

# DT-AVR

## DT-AVR *Application Note*

### AN187 – *Monitoring Suhu Nirkabel*

(Bahasa C)

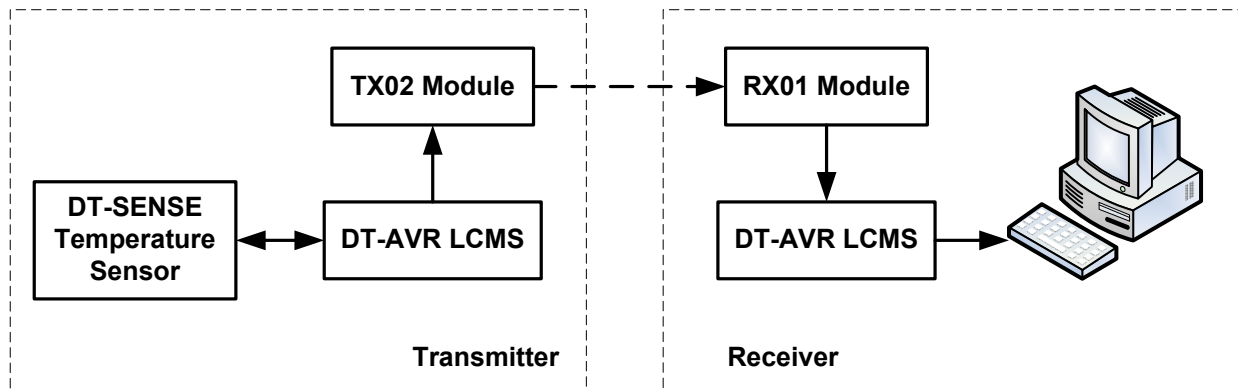
Oleh: Tim IE

**K**omunikasi data secara nirkabel (*wireless*) saat ini berkembang cukup cepat. Cukup banyak media komunikasi yang bisa digunakan untuk aplikasi komunikasi secara nirkabel seperti cahaya, suara, maupun RF (*Radio Frequency*). *Monitoring* suhu nirkabel pada aplikasi ini memanfaatkan media komunikasi RF dengan pita frekuensi 915 MHz menggunakan sepasang modul RF yaitu RX01 dan TX02. Sepasang modul ini digunakan untuk mengirimkan data pembacaan suhu oleh DT-SENSE Temperature Sensor. Aplikasi ini terdiri dari 2 bagian yaitu bagian *receiver* (penerima) yang terhubung dengan komputer dan bagian *transmitter* (pengirim) yang terhubung dengan DT-SENSE Temperature Sensor. *Transmitter* akan mengirimkan data secara terus menerus ke *receiver* dengan interval waktu tertentu. *Receiver* akan mengolah data yang diterima kemudian dikirimkan ke komputer. Bagian *transmitter* terdiri dari DT-AVR Low Cost Micro System (DT-AVR LCMS) dan modul TX02 sedangkan bagian *receiver* terdiri dari DT-AVR LCMS dan modul RX01.

Modul & komponen yang digunakan pada aplikasi ini adalah:

- 2x DT-AVR LCMS.
- 1x modul RX01.
- 1x modul TX02.
- 1x resistor 10 K/0.25W.
- 1 set kabel penghubung.

**B**lok diagram aplikasi “*Monitoring Suhu Nirkabel*” sebagai berikut:



Gambar 1  
Blok Diagram Aplikasi *Monitoring Suhu Nirkabel*

**H**ubungan antar modul pada aplikasi ini sebagai berikut:

DT-AVR LCMS	DT-SENSE Temperature Sensor
VCC (J13 pin 2)	VCC (J1 pin 2)
GND (J13 pin 1)	GND (J1 pin 1)
PD.0 (J13 pin 3)	TX TTL (J1 pin 4)
PD.1 (J13 pin 4)	RX TTL (J1 pin 3)

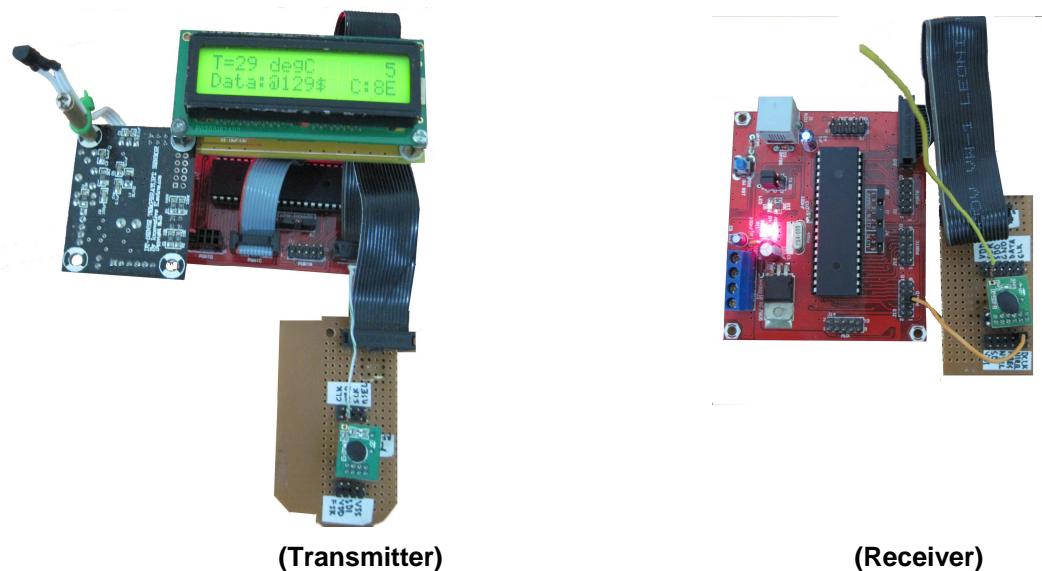
Tabel 1  
Hubungan antara DT-AVR LCMS dengan DT-SENSE Temperature Sensor secara serial

DT-AVR LCMS	RX01 Module
VCC (J10 pin 2)	VDD
GND (J10 pin 1)	GND
PA.4 (J10 pin 7)	NSEL
PA.5 (J10 pin 8)	SDI
PA.6 (J10 pin 9)	SDO
PA.7 (J10 pin 10)	SCK
PD.2 (J13 pin 5)	NIRQ

**Tabel 2**  
Hubungan antara DT-AVR LCMS dengan RX01 Module

DT-AVR LCMS	TX02 Module
VCC (J10 pin 2)	VDD
GND (J10 pin 1)	GND
PA.1 (J10 pin 4)	FSK
PA.4 (J10 pin 7)	NSEL
PA.5 (J10 pin 8)	SDI
PA.6 (J10 pin 9)	NIRQ
PA.7 (J10 pin 10)	SCK

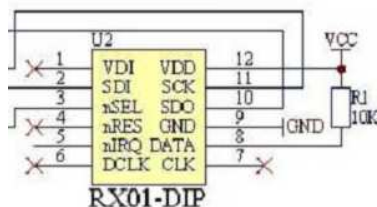
**Tabel 3**  
Hubungan antara DT-AVR LCMS dengan TX02 Module



**Gambar 2**  
Bagian pengirim (transmitter) dan penerima (receiver) yang sudah dirakit

**D**T-SENSE Temperature Sensor terhubung dengan DT-AVR LCMS melalui antarmuka serial TTL. Konfigurasi komunikasi serial yang digunakan adalah 38400 bps, N, 8, 1. Oleh karena itu, *jumper* J4 & J5 pada modul DT-AVR LCMS (bagian *transmitter*) harus dipasang pada posisi 2-3 (UART TTL). Data temperatur yang dibaca pada DT-

SENSE Temperature Sensor adalah data temperatur dalam satuan derajat Celcius. Pembacaan temperature dimungkinkan juga dalam satuan derajat Fahrenheit dengan melakukan perubahan program. Selanjutnya data pembacaan temperatur oleh DT-SENSE Temperature Sensor akan diolah oleh DT-AVR LCMS dan dipaket menjadi sebuah *frame data* sebelum dikirimkan melalui modul TX02. *Update* data pembacaan temperatur dan pengiriman data secara nirkabel dilakukan setiap 500 ms. Data yang dikirim secara nirkabel via modul TX02 akan diterima oleh modul RX02. Pada rangkaian modul RX01, perlu ditambahkan resistor sebesar 10K/0.25W dengan pemasangan sebagai berikut:



**Gambar 3**  
**Letak pemasangan resistor 10K/0.25W**

Pin NIRQ pada modul RX01 dimanfaatkan sebagai indikator untuk mengetahui ada tidaknya data yang masuk ke *buffer* penerima modul RX01. Selanjutnya *frame data* yang diterima akan dikirimkan ke komputer untuk diurai (*di-parsing*) sehingga mendapatkan beberapa informasi yang diperlukan. Proses penguraian data yang diterima ini dilakukan oleh program pada komputer yang ditulis menggunakan bahasa BASIC dengan IDE Visual BASIC 6.0. DT-AVR LCMS pada bagian *receiver* ini berfungsi untuk melakukan inisialisasi awal pada modul RX02 sebelum dapat menerima data dan meneruskan *frame data* yang diterima oleh modul RX01 ke komputer untuk pemrosesan lebih lanjut. DT-AVR LCMS terhubung dengan komputer melalui antarmuka serial UART RS-232, sehingga *jumper* J4 & J5 harus diatur pada posisi 1-2 (UART RS-232). Jika komputer yang digunakan tidak memiliki *port serial*, maka dapat digunakan *USB to Serial Converter* sehingga DT-AVR LCMS dapat terhubung dengan komputer melalui *port USB*. Setelah semua modul dihubungkan sesuai dengan tabel koneksi antar modul, hubungkan DT-AVR LCMS dengan catu daya dan isikan program tx.hex ke DT-AVR LCMS bagian pengirim (*transmitter*) dan program rx.hex ke DT-AVR LCMS bagian penerima (*receiver*) menggunakan DT-AVR USB ISP atau divais programmer lain yang kompatibel. Setelah proses pengisian program selesai, tekan tombol *reset* pada kedua modul untuk memastikan program dieksekusi mulai awal.

*Frame data* yang dikirimkan oleh *transmitter* memiliki susunan sebagai berikut:

STX [byte 1]	NID [byte 2]	TC [byte 3]	CS [byte 4]	ETX [byte 5]
'@'	'1'	[0x01 – 0x64]	[0x01 – 0xFF]	'\$'

Keterangan:

- **STX**, merupakan awal *frame data*. STX menggunakan karakter '@'.
- **NID**, merupakan nomor ID sensor. Pada aplikasi ini digunakan nomor ID sensor '1'.
- **TC**, merupakan data suhu dalam satuan derajat Celcius. *Range* nilainya antara 0x01 sampai 0x64 atau 0 sampai 100 derajat Celcius.
- **CS**, merupakan nilai Check Sum dari *frame data*. *Range* nilai Check Sum antara 0x01 sampai 0xFF. Nilai Check Sum merupakan nilai penjumlahan data byte mulai awal *frame* sampai TC dengan mengambil 1 byte data saja.
- **ETX**, merupakan akhir *frame data*. STX menggunakan karakter '\$'.

**P**rogram pada komputer berupa sebuah GUI (*Graphical User Interface*) yang akan menampilkan *frame data* yang dikirimkan oleh modul TX02 dan diterima oleh modul RX01. Program ini memanfaatkan komponen MSCOMM32.OCX pada Visual BASIC 6.0. Penerimaan data yang masuk ke komputer secara serial memanfaatkan event *comEvReceive* (*RS-232 Interrupt Service Routine*) pada MSCOMM. Pada pemrosesan data di komputer juga dilakukan pengecekan validitas data kembali. Prinsip proses pengecekannya sama dengan proses pengecekan pada mikrokontroler. Hanya data yang valid saja yang akan ditampilkan pada *textbox*. Data pada *textbox* terdiri dari beberapa informasi seperti:

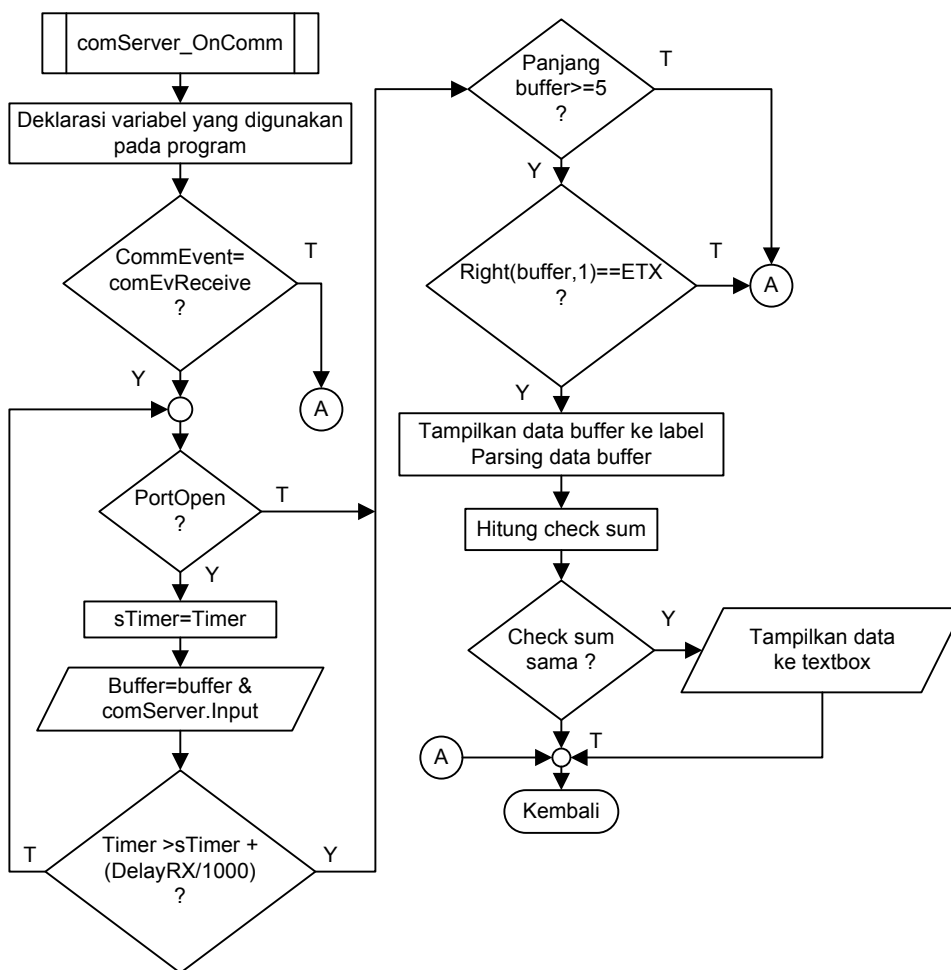
- Informasi waktu saat data diterima oleh komputer (berdasarkan waktu jam pada BIOS).

- Informasi Node ID. Informasi ini akan bermanfaat jika DT-SENSE Temperature Sensor yang terpasang lebih dari 1.
- Informasi data suhu dalam satuan derajat celcius.
- Informasi nilai check sum dari *frame data* yang diterima.



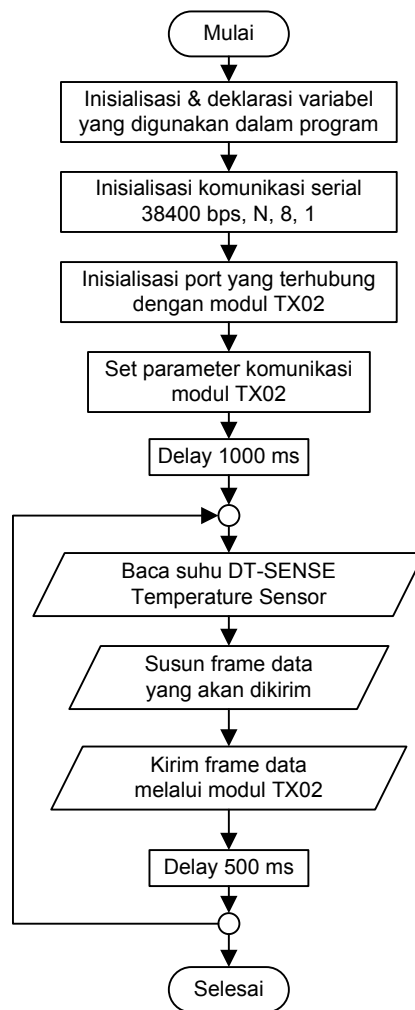
**Gambar 4**  
Tampilan GUI pada komputer saat dijalankan.

Proses pengolahan data yang diterima saat terjadi even penerimaan data pada *port serial* komputer (*comEvReceive*) dapat diilustrasikan pada *flowchart* berikut ini:



**Gambar 5**  
*Flowchart* event *comServer\_OnCom*

**F**lowchart program tx.c pada bagian *transmitter* sebagai berikut:



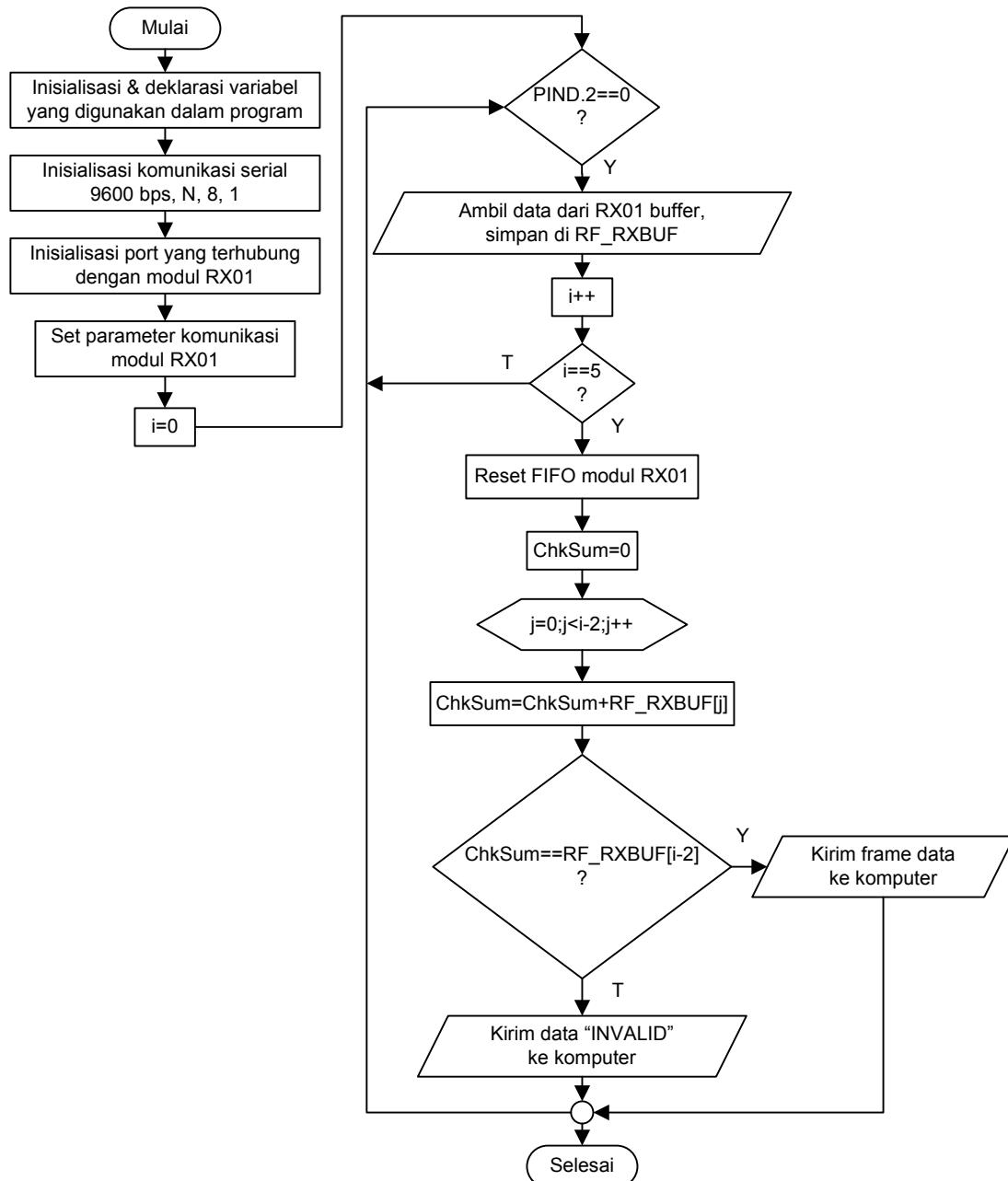
**Gambar 6**  
**Flowchart program tx.c pada bagian pengirim (*transmitter*)**

Program tx.c pada DT-AVR LCMS bagian pengirim (*transmitter*) dapat dijelaskan sebagai berikut:

- Pertama kali program akan mendeklarasikan dan melakukan inisialisasi variabel-variabel yang digunakan pada program.
- Selanjutnya program akan melakukan inisialisasi komunikasi serial yang nantinya digunakan untuk mengakses DT-SENSE Temperature Sensor. Konfigurasi komunikasi serial yang digunakan adalah 38400 bps, N, 8,1 (d disesuaikan dengan konfigurasi komunikasi serial modul DT-SENSE Temperature Sensor).
- Program melakukan inisialisasi *port* I/O mikrokontroler ATmega8535 pada DT-AVR LCMS yang terhubung dengan modul TX02. Pin-pin modul TX02 yang terhubung dengan modul TX02 adalah VDD, GND, FSK, NSEL, SDI, SCK, dan NIRQ. Pin-pin modul TX02 tersebut terhubung dengan PORT A ATmega8535 pada DT-AVR LCMS.
- Kemudian program melakukan *setting* konfigurasi komunikasi modul TX02. *Setting* konfigurasi meliputi beberapa hal sebagai berikut:
  1. Pengaturan frekuensi kerja pada 915 MHz.
  2. Pengaturan *data rate* pada 4800 bps.
  3. Aktifasi Bit Sync (Bit Sinkronisasi).
- Program melakukan delay selama 1000 ms (1 detik).
- Program selanjutnya akan membaca data temperatur (dalam satuan derajat Celcius) dari modul DT-SENSE Temperature Sensor. Perintah yang digunakan untuk membaca data suhu dalam satuan derajat Celcius adalah READ TEMPERATURE C dengan command 00H. Setelah melakukan pengiriman command 00H, program akan menunggu sekitar 30 ms untuk mendapatkan respon yang valid dari DT-SENSE Temperature Sensor.

- Hasil pembacaan data suhu selanjutnya akan diproses oleh program dan dipaket menjadi *sebuah frame data* untuk dikirimkan melalui modul TX02.
- *Frame data* yang telah selesai dipaket selanjutnya dikirimkan oleh modul TX02 ke modul RX01. Kemudian program melakukan *delay* selama 500 ms.
- Program kembali melakukan pembacaan data suhu.

**F**lowchart program rx.c pada bagian *receiver* sebagai berikut:



**Gambar 7**  
**Flowchart program rx.c pada bagian penerima (receiver)**

Program rx.c pada DT-AVR LCMS bagian penerima (*receiver*) dapat dijelaskan sebagai berikut:

- Pertama kali program akan melakukan deklarasi dan inisialisasi variabel-variabel yang digunakan dalam program.

- Program melakukan inialisasi komunikasi serial yang nantinya digunakan untuk komunikasi antara komputer dengan DT-AVR LCMS. Konfigurasi komunikasi serial yang digunakan adalah 9600 bps, N, 8, 1. Komputer terhubung dengan DT-AVR LCMS melalui komunikasi serial UART RS-232.
- Program melakukan inialisasi *port* I/O mikrokontroler ATmega8535 pada DT-AVR LCMS yang terhubung dengan modul RX01. Pin-pin modul RX01 yang terhubung dengan modul RX01 adalah VDD, GND, FSK, NSEL, SDI, SCK, dan NIRQ. Pin-pin modul RX01 tersebut terhubung dengan PORT A ATmega8535 pada DT-AVR LCMS.
- Kemudian program melakukan *setting* komunikasi modul RX01. *Setting* konfigurasi yang dilakukan meliputi:
  1. Pengaturan frekuensi kerja pada 915 MHz dan *bandwidth* komunikasi 134 kHz.
  2. Pengaturan *data rate* sebesar 4800 bps.
  3. Aktifasi fungsi FIFO dan *receiver* mode untuk penerimaan data.
- Program melakukan inialisasi variabel *i*=0, yang nantinya digunakan untuk *pooling* penerimaan data.
- Program selanjutnya melakukan pengecekan PIND.2. Jika nilainya 0 berarti pin NIRQ aktif. Hal ini menandakan bahwa ada data yang masuk ke *buffer* FIFO modul RX01.
- Jika terdeteksi ada data yang masuk ke *buffer* FIFO, maka program akan mengambil data tersebut dan selanjutnya diisikan ke variabel *array* RF\_RXBUF dengan indeks sesuai nilai *i*. Selanjutnya nilai *i* di-*increment*.
- Selanjutnya program akan melakukan pengecekan nilai *i*.
- Jika nilai *i*=5 maka program akan me-*reset* *buffer* FIFO agar nantinya siap untuk menerima data yang baru. Selanjutnya dilakukan pemrosesan data yang diterima. Program melakukan inialisasi variabel ChkSum=0 dan melakukan perhitungan check sum untuk mengetahui validitas data. Nilai ChkSum merupakan hasil penjumlahan seluruh data yang disimpan di variabel RF\_RXBUF (kecuali data ke-(*i*-2)) dengan mengambil 8 bit data LSB.
- Selanjutnya dari hasil perhitungan check sum yang tersimpan di variabel ChkSum akan dibandingkan dengan nilai data RF\_RXBUF[*i*-2] atau RF\_RXBUF[3]. Jika nilainya sama maka data valid, jika tidak sama maka data tidak valid.
- Jika data valid selanjutnya data tersebut akan dikirimkan ke komputer. Jika data tidak valid maka program akan mengirimkan data "Invalid" ke komputer. Variabel *I* kemudian diset menjadi 0.
- Selanjutnya program akan melakukan pengecekan kembali nilai PIND.2 untuk memulai siklus pengecekan data kembali.

Jika *user* memiliki modul LCD karakter seperti EMS LCD Display atau modul lain yang kompatibel, *user* dapat memantau proses pengiriman data pada sisi pengirim (*transmitter*). Data-data yang dapat di-*monitoring* adalah *frame data* yang dikirim, nilai check sum data, dan *counter* yang menunjukkan proses *update* pembacaan data suhu. Pada program sisi pengirim (*transmitter*) telah dialokasikan PORT C untuk penggunaan LCD. *User* perlu melakukan modifikasi program dengan mengubah baris program berikut ini:

```

73
74 /*My define*/
75 #define DEBUG
76 #define dBaca 30
77 #define dUpdate 500
78 #define STX '@'
79 #define ETX '$'
80 #define NID '1'
81 #define datalen 5 //[@][NID][DAT1][DAT0][$][CHKSUM]->@128$[0xFF]
82

```

ubah di baris ini menjadi  
#define DEBUG

**Gambar 8**  
Posisi modifikasi pada program tx.c

Proses modifikasi ini bertujuan untuk mengaktifkan fitur DEBUG sehingga beberapa data tertentu bisa ditampilkan pada LCD yang dialokasikan untuk terpasang pada PORT C DT-AVR LCMS bagian pengirim (*transmitter*). Koneksi antara EMS LCD Display dengan modul DT-AVR LCMS pada bagian pengirim (*transmitter*) sebagai berikut:

DT-AVR LCMS	EMS LCD Display
VCC (J12 pin 2)	VCC (J3 pin 2)
GND (J12 pin 1)	GND (J3 pin 1)
PORTC.0 (J12 pin 3)	RS (J3 pin 3)
PORTC.1 (J12 pin 4)	R/W (J3 pin 4)
PORTC.2 (J12 pin 5)	EN (J3 pin 5)
PORTC.3 (J12 pin 6)	-
PORTC.4 (J12 pin 7)	DB4 (J3 pin 7)
PORTC.5 (J12 pin 8)	DB5 (J3 pin 8)
PORTC.6 (J12 pin 9)	DB6 (J3 pin 9)
PORTC.7 (J12 pin 10)	DB7 (J3 pin 10)

**Tabel 4**  
**Koneksi antara DT-AVR LCMS dengan EMS LCD Display**

**L**isting program aplikasi ini terdapat pada **AN187.zip**

**S**elamat berinovasi!

*All trademarks, trade names, company names, and product names are the property of their respective owners.  
All softwares are copyright by their respective software publishers and/or creators.*