

Perancangan Simulator PAD (*Packet Assembler Disassembler*) AX.25 Pada Sistem Komunikasi Satelit IINUSAT-01 Untuk *Attitude monitoring* Berbasis FPGA (*Field Programmable Gate Array*)

Gilang Almaghribi S. Putra, Gamantyo Hendranto, Fitri Adi Iskandarianto

Jurusan Teknik Fisika, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: fiskandarianto@ep.its.ac.id

Abstrak—Sistem komunikasi satelit terjadi antara satelit yang berada di luar angkasa dengan stasiun bumi. Salah satu kebutuhan dari satelit adalah *attitude monitoring* yang berfungsi untuk memantau perilaku satelit. Data pemantauan perilaku satelit akan dikirimkan oleh PAD (*Packet Assembler Disassembler*) dalam bentuk paket data. Untuk komunikasi data dibutuhkan sebuah protokol yang mengatur tata cara komunikasi data. Protokol yang digunakan untuk komunikasi data adalah protokol AX.25. Protokol ini digunakan untuk mengirimkan informasi yang berupa teks dan data *attitude monitoring*. Dalam penelitian ini, protokol AX.25 ini akan diimplementasikan ke dalam perangkat lunak dan perangkat keras berbasis FPGA (*Field Programmable Gate Array*). *Attitude monitoring* PAD AX.25 yang diimplementasikan pada FPGA mampu mengirim paket data dalam waktu 0,449 detik pada *baud rate* 1200, dan semakin cepatnya pengiriman paket data sebanding dengan kenaikan *baud rate*. PAD AX.25 ini mampu mengirimkan paket data secara benar dengan error 0% pada *baud rate* 1200 – 9600. Sedangkan error koreksi sensor *attitude monitoring* yang digunakan sebesar $\pm 0,072g$.

Kata Kunci—Satelit, *Attitude monitoring*, PAD AX.25, FPGA.

I. PENDAHULUAN

Berawal dari INSPIRE (*Indonesian Nano-Satellite Platform Initiative for Research & Education*) yaitu suatu inisiatif proyek nir-laba yang bertujuan untuk membangun & mengembangkan platform teknologi satelit (satelit - nano khususnya) di kalangan perguruan tinggi di Indonesia, dengan menempatkan mahasiswa sebagai motor/penggerak utamanya. Maka dibentuklah IINUSAT (*Indonesian Inter university Satellite*) yang saat ini ada tujuh instansi yang bekerjasama dalam program tersebut yaitu UGM, ITS, PENS, IT Telkom, ITB, UI, dan LAPAN. Di dalam proyek ini akan dikembangkan sebuah satelit nano beserta stasiun buminya untuk dapat saling berkomunikasi [1].

Pada sistem komunikasi satelit, satelit berfungsi sebagai penerima data. Satelit Iinusat-1 ini menggunakan protokol AX.25 sebagai protokol komunikasi yang digunakan untuk komunikasi data. AX.25 merupakan sebuah protokol yang mengatur mengenai proses terjadinya koneksi *link*, metode pertukaran data dan pemutusan hubungan *link* [2]. Paket data ini di dalam protokol AX.25 disebut sebagai *Frame*. Protokol

AX.25 ini akan diimplementasikan ke dalam sebuah FPGA. Penggunaan FPGA untuk implementasi protokol AX.25 dinilai mampu untuk mengirim atau menerima data layaknya sebuah komputer biasa dan konsumsi dayanya kecil.

Satelit mengorbit di luar angkasa dengan pergerakan orbitnya yang beragam, baik berputar maupun bergeser. Namun pergerakan satelit tidak bisa dipantau secara langsung dari muka bumi dengan mata. Oleh karena itu perlu adanya sensor yang diletakkan di satelit agar bisa mendeteksi perubahan posisi dari satelit dan kemudian data perubahan posisi tersebut dikirimkan ke stasiun bumi, sehingga perubahan posisinya atau perilaku satelit dapat dipantau dari bumi.

Pada penelitian ini akan dilakukan perancangan protokol komunikasi AX.25 pada FPGA dengan fungsi utama untuk menerima data teks dan mengirim konfirmasi *link* dari stasiun bumi. Selain itu akan dirancang pula simulator *attitude monitoring* satelit untuk memantau perilaku satelit saat mengorbit di luar angkasa.

II. TINJAUAN PUSTAKA

A. Satelit

Satelit adalah benda yang mengelilingi planet dengan periode revolusi dan rotasi tertentu dan memiliki orbit peredaran sendiri. Satelit bergerak berdasarkan orbitnya, Masalah utama dalam lintasan satelit adalah bagaimana suatu satelit bergerak pada orbitnya tanpa jatuh ke bumi. Jadi variabel yang diperhatikan yaitu kecepatan, arah dan ketinggian orbitnya [8]. Salah satu satelit buatan adalah satelit komunikasi. Syarat utama suatu satelit komunikasi adalah satelit tersebut harus dapat diakses atau terlihat setiap saat dari suatu titik dipermukaan bumi. Dalam sistem komunikasi satelit terdapat dua bagian penting, yaitu *space segment* (bagian yang berada di angkasa) dan *ground segment* (stasiun bumi).

B. Attitude monitoring

Attitude monitoring berarti pemantauan sikap atau perilaku, dalam hal ini perilaku yang akan dipantau adalah perilaku satelit yang mengorbit bumi. Fungsi dari *attitude monitoring* adalah untuk mengetahui perilaku satelit dalam

arah sumbu X,Y,dan Z. Setelah mengetahui perilakunya, maka akan diteruskan ke sub sistem ADCS (*Attitude Determination Control System*), sub sistem ini digunakan untuk mendukung tercapainya misi satelit yang berorientasi pada sikap satelit dalam orbitnya agar satelit selalu terkendali sesuai dengan misi operasinya serta mempertahankan sikap satelit secara akurat [8].

C. PAD (Packet Assembler/Disassembler)

Adalah alat yang umumnya ditemukan di jaringan X.25. PAD digunakan ketika sebuah perangkat DTE (*Data Terminal Equipment*), seperti karakter-modus terminal, terlalu sederhana untuk mengimplementasikan fungsionalitas X.25 secara penuh. PAD ini terletak antara perangkat DTE dan perangkat DCE (*Data Circuit-terminating Equipment*), dan melakukan tiga fungsi utama: *buffering* (menyimpan data sampai perangkat yang siap untuk proses itu), paket perakitan, dan pembongkaran paket. PAD *buffer* data yang dikirim ke atau dari perangkat DTE. Ini juga merakit data ke dalam paket dan meneruskan perangkat DCE (Ini termasuk menambahkan sebuah *header X.25*.) Akhirnya, *disassembles* PAD masuk sebelum meneruskan paket data ke DTE. (Termasuk menghapus *header X.25*) [5].

D. Protokol AX.25

Pada protokol AX.25 ini, data yang akan ditransmisikan melalui radio dibentuk ke dalam blok-blok data kecil yang disebut *frame*. Terdapat 3 tipe umum dari *frame* AX.25,yaitu :

- a) *Information frame (I frame)*;
- b) *Supervisory frame (S frame)*;
- c) *Unnumbered frame (U frame)*.

Masing-masing *frame* ini terdiri dari beberapa grup kecil yang disebut *field*. tabel 1 dan tabel 2 akan mengilustrasikan 3 tipe dasar dari *frame*. Catat bahwa bit pertama yang akan ditransmisikan berada pada bagian kiri.

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

Tabel 1. Konstruksi U dan S *frame* [2]

Flag	Address	Contro l	PI D	Info	FCS	Flag
01111111 0	112/224 Bits	8/16 Bits	8 bits	N*8 Bits	16 Bits	01111111 0

Tabel 2. Konstruksi I *frame* [2]

E. FPGA

Field-Programmable Gate Array (FPGA) adalah komponen elektronika dan semikonduktor yang mempunyai komponen gerbang terprogram (*programmable logic*) dan sambungan terprogram. Blok-blok komponen di dalam *FPGA* bisa juga mengandung elemen memori (*register*) mulai dari flip-flop sampai pada RAM (*Random Access Memory*) [3].

FPGA ini sendiri juga merupakan *IC* bertipe *Hardware Definition Language (HDL)* yang mana dipakai dapat merancang *hardware* sesuai yang diinginkan di dalam *IC* *FPGA*. Perkembangan *FPGA* pada saat ini berlangsung

dengan cepat dan dewasa ini terdapat bermacam-macam keluarga *FPGA* dengan kebutuhan perancangan dan perangkat perancangan (*design tools*) yang berbeda [4].

F. VHDL

VHDL merupakan singkatan dari *VHSIC Hardware Description Language* (Bahasa Pemrograman *hardware* *VHSIC*). VHDL merupakan bahasa pemrograman *hardware*/alat/piranti keras yang memiliki fungsi tambahan dari bahasa pemrograman *hardware* sebelumnya yaitu PLD dan *netlist*. Hampir semua *software* VHDL synthesis mendukung bahasa pemrograman VHDL. Bahasa pemrograman VHDL menganut standar yang ditetapkan IEEE [3].

VHSIC sendiri merupakan *IC* dengan kecepatan sangat tinggi (*Very High Speed Integrated Circuit*). Bahasa pemrograman HDL dapat digunakan pada beberapa cara dan dapat dijadikan jalan alternatif untuk merepresentasikan diagram rangkaian untuk rangkaian digital ataupun untuk sebagian sub sistem rangkaian digital [6]. Dua cara untuk memodelkan sistem digital yaitu secara struktur maupun secara behavioral (penjelasan *output* sistem).

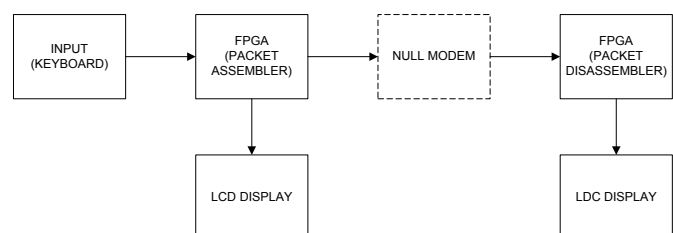
Cara penulisan *syntax* dalam bahasa pemrograman VHDL. VHDL didesain dengan prinsip yang diambil dari bahasa pemrograman *software* pascal. Desain dalam VHDL terdiri atas beberapa bagian utama diantaranya :

1. *Library (design unit)*
2. *Entity*
3. *Architecture*
4. *Signal*

III. PERANCANGAN PAD AX.25

A. Perangkat Keras PAD AX.25

Pada penelitian ini akan dibuat sebuah perangkat keras dan perangkat lunak. Perangkat keras terdiri dari dua buah *FPGA*, sebagai *transmitter* dan *receiver*, rangkaian *keyboard* dan rangkaian LCD. Sedangkan perangkat lunak sendiri adalah sebuah program yang berupa implementasi protokol AX.25 untuk diisikan ke dalam *FPGA*. 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. Untuk rangkaian LCD hanya menggunakan beberapa resistor dan 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 Quartus 2 Altera sebagai kompilernya.



Gambar 1. Blok Diagram Perancangan Perangkat Keras

FPGA mempunyai pengaturan pin yang akan digunakan untuk input/output pada rangkaian lainnya, pengaturan I/O pada FPGA ada di bagian *pin planner*, berikut ini adalah pengaturan pin I/O :

Tabel 3. Pengaturan pin-pin I/O pada FPGA

Node Name	Direction	Location	I/O Standard
data2lcd[7]	Output	PIN_F9	3.3-V LV...default
data2lcd[6]	Output	PIN_E8	3.3-V LV...default
data2lcd[5]	Output	PIN_E7	3.3-V LV...default
data2lcd[4]	Output	PIN_C8	3.3-V LV...default
data2lcd[3]	Output	PIN_A7	3.3-V LV...default
data2lcd[2]	Output	PIN_B7	3.3-V LV...default
data2lcd[1]	Output	PIN_B6	3.3-V LV...default
data2lcd[0]	Output	PIN_A5	3.3-V LV...default
lcd2e	Output	PIN_C11	3.3-V LV...default
lcd2rs	Output	PIN_D12	3.3-V LV...default
lcd2rw	Output	PIN_A12	3.3-V LV...default
led[7]	Output	PIN_L3	3.3-V LV...default
led[6]	Output	PIN_B1	3.3-V LV...default
led[5]	Output	PIN_F3	3.3-V LV...default
led[4]	Output	PIN_D1	3.3-V LV...default
led[3]	Output	PIN_A11	3.3-V LV...default
led[2]	Output	PIN_B13	3.3-V LV...default
led[1]	Output	PIN_A13	3.3-V LV...default
led[0]	Output	PIN_A15	3.3-V LV...default
mclk	Input	PIN_R8	3.3-V LV...default
out1	Output	PIN_C3	3.3-V LV...default
pin_name	Bidir	PIN_D3	3.3-V LV...default
ps2_clk	Bidir	PIN_A2	3.3-V LV...default
ps2_data	Bidir	PIN_A3	3.3-V LV...default
reset	Input	PIN_E1	3.3-V LV...default

‘Data2lcd[0-7]’ adalah pin *output* yang dihubungkan pada LCD karakter, dimana pin-pin tersebut mewakili 8 bit data yang akan ditampilkan. ‘Led[0-7]’ adalah pin *output* 8 bit yang dihubungkan pada led yang terdapat di FPGA, pin ini berfungsi untuk menunjukkan bilangan biner dari data 8 bit ASCII yang diketikkan pada *keyboard*, apabila bernilai 1 maka led akan menyala, dan apabila 0 maka led akan mati. LED ini juga sebagai validasi apakah karakter yang diketikkan pada *keyboard* sudah sesuai, yaitu dengan mengecek nilai binernya. Pin ‘lcd2e’ merupakan pin *enable* pada LCD, pin ‘lcd2rs’ adalah pin *register select* pada lcd, pin ‘lcd2rw’ adalah pin *lcd read/write*. Pin *bidir* ‘ps2_clk’ dan ‘ps2_data’ dihubungkan ke pin port ps2 *keyboard* sebagai sinkronisasi data dan *clock* pada *keyboard*. Pin ‘mclk’ adalah pin *input clock* yang berasal dari internal *clock* FPGA sebesar 50MHz, pin ini sebagai pensuplai *clock* rangkaian secara keseluruhan. Pin ‘reset’ dihubungkan pada *push button* FPGA yang berfungsi untuk *mereset* program kembali ke awal.



Gambar 2. Rangkaian Total PAD AX.25

B. Perangkat Keras Simulator Attitude monitoring

Accelerometer berfungsi sebagai sensor percepatan yang dapat mengetahui perilaku dari satelit, *accelerometer* ini dihubungkan ke FPGA sebagai pemroses sinyalnya, kemudian dari FPGA dihubungkan ke GUI (*Graphic User Interface*) untuk melihat perilaku dari satelit. *Accelerometer* yang digunakan adalah MMA7361L, merupakan *accelerometer* 3-axis dengan resolusi tinggi dengan resolusi mencapai 10 bit pada ±6g. *Accelerometer* digital ini dapat dikoneksikan melalui SPI (3-wire) atau I2C (2-wire). Data *Attitude Monitoring* akan dikirimkan melalui PAD AX.25.

C. Perangkat Lunak

Pada perancangan program akan didesain dengan kemampuan menerima data melalui rangkaian *keyboard* dan menampilkan data melalui rangkaian LCD. Selain itu program akan mampu membuat *frame* AX.25 dari data yang dimasukkan dari *keyboard* 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 dinamakan 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 Sub-*frame* *info* akan berisi data teks yang akan dikirimkan. Besar dari sub-*frame* ini adalah sebesar 225 byte untuk menjamin panjang maksimal dari setiap *frame* AX.25 adalah 256 byte. Pada sub-*frame* FCS (*Frame Check Sequence*) berisi sebuah CRC (*Cyclic Redundancy Check*) yang berfungsi sebagai pendeteksi kesalahan. Program PAD secara umum dibagi menjadi tiga bagian, antara lain :

1. Pengambilan Data Karakter dan Menampilkan ke LCD

Pengambilan data dimulai dengan memasukkan karakter dari *keyboard* yang dihubungkan dengan rangkaian PS2 ke FPGA, setelah itu dihubungkan dengan rangkaian LCD untuk menampilkan karakter yang dimasukkan tadi. LCD yang digunakan adalah LCD 2x16 karakter, yang artinya 2 baris dan 16 kolom, dimana setiap kolom akan menampilkan 1 karakter dan akan terus bergeser sesuai dengan penambahan karakter, jadi jumlah maksimal karakter yang bisa dimuat adalah 32. Apabila jumlah karakter yang dimasukkan sudah 16, maka penampilan pada LCD karakter akan berpindah ke baris kedua. Namun jika jumlah karakter yang dimasukkan melebihi 32 karakter, maka LCD akan menghapus semua karakter yang ditampilkan dan akan memulai tampilan baru pada kolom pertama baris pertama untuk melanjutkan karakter ke-33 dan seterusnya.

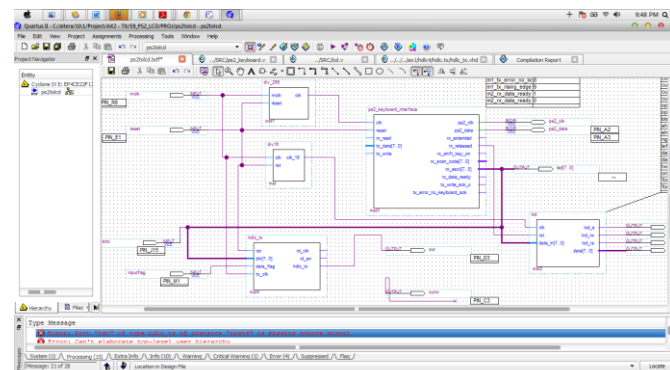
2. Packing data ke *frame* AX.25

Data karakter 8 bit ASCII yang dikirimkan melalui keyboard akan diproses pada bagian *packing* data AX.25 di *Transmitter*. Data tersebut akan dibungkus oleh *frame* AX.25, data ini akan dibungkus oleh *field flag*, *address*, *control*, PID dan FCS dengan format *field* data AX.25. Setelah data ini *dipacking*, maka siap untuk dikirimkan.

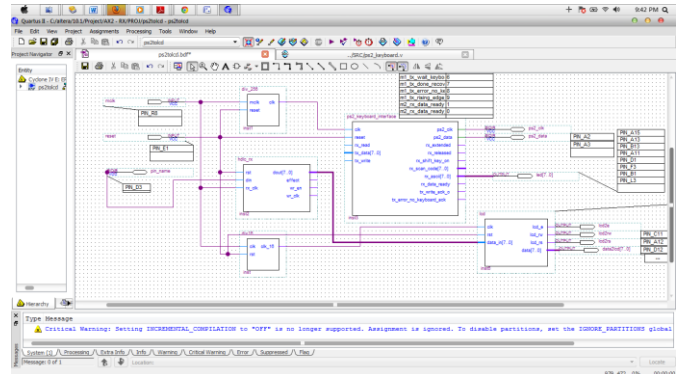
3. *Unpack* data

Data hasil *packing* yang dikirimkan akan diterima pada bagian *unpack* data untuk membongkar paket data. Bagian *unpack* data akan mengidentifikasi setiap *field* yang masuk dari *packing* data, apabila semua *field* yang masuk sudah teridentifikasi, maka akan diambil *field* informasinya untuk ditampilkan pada LCD karakter, namun apabila dari kalkulasi FCS ternyata ada *field* yang tidak lengkap, maka akan mengirimkan permintaan untuk mengirimkan data kembali.

Program *packing* dan *unpack* yang diimplementasikan pada FPGA Altera di *Software Quartus* dibagi menjadi dua bagian, yaitu blok *transmitter* dan blok *receiver*. Secara garis besar program *Transmitter* AX.25 dibagi menjadi lima code blok program, yang pertama adalah blok *clock keyboard*, *clock lcd*, *keyboard interface*, *lcd interface* dan *transmitter* AX.25. Blok *clock keyboard* digunakan untuk mensuplai *clock* yang dibutuhkan keyboard dalam pengiriman data, sedangkan blok *clock lcd* digunakan untuk mensuplai *clock* lcd dalam menerima data dan sinkronisasinya dengan FPGA. *Keyboard interface* berfungsi sebagai jembatan untuk menghubungkan data yang dikirim dari keyboard ke FPGA, dan juga sebagai penerjemah dari bilangan biner ke ASCII sehingga bisa dibaca ke LCD. *LCD interface* berfungsi untuk menampilkan data hasil masukan data yang akan *dipacking* ke dalam *frame* AX.25. Sedangkan blok kode *transmitter* AX.25 itu sendiri merupakan program untuk *packing* data ke dalam *frame* AX.25. Untuk *Receiver* AX.25 mempunyai blok kode yang hampir sama dengan *Transmitter* AX.25, hanya berbeda di blok *Receiver* AX.25, karena di dalam blok ini terdapat program untuk *unpack* *frame* AX.25 ke data yang sama dengan masukan. Kemudian data itu diterjemahkan di dalam blok *keyboard interface* untuk mengubahnya menjadi bentuk ASCII agar bisa ditampilkan di LCD. Di blok kode *Receiver* AX.25 juga terdapat blok *clock keyboard* dan *clock lcd* dengan fungsi yang sama seperti blok kode *Transmitter* AX.25.



Gambar 3. Code Block AX.25 TX (Transmitter)



Gambar 4. Code Block AX.25 RX (Receiver)

IV. PENGUJIAN DAN PEMBAHASAN

A. Pengujian Algoritma

Pengujian Algoritma diperlukan untuk mengetahui apakah algoritma dari protokol AX.25 sudah benar, mulai dari format data dan jumlah karakter. Pengujian algoritma dilakukan dengan cara menghubungkan FPGA dengan program *interface* GUI (*Graphical User Interface*) *ground station* inusat 01 yang dibuat menggunakan *software* Delphi 7. Pada Gambar dibawah ditampilkan hasil *packing* data dalam bentuk ASCII, dimana data yang dikirimkan berjumlah 17 karakter. Pada protokol ini akan dikirimkan sebanyak 500 karakter, atau menyediakan ruang kosong untuk 500 karakter yang akan dikirim dengan 3 *frame*, karena setiap *frame* hanya mampu menampung *field* info sebanyak 225 byte. Sedangkan karakter yang dikirim pada percobaan ini berjumlah 17, maka 483 karakter yang lain akan diisi oleh karakter kosong berupa spasi (hex : 20). Untuk mengetahui apakah format yang dikirimkan benar, maka perlu diubah ke dalam bilangan hexadecimal seperti yang ditampilkan pada gambar. Dan dari hasil analisa dengan pencocokan format data dan karakter yang dikirimkan dengan bilangan heksadesimalnya, didapatkan kecocokan hasil format data dan info yang dikirimkan dalam *packing* AX.25 dengan bilangan heksadesimalnya. Hal ini membuktikan bahwa algoritma yang dibuat sudah benar.



Gambar 5. Hasil *packing* data dalam ASCII

B. Pengujian Kecepatan Proses *Packing* dan *Unpack* Data

Pada pengujian ini untuk mengetahui waktu yang dibutuhkan untuk mengubah data yang dimasukkan melalui keyboard menjadi sebuah *frame* yang lengkap. Metode

pengukuran yang digunakan adalah dengan menggunakan *timer* FPGA untuk menghitung waktu pemrosesan. *Timer* akan dimulai pada saat data mulai di proses *packing* dan berhenti pada saat data telah *dipacking*. Hasil dari *timer* akan ditampilkan pada LCD dalam bentuk detik. Selain itu hasil pembacaan *timer* ini sendiri dipengaruhi oleh nilai *clock* FPGA yang bernilai 50 MHz. Pengujian dilakukan dengan mengisi *frame* secara penuh yaitu sepanjang 256 byte pada setiap *frame*.

Kecepatan proses ditentukan oleh *bit rate* dan *baud rate*, *bit rate* adalah banyaknya bit per detik, sedangkan *baud rate* yaitu banyaknya unit atau satuan sinyal per detik. Pada kasus ini setiap unit sinyal membawa 8 bit data. Untuk mengetahui kecepatan proses pengiriman data dapat dihitung dengan rumus berikut :

$$t = \frac{\text{Jumlah Bit Data (bit)}}{\text{Bit Rate (bit/s)}}$$

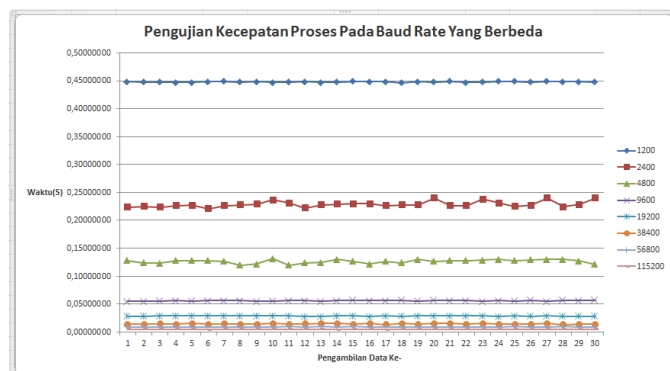
$$\text{Bit Rate} = \text{Baud Rate} \times \text{Unit Sinyal}$$

Jumlah bit data secara keseluruhan yang dikirimkan pada protokol ini adalah 4168 bit, dengan rincian sebagai berikut :

- Flag : 2 byte
- Address : 14 byte
- Control : 2 byte
- FCS : 2 byte
- PID : 1 byte
- Info : 500 byte
- Total : 521 byte = 521 x 8 bit = 4168 bit

Apabila menggunakan *baud rate* 1200 dengan 8 bit setiap unit sinyalnya, maka *bit rate*-nya sebesar 1200 x 8 = 9600 bit/s. Waktu pengiriman datanya adalah 4168/9600 = 0,43416666s.

Berikut ini adalah grafik yang menunjukkan waktu kecepatan proses pada *baud rate* yang berbeda dengan pengiriman data sebanyak tiga puluh kali. Dari grafik dapat diketahui bahwa semakin tinggi nilai *baud rate*, maka akan semakin cepat pula proses pengiriman datanya.



Gambar 6. Perbandingan kecepatan pengiriman data pada *Baud rate* yang berbeda

C. Pengujian Dengan *Baud rate* yang berbeda

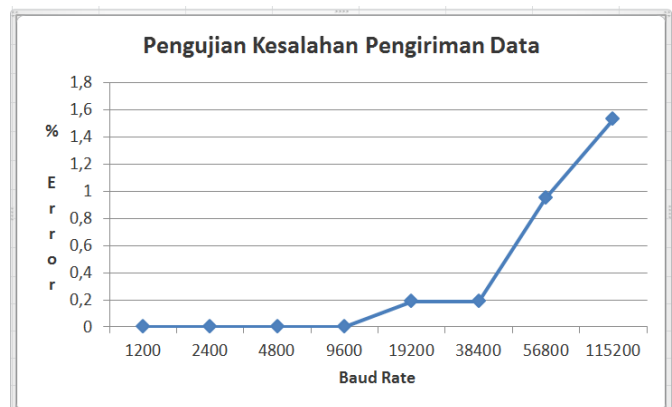
Pengujian ini bertujuan untuk mengetahui kemampuan FPGA untuk mengirimkan data dengan modem yang memiliki *baud rate* terima yang berbeda-beda serta mengetahui prosentase kesalahan dalam pengiriman data. Pengujian ini

dilakukan dengan mengirimkan *frame* AX.25 yang berisi informasi tetap namun dengan *baud rate* diubah ubah.

Untuk mengetahui hasil pengiriman dilakukan dengan menggunakan GUI *ground station* iinusat. Pengiriman data dilakukan sebanyak tiga puluh kali, kemudian dari hasil GUI yang telah ditangkap akan dibandingkan dengan parameter-parameter yang telah ditetapkan. Dari pengujian akan dihitung nilai *data rate error*. *Data rate error* adalah nilai kesalahan pada data yang diterima. Perhitungan *data rate error* dapat dilakukan dengan menggunakan rumus :

$$\text{data rate error} = \frac{\text{jumlah error pada data}}{\text{jumlah data yang dikirimkan}}$$

Dari hasil pengujian dapat diketahui bahwa semakin tinggi nilai *baud rate*, maka akan semakin tinggi pula nilai prosentase dalam pengiriman datanya. Setiap kanal komunikasi memiliki *bandwidth* yang terbatas, semakin tinggi *data rate*, durasi pulsa digital yang digunakan akan semakin pendek, dan apabila durasi pulsa semakin pendek maka *bandwidth* yang digunakan akan semakin lebar. Namun apabila jumlah bit per simbol dinaikkan agar *bit rate* transmisi menjadi besar, maka deteksi simbol akan semakin sulit dilakukan dan pengaruh noise akan semakin signifikan dan bisa menyebabkan perubahan level simbol yang mengakibatkan kesalahan pada data yang dikirimkan. *Noise* menurunkan kualitas komunikasi analog dan memunculkan eror pada komunikasi digital.

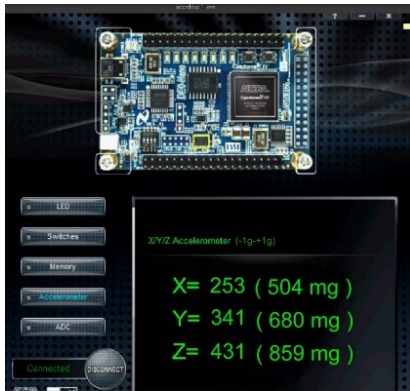


Gambar 7. Hasil pengujian kesalahan pengiriman data

Grafik diatas menunjukkan bahwa *baud rate* yang cocok digunakan untuk pengiriman data adalah antara 1200 – 9600, karena data dikirimkan dengan sempurna tanpa kesalahan.

D. Pengujian Simulator *Attitude monitoring*

Dalam implementasi *attitude monitoring* ini menggunakan *accelerometer* sebagai sensor untuk mengetahui perilaku dari satelit. *Accelerometer* mengukur keluaran dari 3 sumbu x, y, dan z. Range pengukuran dari *accelerometer* ini ±1g. Hasil pengukuran akan ditampilkan di GUI (*Graphic User Interface*) agar mudah dalam pembacaannya. Nilai dari pengukuran percepatan gravitasi dihitung dari nilai resolusi 10 bit dari register ADC *accelerometer*.



Gambar 8. GUI Attitude monitoring

DI. Pengujian Accelerometer

Pengujian percobaan ini dilakukan dengan mengambil data perputaran *accelerometer* pada sudut *pitch,yaw* dan *roll* setiap 10 derajat sebanyak 20 kali setiap detik dalam rentang 0⁰-180⁰, karena pada 0⁰ *accelerometer* akan bernilai (+1g) sedangkan pada 180⁰ akan bernilai (-1g) .

Tabel 4. Data Pengujian Accelerometer Pada Sumbu X

X-AXIS				
SUDUT	PERHITUNGAN	EKSPERIMEN	SELISIH	ERROR
10	192,1	201,5	9,4	0,048933
20	210,29	218,3	8,01	0,03809
30	228,48	237,45	8,97	0,039259
40	246,67	253,7	7,03	0,0285
50	264,85	272,1	7,25	0,027374
60	283,04	292,7	9,66	0,034129
70	301,23	311,7	10,47	0,034757
80	319,42	328,7	9,28	0,029053
90	337,61	348,1	10,49	0,031071
100	355,8	364,3	8,5	0,02389
110	373,99	381,75	7,76	0,020749
120	392,18	404,7	12,52	0,031924
130	410,37	417,65	7,28	0,01774
140	428,56	438,1	9,54	0,022261
150	446,74	455,7	8,96	0,020056
160	464,93	473,7	8,77	0,018863
170	483,12	491,75	8,63	0,017863
180	501,31	510,95	9,64	0,01923
MEAN			9,008889	0,027986

Dari perbandingan nilai diatas didapatkan selisih nilai ADC yang akan digunakan untuk menentukan koreksi pengukuran. Untuk rata-rata selisih pengukuran pada sumbu x (*pitch*) didapatkan nilai 9,01, pada sumbu y (*yaw*) didapatkan nilai 4,52, sedangkan sumbu z (*roll*) didapatkan nilai 21,68. Sehingga didapatkan nilai koreksi total pengukuran ADC *accelerometer* sebesar 11,74. Nilai ADC 11,74 apabila dikonversi kedalam (g) adalah ±0,072g. Untuk mengetahui prosentase error dari pengukuran, maka diperlukan data selisih hasil perhitungan dengan eksperimen untuk setiap sumbu. Dari data tersebut dapat diperoleh prosentase error dari setiap sumbu, prosentase error untuk sumbu x adalah 2,80%, error sumbu y adalah 1,35%, dan error sumbu z adalah 6,42%. Sedangkan error total untuk *accelerometer* secara keseluruhan diperoleh dari rata-rata error setiap sumbu, yaitu 3,52%.

DI. Pengujian Pengiriman Data Accelerometer Melalui PAD AX.25

Pengujian pengiriman data *accelerometer* melalui PAD AX.25 menghasilkan nilai yang sama pada *Packet Assembler* dan *Packet Disassembler* di *baud rate* 1200-9600, bisa

dibuktikan dengan melihat data yang dikirim dengan data yang diterima, selain itu bisa dicocokkan dengan nilai heksadesimalnya. Format data dari *accelerometer* adalah “xxxx yyyy zzzz” dimana x,y dan z menunjukkan data masukan ADC dari *accelerometer* dengan nilai maksimal 1023 dari setiap sumbu. Pada dasarnya data dari ADC bisa terbaca sebagai data karakter, sehingga PAD AX.25 mampu mengirimkan datanya, hanya saja batasan untuk karakternya tetap 500. Oleh karena itu PAD AX.25 ini mampu mengirimkan data telemetri berupa posisi dari satelit.

V. KESIMPULAN

Dari hasil implementasi dan analisa pada pengujian *packing* data yang telah dilakukan, beberapa hal yang dapat disimpulkan antara lain :

1. Modul PAD dapat diimplementasikan kedalam FPGA dengan kemampuannya mengirim data 500 karakter. Kinerja dari modul sangat baik pada pengaturan *baud rate* 1200 – 9600, karena dari hasil pengujian 30 kali pengiriman, tidak ada error yang terjadi, error baru terjadi pada saat nilai *baud rate* diatas 19.200.
2. Pada pengujian kecepatan pengiriman proses, waktu proses pengiriman *frame* AX.25 dengan panjang 256 byte membutuhkan waktu pada *baud rate* 1200 : 0,449 s, 2400 : 0,224s, 4800 : 0,114s, 9600 : 0,055s, 19200 : 0,028s, 38400 : 0,015s, 56800 : 0,009s, 115200 : 0,005s.
3. Pada pengujian pengiriman data *attitude monitoring* didapatkan koreksi pengukuran sebesar ±0,072 g, kesalahan tersebut bukan karena kesalahan pengiriman data, tetapi dari error koreksi sensor *attitude monitoring* yang digunakan.

DAFTAR PUSTAKA

- [1] Dicky Rismawan Raharjo. *Perancangan dan Pembuatan Modul PAD (Packet Assembler Disassembler) dengan Protokol AX.25 untuk Stasiun Bumi Portabel pada Sistem Komunikasi Satelit*. Tugas Akhir Jurusan Teknik Elektro ITS. Surabaya. 2011
- [2] William A.Beech, Douglas E. Nielsen, Jack Taylor. *AX.25 Link Access Protocol for Amateur Packet Radio*.Tucson Amateur Packet Radio Corporation .1997.
- [3] Pong P. Chu. *FPGA Prototyping by VHDL Examples*.Wiley. 2008.
- [4] Altera Corporation. *DE0-Nano User Manual*. Terasic Altera FPGA. 2012.
- [5] X.25 Devices and Protocol Operation. 1999.
- [6] Altera Corporation. *Quartus II Handbook Version 12.0*. 212.
- [7] Pedroni, Volnei A. *Circuit Design With VHDL*. MIT Press. 2004
- [8] Gerard Maral, Michel Bousquet, *Satellite Communications Systems*, Wiley, 2010