

# Perancangan dan Pembuatan Modul PAD (*Packet Assembler Disassembler*) dengan Protokol AX.25 untuk Stasiun Bumi Portabel pada Sistem Komunikasi Satelit

Dicky Rismawan<sup>1)</sup>, Rudy Dikairono<sup>2)</sup>, Gamantyo Hendranto<sup>3)</sup>.

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember  
Kampus ITS, Surabaya 60111

[dicky.rismawan.raharjo@gmail.com](mailto:dicky.rismawan.raharjo@gmail.com)<sup>1)</sup>, [rudydikairono@gmail.com](mailto:rudydikairono@gmail.com)<sup>2)</sup>, [gamantyo@yahoo.com](mailto:gamantyo@yahoo.com)<sup>3)</sup>

**Abstrak**—Pada Sistem Komunikasi Satelit, komunikasi terjadi antara satelit yang berada di luar angkasa dengan stasiun pengendali yang berada di bumi atau sebaliknya. Umumnya stasiun bumi mempunyai posisi yang tetap, sehingga untuk berkomunikasi dengan satelit terbatas pada stasiun bumi yang telah ditentukan. Tujuan pembuatan stasiun bumi portabel adalah untuk menjamin ketersediaan komunikasi dengan satelit tanpa batasan tempat. Inti utama stasiun bumi portabel adalah modul PAD (*Packet Assembler Disassembler*) yang berfungsi untuk mengirimkan dan/atau menerima informasi kepada/dari satelit dalam bentuk paket radio. Untuk komunikasi data dibutuhkan sebuah protokol yang mengatur tata cara komunikasi dalam bentuk paket radio. Protokol yang digunakan untuk komunikasi paket radio adalah protokol AX.25. Protokol ini digunakan oleh komunitas radio amatir untuk mengirimkan informasi berupa teks. Dalam penelitian ini, protokol AX.25 ini akan diimplementasikan ke dalam perangkat lunak dan perangkat keras berbasis mikrokontroler ATmega1280. Hasil yang implementasi protokol AX.25 adalah stasiun bumi portabel yang mampu mengirim dan/atau menerima data teks kepada atau dari satelit.

**Kata Kunci**—AX.25, mikrokontroler, paket radio, protokol.

## I. PENDAHULUAN

Dalam setiap sistem komunikasi terdapat sebuah pengirim dan sebuah penerima. Di dalam sistem komunikasi satelit pengirim bisa berupa sebuah satelit dan penerima bisa berupa sebuah stasiun yang berada di bumi atau sebaliknya. Karena jarak antar pengirim dan penerima sangat jauh maka digunakan media transmisi yang berupa gelombang radio yang mampu mencapai jarak yang jauh, dimana data akan dimodulasi ke dalam gelombang radio untuk ditransmisikan. Data disini harus memiliki bentuk khusus untuk dapat ditransmisikan. Salah satu metodenya adalah mengubah data ke dalam bentuk paket radio. Salah satu protokol yang mengatur paket radio adalah AX.25. Protokol ini umumnya digunakan oleh komunitas radio amatir untuk mengirimkan data teks melalui gelombang radio. Selain itu juga protokol AX.25 telah terbukti mampu untuk mengirim data teks dengan

pendeteksi kesalahan yang mungkin muncul di dalam proses transmisi.

AX.25 merupakan sebuah protokol yang mampu mengatur mengenai proses terjadinya koneksi *link*, metode pertukaran data dan pemutusan hubungan *link*. Paket data ini di dalam protokol AX.25 disebut sebagai *Frame*.

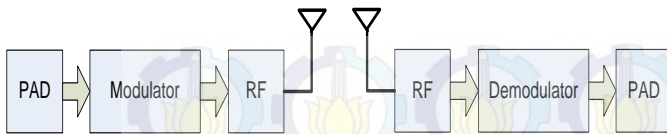
Protokol AX.25 ini akan diimplementasikan ke dalam sebuah mikrokontroler. Penggunaan mikrokontroler untuk implementasi protokol AX.25 dinilai mampu dalam hal diprogram untuk mengirim atau menerima data selayaknya sebuah komputer biasa, selain itu juga kemampuannya untuk portabel selama masih terdapat sumber tegangan untuk operasinya dan juga bentuknya yang kecil. Data dari mikrokontroler yang telah berbentuk *frame* AX.25 akan diteruskan ke dalam sub-modul protokol FX.25 yang berfungsi sebagai pendeteksi dan perbaikan kesalahan data yang muncul di dalam proses transmisi.

## II. SISTEM KOMUNIKASI DATA PROTOKOL AX.25

### A. Model Komunikasi Data

Mikrokontroler berfungsi sebagai otak dari modul PAD. Modul ini menerima input data karakter dengan menggunakan sebuah *keyboard* dan menampilkan data yang dimasukkan ke dalam sebuah layar LCD. Di dalam modul PAD terdapat 2 sub-bagian yaitu bagian protokol AX.25 dan bagian protokol FX.25. Data yang telah dimasukkan kemudian dikapsulasi ke dalam bentuk paket data yang sesuai dengan protokol AX.25 pada bagian protokol AX.25.

Data yang telah dikapsulasi sesuai dengan protokol AX.25 selanjutnya akan diteruskan ke bagian protokol FX.25 dengan fungsi sebagai pendeteksi dan pengoreksi kesalahan. Setelah melalui sub-bagian tersebut maka diteruskan ke bagian modulator untuk dimodulasikan. Setelah melalui modulator maka data yang telah berbentuk sinyal akan dilewatkan ke modul RF untuk selanjutnya dikirimkan melalui gelombang radio.



**Gambar 2.1. Model sistem komunikasi data protokol AX.25**

**B. Model Protokol AX.25**

Protokol AX.25 jika dipandang melalui OSI *Reference Layer* berada pada lapisan 1 dan 2.

Layer	Function(s)
Data Link (2)	(DLSAP) Segmenter Management
	Data Link Data Link
	Link Multiplexer
Physical (1)	Physical
	Silicon/Radio

**Gambar 2.2 Model Protokol AX.25**

Sebagai *layer 2* protokol ini bertanggungjawab untuk mengatur bagaimana *link* terbentuk, proses pertukaran informasi dan pemutusan *link*.

**C. Bentuk Frame AX.25**

Protokol AX.25 mempunyai bentuk data yang disebut *Frame*. Selain itu terdapat 3 tipe *frame* yang ada di dalam protokol AX.25 yaitu: *Information Frame (I Frame)*, *Supervisory Frame (S Frame)* dan *Unnumbered Frame (U Frame)*. Bentuk dari setiap tipe *frame* adalah sebagai berikut.

Flag	Address	Control	Info	FCS	Flag
01111110	112/224 Bits	8/16 Bits	N*8 Bits	16 Bits	01111110

**Gambar 2.3 Konstruksi U dan S frame**

Flag	Address	Control	PID	Info	FCS	Flag
01111110	112/224 Bits	8/16 Bits	8 Bits	N*8 Bits	16 Bits	01111110

**Gambar 2.4 Konstruksi I frame**

Setiap tipe *frame* bisa digunakan untuk mengirim informasi. Perbedaan dari masing-masing tipe *frame* adalah fungsi khususnya yaitu pada *I frame* untuk mengirim informasi dengan urutan untuk data teks yang panjang, *S frame* untuk mengirim data kontrol dari protokol AX.25 itu sendiri dan *U frame* untuk mengirim data informasi tanpa urutan untuk data teks yang pendek.

III. PERANCANGAN DAN IMPLEMENTASI

**Perancangan Perangkat Keras**

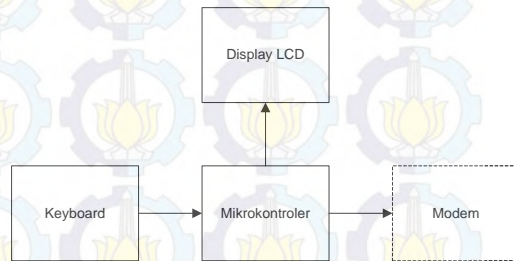
Pada penelitian ini akan dibuat sebuah perangkat keras dan perangkat lunak. Perangkat keras terdiri dari sebuah mikrokontroler Atmega 1280, rangkaian *keyboard* dan rangkaian LCD. Sedangkan perangkat lunak sendiri adalah

sebuah program yang berupa implementasi protokol AX.25 untuk diprogram ke dalam mikrokontroler.

Spesifikasi umum sistem ini adalah mampu membangun koneksi dengan satelit, melakukan pertukaran informasi, dan memutuskan koneksi. Untuk spesifikasi khusus sistem ini adalah mampu untuk mengirim data teks sebanyak 500 karakter, serta membentuk data teks tersebut menjadi beberapa *frame* karena setiap *frame* dibatasi hanya sepanjang 239 byte untuk diteruskan dan diproses oleh sub-modul berikutnya yaitu sub-modul protokol FX.25.

Untuk rangkaian LCD hanya menggunakan beberapa komponen seperti resistor, dioda, dan potensiometer serta sebuah sumber tegangan yang dihubungkan dengan pin-pin LCD untuk dapat beroperasi. Sedangkan pada rangkaian *keyboard* dipilih menggunakan *keyboard* dengan konektor PS/2. Untuk implementasi kedua rangkaian dan protokol AX.25 digunakan *software* CodeVision AVR sebagai kompilernya.

Gambar 3.1 akan menjelaskan diagram blok dari modul PAD yang akan dirancang.



**Gambar 3.1 Diagram blok modul PAD**

**Perangkat Lunak**

Pada perancangan program akan didesain dengan kemampuan menerima data melalui rangkaian *keyboard* dan menampilkan data melalui rangkaian LCD. Selain itu program akan mengkapsulasi data yang dimasukkan dari *keyboard* ke dalam bentuk *frame* AX.25 sesuai dengan konstruksi *frame* yang digunakan. Secara umumnya sebuah *frame* akan terdiri dari sub-*frame* *flag*, *address*, *control*, *info*, dan FCS.

*Flag* terdiri dari 1 byte yang digunakan sebagai penanda awal dan akhir dari *frame*. *Flag* sendiri mempunyai nilai 01111110 dalam biner atau 7E dalam hex.

*Address* sendiri terdiri dari 2 atau 3 alamat sesuai dengan sistem, apabila tanpa *repeater* maka terdapat 2 alamat sedangkan apabila dengan *repeater* akan terdapat 3 alamat. Di dalam sistem akan digunakan contoh *address* dengan *repeater* dengan aturan alamat penerima, alamat sumber dan alamat *repeater*. Setiap alamat baik alamat pengirim, penerima dan *repeater* terdiri dari 7 byte data yang dibagi ke dalam 6 byte untuk *callsign* dan 1 byte untuk SSID dari *callsign* tersebut.

Sub-*frame* *control* akan berisi data kontrol mode protokol AX.25 yang digunakan dan terdiri dari 1 byte yang menunjukkan perintah *control* yang digunakan.

Control Field Type	Control-Field Bits				
	7	6	5	4	3 2 1 0
I Frame	N(R)	P	N(S)	0	
S Frame	N(R)	P/F	S S	0 1	
U Frame	M M M	P/F	M M	1 1	

Gambar 3.2 Format Sub-frame Control

Sub-frame PID (*Protocol Identifier field*) berisi mengenai fungsi dari frame AX.25 yang digunakan. Pada perancangan field ini diberi nilai F0 hexa karena tidak menggunakan fungsi layer 3 seperti yang didefinisikan pada gambar 2.6.

HEX	M	Z	Translation
**	yy01yyyy		AX.25 layer 3 implemented.
**	yy10yyyy		AX.25 layer 3 implemented.
0x01	00000001		ISO 8208/CCITT X.25 PLP
0x06	00000110		Compressed TCP/IP packet. Van Jacobson (RFC 1144)
0x07	00000111		Uncompressed TCP/IP packet. Van Jacobson (RFC 1144)
0x08	00001000		Segmentation fragment
0xC3	11000011		TEXNET datagram protocol
0xC4	11000100		Link Quality Protocol
0xCA	11001010		Appletalk
0xCB	11001011		Appletalk ARP
0xCC	11001100		ARPA Internet Protocol
0xCD	11001101		ARPA Address resolution
0xCE	11001110		FlexNet
0xCF	11001111		NET/ROM
0xF0	11110000		No layer 3 protocol implemented.
0xFF	11111111		Escape character. Next octet contains more Level 3 protocol information.
Escape character. Next octet contains more Level 3 protocol information.	00001000		

Gambar 3.3 Definisi nilai dari sub-frame PID

Sub-frame info akan berisi data teks yang akan dikirimkan. Besar dari sub-frame ini adalah sebesar 212 byte untuk menjamin panjang maksimal dari setiap frame AX.25 adalah 239 byte.

Pada sub-frame FCS (*Frame Check Sequence*) berisi sebuah CRC (*Cyclic Redundancy Check*) yang berfungsi sebagai pendeteksi kesalahan. Dengan adanya FCS maka data dapat diperiksa oleh penerima apakah terjadi *data error* pada saat transmisi, apabila terdapat kesalahan maka penerima akan meminta pengirim mengirim ulang. Namun karena frame AX.25 ini akan diteruskan ke dalam sub-bagian modul dengan protokol FX.25 yang berfungsi untuk mendeteksi dan memperbaiki kesalahan, dimana mempunyai cara kerja yang sama namun dengan tambahan pengkoreksi kesalahan dari AX.25 maka pada FCS bisa diberi nilai bebas. Namun pada frame AX.25 FCS sendiri terdiri dari 4 byte data dimana byte LSB akan dikirimkan lebih dahulu dari pada byte MSB. Dalam perancangan akan menggunakan nilai FF dalam hexa pada byte LSB dan MSB. Berikut bagian program yang menginisialisasikan nilai setiap sub-frame, kecuali sub-frame control dan sub-frame info yang isinya akan sesuai dengan data masukan keyboard, ke dalam bentuk hex dengan

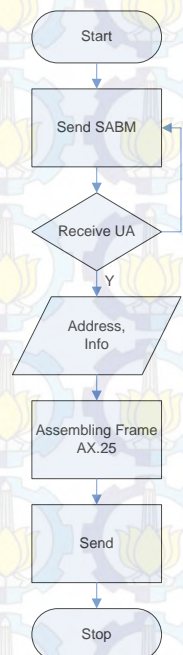
menggunakan tipe *Information frame*:

```
char flag=0xFE;
char control=0x03;
char pid=0xF0;
char fcslo=0xFF;
char fcshi=0xFF;
```

Untuk membangun sebuah link dengan satelit maka pada awalnya dikirimkan frame *Supervisory* dengan sub-frame control berisi perintah SABME yang berfungsi untuk meminta link dengan satelit. Setelah itu timer pada mikrokontroler akan berjalan untuk menunggu autentikasi link dari satelit yang berupa S frame dengan sub-frame control berisi perintah UA. Apabila selama timer berjalan hingga berhenti tidak ada balasan dari satelit maka permintaan link akan dikirimkan terus menerus. Setelah link terhubung maka kemudian akan dikirimkan frame informasi. Rutin di atas umunya disebut sebagai rutin *handshaking*.

### AX.25 Encoder

Dengan sub-frame info berisi data masukan dari keyboard maka untuk mengirimkan frame AX.25 bisa dilakukan dengan membuat suatu variabel yang akan menggabungkan semua sub-frame menjadi sebuah frame yang utuh. Berikut adalah flowchart dan program utama yang bekerja terus menerus, memanggil anak program untuk mengirimkan atau menerima frame AX.25.



Gambar 3.4 Flowchart AX.25 Encoding

```
main(){
  transmit=1;
  while (1){
    kirim_frame(SABME);
    for (i=0;i<200;i++){
```

```

terima_frame();
if (terima_frame==UA) {
    break; }
delay_ms(25);}
if (transmit==1) {kirim_frame(frame_ax);}
else {terima_frame();}
}

```

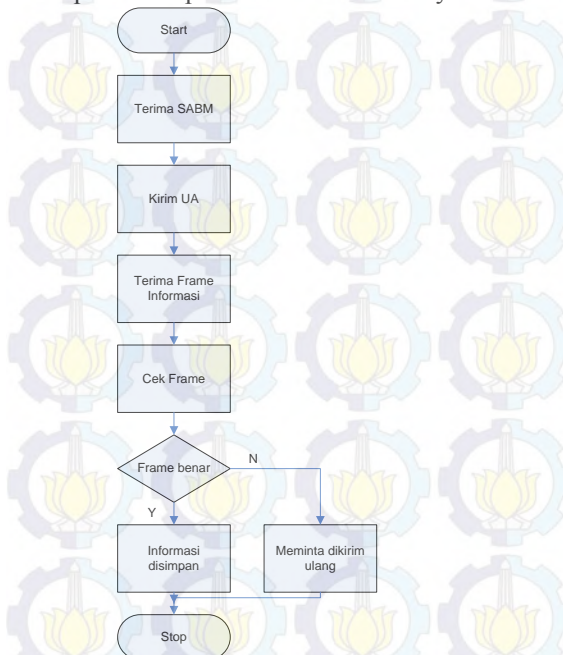
Dari program utama di atas menggunakan anak program yang bernama kirim\_frame yang memiliki fungsi mengirimkan frame\_ax yang telah lengkap menuju ke sub-modul dengan protokol FX.25. Dimana variabel frame\_ax sendiri adalah hasil gabungan dari setiap sub-frame mulai dari flag, address, control, pid, info, FCS, dan flag.

Variabel transmit sendiri berfungsi untuk penanda mode yang digunakan. Untuk sistem ini mode yang digunakan adalah half duplex sehingga hanya dapat mengirim atau menerima saja. Variabel transmit=1 menunjukkan bahwa modul sedang digunakan untuk mengirim data sedang apabila variabel transmit=0 maka modul digunakan untuk menerima data.

### AX.25 Decoder

Untuk men-decode-kan frame yang diterima maka seluruh panjang frame akan diterima dan disimpan terlebih dahulu ke dalam memory mikrokontroler. Kemudian akan diperiksa tipe dari frame tersebut dan isinya. Apabila bentuk frame tersebut benar maka data akan disimpan dan ditampilkan ke layar LCD, sedangkan apabila salah maka akan dikirim permintaan untuk mengirim ulang. Berikut adalah flowchart dan program dari AX.25 decoder.

Dari program tersebut fungsi variabel count adalah sebagai penghitung jumlah flag di dalam frame yang diterima. Apabila flag ditemukan mempunyai jumlah sebanyak 2 buah maka frame akan disimpan dan diperiksa data informasinya.



Gambar 3.5 Flowchart AX.25 Decoding

```

main(){
    transmit=0;
    while(1){
        terima_frame();
        if (terima_frame==0x7E) count=count+1;
        if (count==2) break;}
        frame_ax=terima_frame;
    }
}

```

Dari flowchart tersebut maka akan didesain sebuah protokol AX.25 yang mampu kapsulasi dan enkapsulasi data. Untuk protokol AX.25 sendiri menggunakan 2 tipe frame yaitu :

#### a) Unnumbered frame

Digunakan pada proses handshaking saja dengan aturan sebagai berikut :

- Field address berisi alamat amatir stasiun tujuan, alamat amatir dari stasiun pengirim dan alamat amatir dari stasiun repeater yang dimasukkan melalui keyboard.
- Field Control bernilai hexa 7F untuk mengirimkan perintah SABM.
- Field PID bernilai hexa F0 karena hanya menggunakan layer 2.
- Field Info diset kosong atau diisi nilai spasi (20 hexa).
- Field FCS diset kosong atau diisi nilai hexa FF.

#### b) Information frame

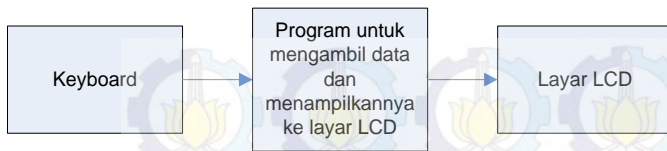
Digunakan untuk mengirimkan informasi apabila proses handshaking telah terjadi dengan aturan sebagai berikut :

- Field address berisi alamat amatir stasiun tujuan, alamat amatir dari stasiun pengirim dan alamat amatir dari stasiun repeater yang dimasukkan melalui keyboard.
- Field Control bernilai hexa 02 untuk menginformasikan bahwa data yang dikirim merupakan data dengan urutan 1.
- Field PID bernilai hexa F0 karena hanya menggunakan layer 2.
- Field Info diset kosong atau diisi nilai spasi (20 hexa).
- Field FCS diset kosong atau diisi nilai hexa FF.

## IV. METODE PENGUJIAN SISTEM

### Pengujian Algoritma

Pada bagian ini akan diuji kemampuan dari algoritma dari modul PAD untuk mengkapsulasi data. Algoritma yang akan diuji antara lain algoritma untuk mengambil data dari keyboard dan langsung menampilkannya di layar LCD. Kemudian algoritma untuk mengkapsulasi setiap data yang dimasukkan melalui keyboard ke dalam setiap field dan subfield dari frame AX.25. Hasil dari kapsulasi akan ditampilkan pada hyperterminal. Gambar 4.1 dan gambar 4.2 akan menjelaskan mengenai metode pengujian yang dilakukan.



**Gambar 4.1** Diagram blok pengujian algoritma pengambilan data



**Gambar 4.2** Diagram blok pengujian algoritma kapsulasi data

### Pengujian Algoritma Pemasukan dan Penampil Data

Hasil dari pengujian algoritma pengambilan data dari *keyboard* akan ditunjukkan pada gambar 4.3. Pada gambar tersebut LCD akan menampilkan karakter yang bertuliskan “MASUKKAN ALAMAT PENERIMA”, kemudian data yang dimasukkan dari *keyboard* akan ditampilkan pada bawah karakter tersebut. Setiap data yang ditampilkan merupakan data yang diambil dari *keyboard*.



**Gambar 4.3** Hasil pengujian pengambilan dan penampil data

### Pengujian Algoritma Kapsulasi Data

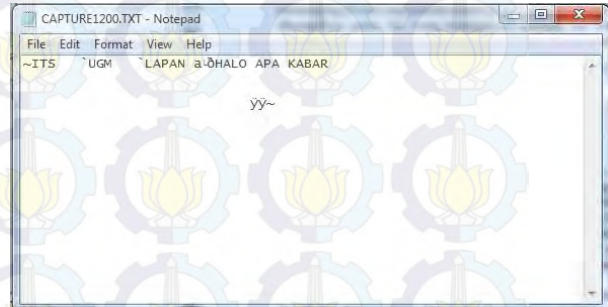
Pengujian algoritma ini dilakukan dengan melihat hasil pembacaan pada program *hyperterminal*. Di dalam *hyperterminal* sendiri terdapat perintah khusus yang mampu menangkap setiap data yang masuk.

Dari program *hyperterminal* sendiri terdapat fungsi untuk menangkap *frame* yang masuk melalui *port*. *Frame* yang ditangkap akan ditampilkan pada *file* yang mempunyai ekstensi *.txt*. Gambar 4.11 adalah *file* dengan ekstensi *.txt* yang telah menangkap *frame* AX.25. Pada *frame* yang akan dikirimkan kita atur setiap *field* dengan parameter sebagai berikut :

- ITS sebagai alamat tujuan
- SSID alamat tujuan adalah 0 (60 hexa)
- UGM sebagai alamat pengirim
- SSID alamat pengirim adalah 0 (60 hexa)
- LAPAN adalah alamat *repeater*
- SSID *repeater* adalah 0 (61 hexa)
- *Control field* bernilai 03 hexa
- *PID field* bernilai F0 hexa

- *Info field* berisi kalimat “HALO APA KABAR”
- *Info field* sepanjang 212 karakter sehingga setelah kalimat di atas dikirim akan diberi spasi (20 hexa) secara otomatis hingga panjang total mencapai 212 karakter.
- *FCS field* diberi nilai FFFF hexa.

Pada gambar 4.4 dan 4.5 merupakan hasil pembacaan data yang telah dikapsulasi pada program *hyperterminal* yang telah *di-capture* ke dalam file berekstensi *.txt*.



**Gambar 4.4** Frame yang ditangkap oleh *hyperterminal*

### Pengujian Kecepatan Proses

Pada bagian ini kecepatan proses dari modul PAD untuk dapat mengkapsulasi data menjadi bentuk *frame* AX.25. metode pengukuran yang digunakan adalah dengan menggunakan *timer* mikrokontroler ATmega1280 untuk menghitung waktu pemrosesan. *Timer* akan dimulai pada saat data mulai diproses kapsulasi dan berhenti pada saat data telah dikapsulasi. Hasil dari *timer* akan ditampilkan pada LCD dalam bentuk detik. Tabel 4.1 menjelaskan mengenai kecepatan proses kapsulasi.

**Tabel 4.1** Hasil pengujian kecepatan proses

Pengiriman	Waktu (detik)
1	0.000157332
2	0.000158291
3	0.00015867
4	0.000159183
5	0.00015938
6	0.00015961
7	0.000159978
8	0.000161032
9	0.000165982
10	0.000169742
11	0.000171341
12	0.000175974
13	0.000179179
14	0.000182093
15	0.000218402

Dari hasil tabel 4.1 didapat nilai rata-rata dari waktu proses adalah sebesar 0,000169079 detik atau 169 mikro detik pada setiap kali proses kapsulasi terjadi.

### Pengujian dengan *Baud rate* yang berubah-ubah

Pada pengujian *baud rate* pada saat pengiriman data

diubah-ubah nilainya. Pengujian ini bertujuan untuk mengetahui kemampuan mikrokontroler untuk mengirimkan data dengan modem yang memiliki *baud rate* terima yang berbeda-beda. Pengujian dilakukan dengan mengirimkan *frame AX.25* yang berisi informasi tetap namun dengan *baud rate* mikrokontroler diubah-ubah.

Untuk mengetahui hasil pengiriman dilakukan dengan menggunakan *hyperterminal*. Selain itu juga dengan menggunakan fungsi CodeVisionAVR yang bernama Code Wizard AVR tersedia informasi mengenai kesalahan yang mungkin terjadi pada *baud rate* tertentu.

Gambar 4.5 menunjukkan hasil pengujian dengan *hyperterminal*. *Frame* yang sama dikirimkan pada *baud rate* 1200 sebanyak 12 kali.



**Gambar 4.5** Hasil pengujian pada *hyperterminal* dengan *baud rate* 1200 sebanyak 12 kali pengiriman

Dengan menggunakan fungsi yang sama dilakukan pengujian pada *baud rate* yang berbeda-beda. Data kemungkinan kesalahan data dari setiap *baud rate* dapat dilihat pada tabel 4.2.

**Tabel 4.2** Nilai kesalahan data yang mungkin terjadi pada setiap *baud rate*

<i>Baud Rate</i>	Prosentase Kesalahan (%)
110	100
300	0,0
600	0,0
1200	0,1
2400	0,2
4800	0,2
9600	0,2
14400	0,8
19200	0,2
38400	0,2
56000	0,8
57600	3,7
115200	7,8

Dari tabel 4.2 untuk *baud rate* 110 mempunyai prosentase

kesalahan data sebesar 100 % dikarenakan nilai dari *clock* mikroprosesor terlalu tinggi untuk digunakan pada *baud rate* 110. Sedangkan nilai prosentase kesalahan terkecil terdapat pada *baud rate* 300 dan 600 dikarenakan pengiriman data mempunyai kecepatan yang rendah sehingga semua data yang dikirim dapat diterima dengan sempurna. Pada *baud rate* 1200 dan 9600 mempunyai prosentase kesalahan 0,1% dan 0,2% masih baik untuk digunakan pada modem stasiun bumi portabel.

## V. KESIMPULAN

Berdasarkan hasil dari algoritma yang telah diimplementasikan dan hasil analisa pada pengujian kapsulasi data yang telah dilakukan, beberapa hal yang dapat disimpulkan adalah:

1. Modul mampu menerima masukan data melalui sebuah *keyboard* portabel dan akan disimpan pada variabel khusus untuk dilakukan proses kapsulasi. Sedikit kekurangan dalah fungsi standar *keyboard* untuk menghapus karakter (*backspace*) pada saat pengetikan terjadi tidak dapat dilakukan karena mempunyai alamat memory yang sama dengan fungsi khusus mikrokontroler sehingga data tidak dapat dikapsulasi.
2. LCD mampu menampilkan data yang dimasukkan melalui *keyboard*. Data ditampilkan hampir bersamaan dengan saat penekanan tombol. LCD mampu menampilkan karakter sebanyak 500 buah.
3. Modul dapat digunakan untuk proses kapsulasi data. Hasil kapsulasi adalah *frame AX.25* dengan panjang 239 byte.
4. Waktu proses kapsulasi data yang cepat yaitu kurang dari 1 detik pada setiap proses.
5. Dalam hal koneksi dengan modem, modul dapat digunakan pada berbagai macam modem dengan *baud rate* yang berbeda-beda.

## DAFTAR PUSTAKA

- [1] "Tentang INSPIRE" : <http://inspire.or.id/web/>
- [2] "Satelit IINUSAT" : [http://www.nanosat.jp/3rd/sozai\\_report/Day\\_1\\_0202\\_Tri\\_Priyambodo\\_Gadjah\\_Mada\\_University/NSS-03-Day\\_1\\_0202\\_Tri\\_Priyambodo.pdf](http://www.nanosat.jp/3rd/sozai_report/Day_1_0202_Tri_Priyambodo_Gadjah_Mada_University/NSS-03-Day_1_0202_Tri_Priyambodo.pdf)
- [3] "AX.25 Protocol" : <http://www.tapr.org/pdf/AX25.2.2.pdf>
- [4] "APRS (Automatic Packet Reporting System)" : <http://www.qsl.net/yb0emj/aprs-i.html>
- [5] "How to Send AX.25 using Unexpensive Microcontroler" : [www.tnc-x.com/dcc.doc](http://www.tnc-x.com/dcc.doc)
- [6] ISO 3309., Information technology -- Telecommunications and information exchange between systems -- High-level data link control (HDLC) procedures -- Frame structure. 23 Desember 1993
- [7] ISO 4335., Information technology --- Telecommunications and information exchange between systems -- High-level data link control (HDLC) procedures -- Elements of procedures. 1993
- [8] ISO 7809., Information technology -- Telecommunications and information exchange between systems -- High-level

data link control (HDLC) procedures -- Classes of procedures. 1993

- [9] ITU-T Q.920., ISDN user-network interface data link layer - General aspects. Maret 1993.
- [10] ITU-T Q.921., ISDN user-network interface - Data link layer specification. Maret 1993.
- [11] Andrianto, Hari. *Pemrograman Mikrokontroler AVR ATmega16 Menggunakan Bahasa C (CodeVision AVR)*. Informatika, Bandung. Bab 2. 2008.
- [12] "IBM Keyboard" : [www.quadibloc.com/comp/kybint.htm](http://www.quadibloc.com/comp/kybint.htm)
- [13] Usman. "Teknik Antarmuka dan Pemrograman Mikrokontroler AT89S52". Penerbit ANDI, Yogyakarta. Bab 3. 2008.
- [14] BARNETT, RICHARD. H., O'CULL, LARRY., COX, SARAH., "EMBEDDED C PROGRAMMING AND THE ATMEL AVR". CENGAGE LEARNING. CH. 3. 2006.

#### RIWAYAT PENULIS



Dicky Rismawan Raharjo, lahir di Surabaya 18 Januari 1987, merupakan anak pertama dari tiga bersaudara pasangan Mardjuki, MS dan Dra. Sulistyowati.. Penulis menamatkan SD, SMP dan SMA di kota Surabaya kemudian melanjutkan studinya pada tahun 2005 di Institut Teknologi Sepuluh Nopember Surabaya tepatnya pada D3 Teknik Elektro bidang studi

Elektro Industri dan lulus pada tahun 2009. Selanjutnya penulis meneruskan studi sarjana di Teknik Elektro ITS pada Januari 2010, kemudian fokus pada bidang studi Telekomunikasi Multimedia dan menjadi anggota tim riset stasiun bumi portable ITS untuk satelit Inusat-01.