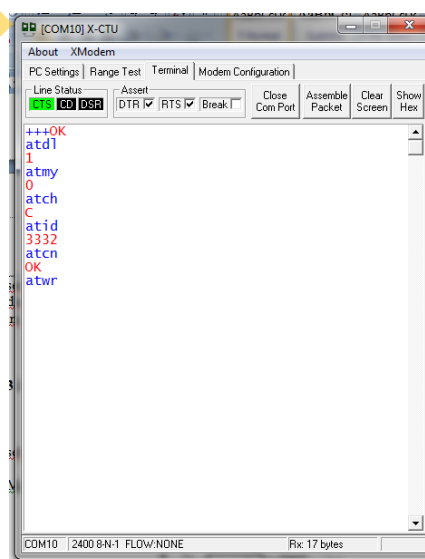
- Mempunyai *channel* ID (CH) yang berfungsi sebagai jalur yang sama.
- Mempunyai *network* ID (PAN ID) yang berfungsi sebagai pengalamatan yang sama.

- Source ID XBee-Pro receiver harus sesuai dengan destination ID dari XBee-Pro *transmitter*

Setting konfigurasi parameter yang harus dilakukan untuk menghubungkan Xbee-Pro berkomunikasi *point to point* yaitu dengan mengetikkan perintah pada Terminal XCTU ditunjukkan pada Gambar 3.7 dan Gambar 3.8.



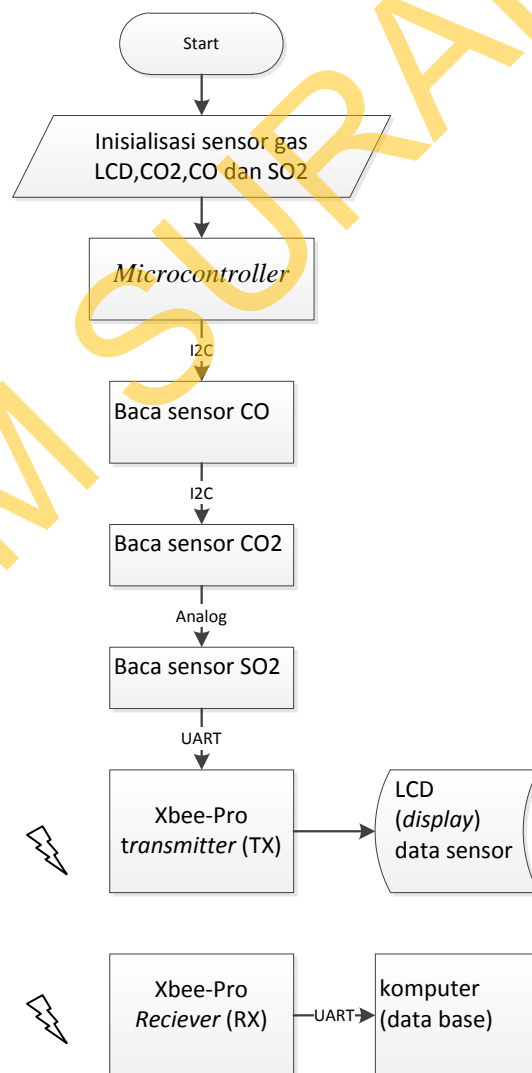
Gambar 3.7 Parameter yang dikonfigurasi pada Xbee-Pro sisi *Transmitter* (Tx)



Gambar 3.8 Parameter yang dikonfigurasi pada Xbee-Pro sisi *Receiver* (Rx)

Setelah semua parameter selesai diatur, kemudian nilai *baudrate* diubah menjadi 2400 bps. Pada Tugas Akhir ini, digunakan *baudrate* rendah yaitu 2400 bps untuk mencegah *overflow* (hilangnya data antara *host* dan modul Xbee-Pro), karena data yang dikirimkan banyak. Sebelumnya pengiriman data telah dicoba dengan *baudrate* standar dari Xbee-Pro, yaitu 9600 bps namun sering terjadi *overflow*.

3.2.4 Program Microcontroller



Gambar. 3.9 Diagram Alir.

1. Dari diagram alir Gambar 3.9 merupakan langkah-langkah alur program pertama yang dijalankan adalah inisialisasi dari *array sensor* dan *delay* atau jeda selama 10 detik untuk memanaskan *heater* dari masing- masing sensor yang ditampilkan pada LCD .

```
lcd_gotoxy (5,0);
lcd_putsf("process");
delay_ms(10000);
lcd_clear();
```

Setelah proses jeda ini selesai selanjutnya LCD akan di kosongkan untuk kemudian berjalan kode program pembacaan sensor CO.

2. Kemudian pembacaan sensor CO menggunakan komunikasi I2C.

```
//===== read sensor co =====
i2c_start(); // Start Condition
i2c_write(0xE0); // Write to module (alamat I2c ke
1)
i2c_write(0x41); // "Read Sensor" Command address
sensor co
i2c_stop(); // Stop Condition

delay_us(10); // 10 us delay

i2c_start(); // Start Condition
i2c_write(0xE1); // Read from module (alamat baca ke
1)
temp1 = i2c_read(1); // Data Sensor
temp2 = i2c_read(0); // Data Sensor
i2c_stop(); // Stop Condition

sensor1 = (temp1 * 256) + temp2 ;
//=====
```

I2C start Digunakan untuk mengirimkan sinyal start kemudian I2C *write* (0xE0) digunakan untuk mengakses atau menuliskan alamat yang dituju yaitu modul sensor CO tersebut, I2C *write* (0x41) digunakan untuk perintah membaca data dari sensor CO dan I2C stop digunakan mengirimkan sinyal stop. Kemudian diberikan jeda selama 10 detik. Kemudian dimulai lagi I2C start Digunakan untuk mengirimkan sinyal start, kemudian I2C *write* (0xE1) digunakan untuk membaca modul sensor

CO. Kemudian temp1 dan temp2 adalah variable untuk data sensor, dan I2C stop digunakan mengirimkan sinyal stop. Sensor1 adalah variable dari penjumlahan temp1 dikalikan 256 dan ditambah temp2.

3. Kemudian pembacaan sensor CO₂ menggunakan komunikasi I2C.

```
//===== read sensor CO2 =====
    i2c_start(); // Start Condition
    i2c_write(0xE2); // Write to module (alamat I2C ke
    2)
    i2c_write(0x42); // address sensor co2
    i2c_stop(); // Stop Condition

    delay_us(10); // 10 us delay

    i2c_start(); // Start Condition
    i2c_write(0xE3); // Read from module (alamat baca ke
    2)
    temp1 = i2c_read(1); // Data Sensor
    temp2 = i2c_read(0); // Data Sensor
    i2c_stop(); // Stop Condition

    sensor2 = (temp1 * 256) + temp2 ;

//=====
```

I2C start Digunakan untuk mengirimkan sinyal start kemudian I2C write (0xE2) digunakan untuk mengakses atau menuliskan alamat yang dituju yaitu modul sensor CO₂ tersebut, I2C write (0x41) digunakan untuk perintah membaca data dari sensor CO₂ dan I2C stop digunakan mengirimkan sinyal stop. Kemudian diberikan jeda selama 10 detik. Kemudian dimulai lagi I2C start digunakan untuk mengirimkan sinyal start, kemudian I2C write (0xE3) digunakan untuk membaca modul sensor CO₂. Kemudian temp1 dan temp2 adalah variable untuk data sensor, dan I2C stop digunakan mengirimkan sinyal stop. Sensor2 adalah variable dari penjumlahan temp1 dikalikan 256 dan ditambah temp2.

4. Kemudian pembacaan sensor SO_2 menggunakan komunikasi data ADC dan pengiriman data ke komputer.

```
//===== read sensor tgs2602=====
    adc = read_adc(3);
    delay_ms(10) ;
    itoa (adc,adc1);
    lcd_gotoxy (8,1);
    lcd_puts ("So2=");
    lcd_gotoxy (12,1);
    lcd_puts (adc1);
    puts("!");
    puts(adc1);
    puts("?");
//=====
```

Pin ADC yang digunakan adalah pin C bit 3 maka ADC membaca pada bit ke 3 kemudian diberikan jeda pembacaan sensor. Dari *microcontroller* mengirim data ke komputer menggunakan komunikasi serial, sedangkan pengiriman serial hanya bisa dengan 8 bit sedangkan data sensor yang dikirimkan adalah 10 bit jadi data dirubah menjadi data string untuk dikirim dengan dua kal pengiriman. Dengan fungsi itoa, tanda ("!") digunakan mengawali pengiriman awal data dan ("?",) digunakan untuk mengakhiri pengiriman akhir data, untuk kemudian di tampilkan pada LCD dan dikirimkan ke komputer.

5. Pengiriman data sensor CO_2 dari *microcontroller* ke komputer.

```
//===== pengiriman data sensor co2 =====
    itoa(sensor1,kirim_co2);
    lcd_gotoxy(0,0);
    lcd_puts ("CO2 = ");
    lcd_puts(kirim_co2);
    puts("*");
    puts(kirim_co2);
    puts("#");
//=====
```

Dari *microcontroller* mengirim data sensor1 dengan variable kirim_co2 ke komputer menggunakan komunikasi serial, sedangkan pengiriman serial hanya bisa dengan 8 bit sedangkan data sensor yang

dikirimkan adalah 10 bit jadi data dirubah menjadi data string untuk dikirim dengan dua kal pengiriman dengan fungsi itoa, tanda (“*”) digunakan mengawali pengiriman awal data dan (“#”) digunakan untuk mengakhiri pengiriman akhir data, untuk kemudian di tampilkan pada LCD dan dikirimkan ke komputer.

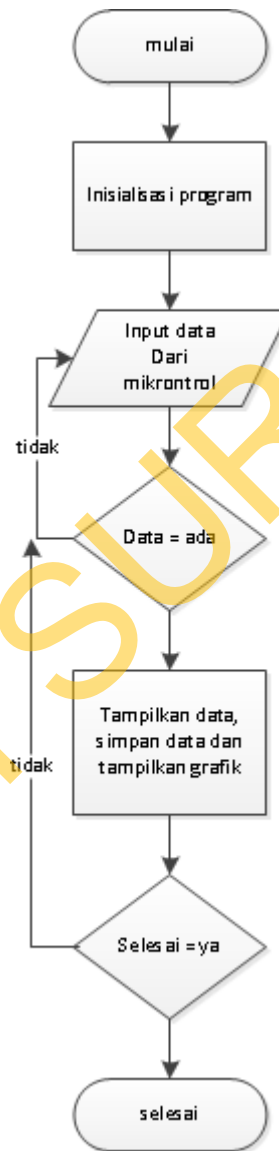
6. Pengiriman data sensor CO dari *microcontroller* ke komputer.

```
//===== peniriman data sensor co =====
    itoa(sensor2,kirim_co);
    lcd_gotoxy(0,1);
    lcd_puts ("CO= ");
    lcd_puts(kirim_co);
    puts("-");
    puts(kirim_co);
    puts("+");
//=====
```

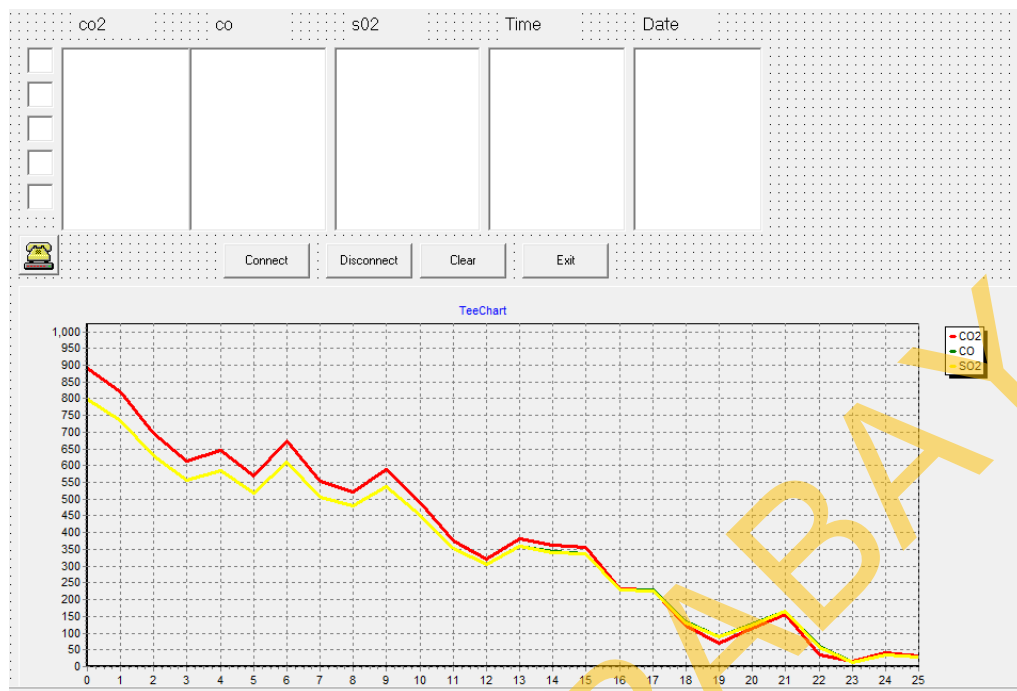
Dari *microcontroller* mengirim data sensor2 ke komputer dengan variable kirim_co menggunakan komunikasi serial, sedangkan pengiriman serial hanya bisa dengan 8 bit sedangkan data sensor yang dikirimkan adalah 10 bit jadi data dirubah menjadi data string untuk dikirim dengan dua kal pengiriman Dengan fungsi itoa, tanda (“-”) digunakan mengawali pengiriman awal data dan (“+”) digunakan untuk mengakhiri pengiriman akhir data, untuk kemudian di tampilkan pada LCD dan dikirimkan ke komputer.

3.2.5 Program pada Visual Basic 6.0

Digunakan untuk menerima dan menyimpan data yang dikirimkan dari *microcontroller*. Data diterima per karakter, lalu diproses dan ditampilkan dalam bentuk grafik dan angka. Untuk lebih mudahnya dapat dilihat pada Gambar 3.9.



Gambar. 3.10 diagram alir program visual basic 6.0



Gambar 3.11 tampilan form aplikasi yang dibuat

Gambar 3.11 adalah tampilan dari form aplikasi yang dibuat untuk berkomunikasi dengan *microcontroller* untuk kemudian menerima dan menyimpan data yang dikirimkan oleh *micorocontroller* dan ditampilkan pada grafik, Berikut ini adalah langkah pembuatan program pada komputer dengan menggunakan program visual basic 6:

a. Inisialisasi komunikasi serial

Pada saat melakukan inisialisasi komunikasi serial di pemrograman visual khususnya program VB ada komponen yang harus digunakan adalah komponen *Microsoft comm control 6.0*. lebih jelasnya lihat Gambar 3.12. Perlu diketahui bahwa komponen ini masih tersimpan di dalam data base komponen program VB maka dari itu untuk memunculkannya yang harus dilakukan adalah berikut:

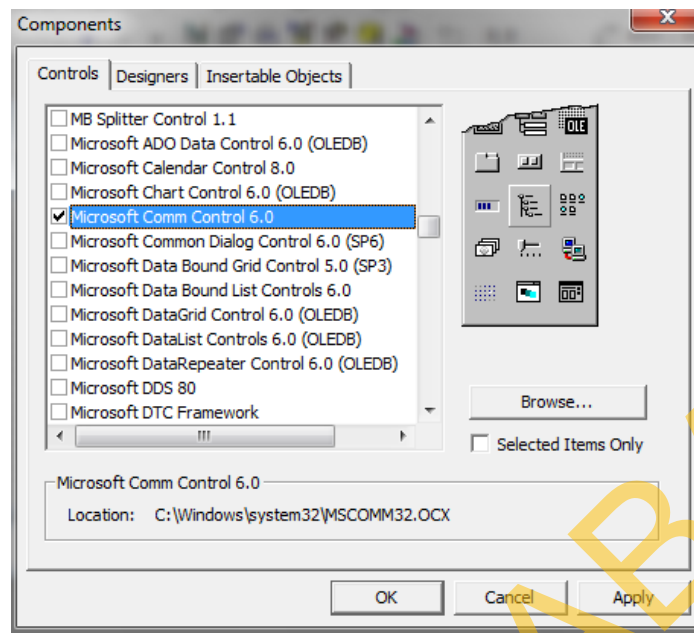
1. Klik kanan pada *Toolbox general* (lihat Gambar 3.13), pilih *component* kemudian akan muncul *dialog components* lihat Gambar 3.14. pada *tab control* cari *Microsoft comm control 6.0*. lalu centang kemudian klik *apply* dan *close*.



Gambar 3.12. Komponen *Microsoft comm control 6.0*.



Gambar 3.13. *Toolbox general*



Gambar 3.14. Dialog components Microsoft comm control 6.0.

2. Setelah itu komponen *Microsoft comm control 6.0*. ini akan muncul pada *Toolbox general* lihat Gambar 3.15.



Gambar 3.15. komponen *MSComm* muncul pada *toolbox general*

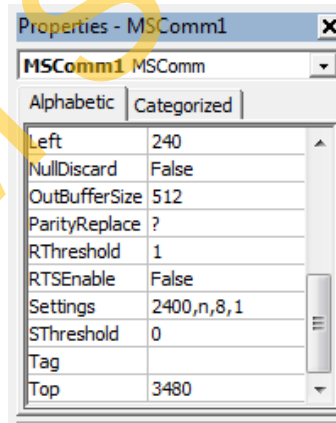
3. Penulisan intruksi ke dalam program

```
.....

MSComm1.PortOpen = True

.....
```

Hal ini berarti komponen *MSComm* telah aktif dan siap melakukan komunikasi dengan aplikasi dari luar. Selanjutnya pengaturan parameter dari *MSComm*, dalam hal pengaturan parameter harus mengerti beberapa hal yang harus disesuaikan dengan hardware yang digunakan misalnya kebutuhan akan nilai dari *baudrate* yang digunakan, jumlah *bit* dalam satu paket data, *parity check* dan *stop bit* yang digunakan. Parameter serial yang digunakan adalah *baudrate* 2400bps, 8 *bit* data, *no parity check* dan menggunakan 1 *stop bit* yang telah disesuaikan dengan modul *microcontroller*, seperti pada Gambar 3.16.



Gambar 3.16. Pengaturan parameter komponen MSComm.

Pengaturan pada proses penerimaan data menggunakan *MSComm* berikut adalah potongan programnya.

```

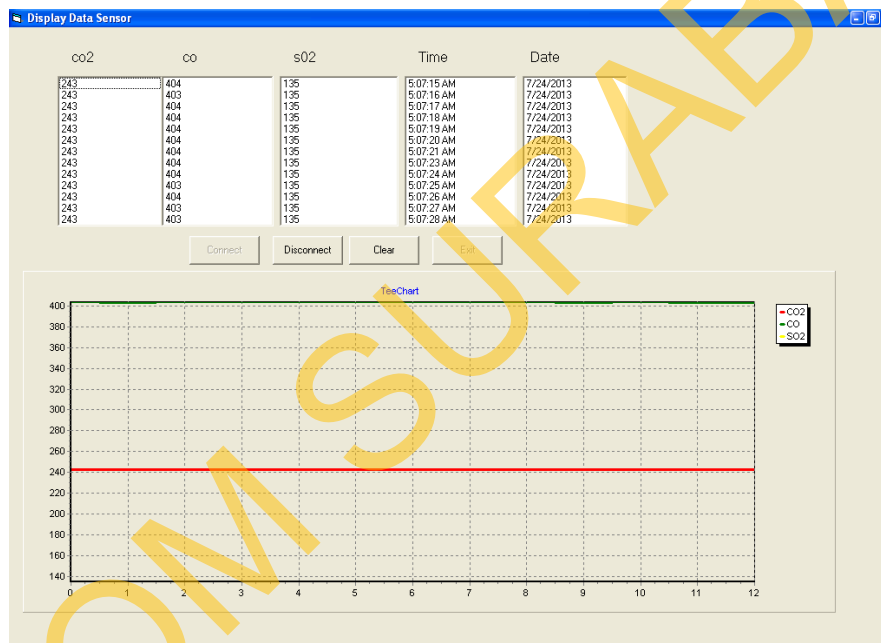
.....

dt = MScComm1.Input

.....

```

Arti dari potongan program tersebut adalah jalur *MScComm* pada jalur *input* mempersilahkan data yang masuk untuk diterima kemudian disimpan pada variabel data. Berikut adalah gambar tampilan awal pada program *visual basic* 6 saat dijalankan seperti pada Gambar 3.17.



Gambar 3.17. Tampilan pada *visual basic* saat dijalankan

Dari Gambar 3.17 saat aplikasi dijalankan dengan menekan tombol *connet* terlihat data diterima dan langsung disimpan pada log data base kemudian di tampilkan pada grafik. Berikut adalah keterangan tiap *point* yang ada pada gambar.

1. Listbox co2

Digunakan untuk menampilkan data sensor CO₂ yang diterima, berikut adalah potongan program untuk menerima data.


```

.....

If dt = "*" Then
    inp = ""
    ElseIf dt = "#" Then
        co2 = CInt(inp)
        List1.AddItem (CStr(co2) + vbTab)
        Text1.Text = (CStr(co2))
.....

```

2. Listbox CO

Digunakan untuk menampilkan data sensor CO₂ yang diterima, berikut adalah potongan program untuk menerima data.

```

.....

If dt = "*" Then
    inp = ""
    ElseIf dt = "#" Then
        co = CInt(inp)
        List4.AddItem (CStr(co) + vbTab)
        Text4.Text = (CStr(co))
.....

```

3. Listbox SO₂

Digunakan untuk menampilkan data sensor SO₂ yang diterima, berikut adalah potongan program untuk menerima data.

```

.....

If dt = "*" Then
    inp = ""
    ElseIf dt = "#" Then
        co = CInt(inp)
        List5.AddItem (CStr(co) + vbTab)
        Text5.Text = (CStr(co))
.....

```

4. Listbox time

Digunakan untuk menampilkan time, Time yang tertampil ini mengacu pada time yang berjalan pada komputer. berikut adalah potongan program untuk menampilkan time.

```

.....

List2.AddItem (Time)
Text2.Text = Time

.....

```

5. Listbox date

Digunakan untuk menampilkan date, date yang tertampil ini mengacu pada date yang berjalan pada komputer. berikut adalah potongan program untuk menampilkan date.

```

.....

List3.AddItem (Date)
Text3.Text = Date

.....

```

6. Tombol connet

Tombol connet digunakan untuk mengaktifkan penerimaan data dan menjalankan aplikasi. berikut adalah potongan program untuk mengaktifkan penerimaan data.

```

.....

Command4.Enabled = True
If MSComm1.InBufferCount <> 0 Then
    dt = MSComm1.Input

.....

```

Selain untuk mengaktifkan penerimaan data tombol connect juga berfungsi menyimpan data pada log file. Berikut potongan program untuk menyimpan data.

```

.....

Command4.Enabled = False
FileName = "D:\_____TA jily\Data Sensor" + ".txt"
fr = FreeFile
Open FileName For Append As #fr

```

```
Print #fr, Text1.Text + vbTab + Text4.Text + vbTab +
Text5.Text + vbTab + Text2.Text + vbTab + Text3.Text
Close #fr
```

Berikut ini Untuk membuat file direktori penyimpanan data atau log file seperti Gambar 3.18. Berikut potongan programnya.

```
Private Sub Form_Load()
On Error Resume Next
If Dir$("d:\_____TA jily\") = "" Then
Mkdir "d:\_____TA jily\"
End If
End Sub
```

ID	Sensor ID	Time	Date
42	411	179	9:38:58 PM 1/15/2002
42	411	179	9:38:59 PM 1/15/2002
42	411	179	9:39:00 PM 1/15/2002
42	411	179	9:39:01 PM 1/15/2002
42	410	179	9:39:32 PM 1/15/2002
42	410	179	9:39:33 PM 1/15/2002
42	410	178	9:39:34 PM 1/15/2002
41	410	179	9:39:35 PM 1/15/2002
42	409	177	9:39:36 PM 1/15/2002
42	409	177	9:39:38 PM 1/15/2002
42	411	176	9:39:40 PM 1/15/2002
42	409	176	9:39:41 PM 1/15/2002
42	409	177	9:39:42 PM 1/15/2002
42	410	176	9:39:43 PM 1/15/2002
42	410	176	9:39:44 PM 1/15/2002
42	411	176	9:39:45 PM 1/15/2002
42	410	176	9:39:46 PM 1/15/2002
42	411	176	9:39:47 PM 1/15/2002
42	411	176	9:39:49 PM 1/15/2002
42	411	176	9:39:50 PM 1/15/2002
42	410	177	9:39:51 PM 1/15/2002
42	410	176	9:39:52 PM 1/15/2002
42	409	177	9:39:53 PM 1/15/2002
42	409	176	9:39:54 PM 1/15/2002
42	409	177	9:39:55 PM 1/15/2002
42	409	178	9:39:56 PM 1/15/2002
42	410	179	9:39:57 PM 1/15/2002
42	410	178	9:39:59 PM 1/15/2002
42	410	178	9:40:00 PM 1/15/2002
42	410	177	9:40:01 PM 1/15/2002
42	409	177	9:40:02 PM 1/15/2002
42	408	177	9:40:03 PM 1/15/2002
42	409	177	9:40:04 PM 1/15/2002
42	409	178	9:40:05 PM 1/15/2002
42	409	178	9:40:06 PM 1/15/2002
42	409	177	9:40:07 PM 1/15/2002
42	410	179	9:40:08 PM 1/15/2002
42	409	177	9:40:09 PM 1/15/2002
42	407	177	9:40:11 PM 1/15/2002
42	407	177	9:40:12 PM 1/15/2002
42	407	177	9:40:13 PM 1/15/2002
42	407	177	9:40:14 PM 1/15/2002
42	406	177	9:40:15 PM 1/15/2002
42	407	177	9:40:16 PM 1/15/2002
42	406	177	9:40:17 PM 1/15/2002
42	407	177	9:40:18 PM 1/15/2002
42	406	177	9:40:19 PM 1/15/2002
40	403	174	9:40:56 PM 7/12/2013
40	404	174	9:40:57 PM 7/12/2013
40	404	173	9:40:58 PM 7/12/2013
40	404	172	9:40:59 PM 7/12/2013

Gambar 3.18 file log data *array sensor*

7. Tombol disconnect

Tombol connet digunakan untuk menghentikan sementara penerimaan data. berikut adalah potongan program untuk mengaktifkan penerimaan data.

```

.....

MSComm1.PortOpen = False
Command4.Enabled = False

.....

```

8. Tombol clear

Tombol clear digunakan untuk mengosongkan lisbox CO,CO₂,SO₂, date dan time. Berikut adalah potongan program untuk mengosongkan listbox CO,CO₂,SO₂, date dan time.

```

.....

List1.Clear
List2.Clear
List3.Clear
List4.Clear
List5.Clear

.....

```

9. Tombol exit

Tombol exit digunakan untuk keluar dari aplikasi. Berikut adalah potongan programnya.

```

.....

End

.....

```

10. Grafik penampil data *array sensor*

Grafik digunakan untuk menampilkan data yang diterima sehingga perubahan data dapat tepantau dengan baik dimana sumbu Y adalah data dari sensor dan sumbu X adalah waktu . Berikut adalah potongan programnya.

```

.....

Call TChart1.Series(0).AddXY(ye, (CStr(co2)), "", clTeeColor)

```

```
Call TChart1.Series(0).AddXY(ye, (CStr(co)), "",  
clTeeColor)
```

```
Call TChart1.Series(0).AddXY(ye, (CStr(so2)), "",  
clTeeColor)
```

.....

STIKOM SURABAYA