

PERANCANGAN SIMULASI KOMUNIKASI DATA MULTI FRAME ANTAR DUA KOMPUTER DENGAN PROTOKOL SDLC

(SEBAGAI MODUL PERCOBAAN SEMU LAB. TELKOM FT. UHN)

Oleh

Ir. Sindak Hutauruk, MSEE.

Dosen tetap Fakultas Teknik



**LEMBAGA PENELITIAN
UNIVERSITAS HKBP NOMMENSEN
MEDAN
2011**

KATA PENGANTAR

Pada komunikasi data peranan protokol memegang peranan yang sangat penting karena protokol dalam suatu komunikasi data sangat berperan dalam suatu jaringan komunikasi data point to multipoint atau multipoint to multipoint. Protokol berfungsi untuk mengatur lalu lintas pengiriman maupun penerimaan data agar dapat berlangsung dengan baik. Secara garis besar, ada dua jenis protokol, yaitu Protokol Asinkron (Asynchronous Protocol) dan Protokol Sinkron (Synchronous Protocol). Protokol Asinkron berorientasi karakter (character oriented) dan protokol sinkron ada yang berorientasi karakter maupun berorientasi bit. Protokol sinkron mempunyai kelebihan dibandingkan dengan protokol asinkron, sebab stasiun sekunder (secondary station) pada umumnya terbatas hanya satu pasang terminal/printer sehingga disebut juga dengan terminal stand alone, sedang pada protokol asinkron, stasiun sekunder dapat lebih dari satu pasang terminal/printer, terminal-terminal tersebut membentuk satu cluster dimana satu cluster dapat terdiri dari 50 devais (terminal dan printer). Oleh sebab itu protokol sinkron sangat banyak diimplementasikan pada jaringan komunikasi data. Salah satu protokol sinkron yang berorientasi karakter adalah protokol BSC (Binary Synchronous Communications) atau disebut juga dengan protokol Bisync, salah satu protokol yang berorientasi bit (Bit Oriented Protocol, BOP) adalah Synchronous Data Link Control (SDLC) yang dikembangkan oleh IBM. Transmisi pada SDLC dilakukan secara serial dengan transfer data dilakukan berbasis bit per bit. Transmisi data pada SDLC dapat dilakukan secara simplex, half duplex, maupun dengan full duplex. Semua transmisi berbentuk frame, dan format frame tunggal memadai untuk seluruh jenis pertukaran data dan kontrol, oleh sebab itu pembentukan format frame pada SDLC sangat penting karena semua address, kontrol, informasi data, karakter pengecekan kesalahan berada dalam satu frame. Frame yang dikirimkan dapat terdiri dari beberapa frame (multi frame), proses pembentukan multi frame pada protokol SDLC ini disimulasikan dengan menggunakan perangkat lunak visual basic sehingga setiap langkah dalam pembentukan frame tersebut dapat diperagakan dan juga dapat dilakukan komunikasi langsung antar dua buah komputer yang dengan demikian dapat digunakan sebagai modul percobaan semu pada laboratorium dan juga sebagai alat bantu pengajaran pada salah satu topik mata kuliah komunikasi data.

Penelitian ini dilakukan selama 6 (enam) bulan di Jurusan Teknik Elektro Fakultas Teknik UHN dan dibantu oleh seorang mahasiswa semester akhir.

Pada kesempatan ini peneliti mengucapkan terimakasih kepada Lembaga Penelitian UHN yang telah mempercayai peneliti untuk melakukan penelitian tersebut, dan terimakasih juga kepada semua pihak yang telah memberikan masukan-masukan sehingga penelitian ini dapat diselesaikan dengan baik. Semoga hasil penelitian ini dapat berguna bagi institusi UHN dan bagi pihak-pihak yang membutuhkannya.

Medan, Pebruari 2011
P e n e l i t i,

Ir. Sindak Hutauruk, MSEE.

DAFTAR ISI

	Halaman
KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	iii
ABSTRAK	iv
BAB I. PENDAHULUAN	1
I.1. Latar Belakang	1
I.2. Perumusan Masalah	2
I.3. Tujuan Penelitian	2
I.4. Kontribusi Penelitian	2
BAB II. TINJAUAN PUSTAKA	3
II.1. Deskripsi SDLC	3
II.1.2. Field Informasi (<i>Information Field</i>)	4
II.1.3. Field Flag (<i>Flag Field</i>)	4
II.1.4. Field Alamat (<i>Address Field</i>)	5
II.1.5. Field Kontrol (<i>Control Field</i>)	6
II.1.5.1. Frame Informasi	6
II.1.5.2. Frame Supervisory	8
II.1.5.3. Frame Unnumbered	9
II.2. Komunikasi Data Serial	10
II.2.1. Pengiriman Data Serial Tak Sinkron (Asinkron)	11
II.2.2. Flow Control	12
II.2.3. Karakteristik Sinyal Port Serial	13
II.2.4. Konfigurasi Port Serial	14
BAB III. METODOLOGI PENELITIAN	15
III.1. Menentukan Flag	16
III.2. Menentukan Alamat	16
III.3. Menentukan Field Kontrol	16
III.4. Menentukan Field Informasi	17
III.5. Menentukan Field FCS	17
BAB IV. HASIL RANCANGAN	18
IV.1. Rancangan Perangkat Lunak (Simulasi)	18
IV.2. Rancangan Perangkat Keras (Hardware)	23
BAB V. KESIMPULAN DAN SARAN	24
V.1. Kesimpulan	24
V.2. Saran	24
DAFTAR PUSTAKA	25
LAMPIRAN	26
<i>Penelitian Sindak Hutauruk , September 2010 – Januari 2011</i>	ii

DAFTAR GAMBAR

Nomor Gambar	Judul Gambar	Halaman
1.	Format Frame SDLC	3
2.	Pengiriman Data Serial	10
3.	Karakter 'A' Tanpa Bit Paritas	11
4.	Sinkronisasi Awal Akhir	12
5.	Karakter 'A' Pada Level Tegangan RS 232	14
6.	Konektor Port Serial DB-9 Pada Bagian Belakang CPU	14
7.	Flow Chart Pembentukan Format SDLC	15
8.	Tampilan Simulasi Pembentukan Format Frame SDLC	18
9.	Tampilan Simulasi Eksekusi Bentuk Frame SDLC	20
10.	Tampilan Simulasi Proses Pengiriman Frame SDLC	21
11.	Tampilan Simulasi untuk Pilihan Error yang Terjadi	21
12.	Tampilan Frame yang Diterima	22
13.	Tampilan Ada Tidaknya Kesalahan yang Diterima	22
14.	Tampilan Informasi Frame yang Diterima	23
15.	Hubungan Antara 2 Buah Port Komputer	23

ABSTRAK

Peranan dalam sistem komunikasi data, Protokol berfungsi untuk mengatur lalu lintas pengiriman maupun penerimaan data agar dapat berlangsung dengan baik. Perancangan simulasi komunikasi pengiriman multiframe pada protokol SDLC terdiri dari beberapa field yang masing-masing field terdiri dari flag, address, control, informasi, FCS (Frame Check Sequence) dan ditutup dengan flag kembali. Ada tiga format pada frame SDLC yang dapat dibentuk sebagai informasi, supervisory, atau unnumbered. Multiframe dari informasi yang dikirimkan disimulasikan dengan mengirimkannya antar dua buah komputer sehingga dapat dengan jelas dilihat proses pengiriman multiframe SDLC tersebut. Simulasi ini memberikan kebebasan menentukan isi setiap field pada multiframe yang dibentuk, sehingga sangat baik untuk digunakan pada percobaan pada laboratorium semu komunikasi data.

I. PENDAHULUAN

1. Latar Belakang

Protokol dalam suatu komunikasi data sangat berperan dalam suatu jaringan komunikasi data point to multipoint atau multipoint to multipoint. Protokol berfungsi untuk mengatur lalu lintas pengiriman maupun penerimaan data agar dapat berlangsung dengan baik.

Secara garis besar, ada dua jenis protokol, yaitu Protokol Asinkron (*Asynchronous Protocol*) dan Protokol Sinkron (*Synchronous Protocol*). Protokol Asinkron berorientasi karakter (*character oriented*) dan protokol sinkron ada yang berorientasi karakter maupun berorientasi bit. Protokol sinkron mempunyai kelebihan dibandingkan dengan protokol asinkron, sebab stasiun sekunder (*secondary station*) pada umumnya terbatas hanya satu pasang terminal/printer sehingga disebut juga dengan terminal stand alone, sedang pada protokol asinkron, stasiun sekunder dapat lebih dari satu pasang terminal/printer, terminal-terminal tersebut membentuk satu cluster dimana satu cluster dapat terdiri dari 50 devais (terminal dan printer). Oleh sebab itu protokol sinkron sangat banyak diimplementasikan pada jaringan komunikasi data. Salah satu protokol sinkron yang berorientasi karakter adalah protokol BSC (*Binary Synchronous Communications*) atau disebut juga dengan protokol Bisync, salah satu protokol yang berorientasi bit (*Bit Oriented Protocol, BOP*) adalah *Synchronous Data Link Control (SDLC)* yang dikembangkan oleh IBM.

Transmisi pada SDLC dilakukan secara serial dengan transfer data dilakukan berbasis bit per bit. Transmisi data pada SDLC dapat dilakukan secara *simplex*, *half duplex*, maupun dengan *full duplex*. Transmisi dapat berbentuk single frame maupun multi frame, dan pertukaran frame dapat dilakukan secara langsung antar dua buah komputer, oleh sebab itu pembentukan format multi frame pada SDLC sangat penting karena semua address, kontrol, informasi data, karakter pengecekan kesalahan berada dalam satu frame.

2. Perumusan Masalah

Pada SDLC, data yang dikirim berada dalam satu frame dengan control, address, dan FCS (Frame Check Sequence) sehingga pembentukan atau penyusunan bit-bit pada setiap field yang membentuk frame sangatlah penting. Pengiriman frame dengan jumlah yang banyak atau disebut dengan multiframe ke sebuah stasiun dengan simulasi komunikasi data yang langsung antar dua buah komputer dilakukan dengan tujuan agar dapat dilihat secara jelas proses pembentukan frame, pengiriman multiframe, dan proses komunikasi data antar dua buah komputer.

3. Tujuan Penelitian

Penelitian ini merupakan kelanjutan dari penelitian yang dilakukan peneliti dengan judul "Perancangan Simulasi Pembentukan Frame Pada Protokol SDLC Dengan Perangkat Lunak Visual Basic". Tujuan dari penelitian ini adalah untuk :

1. Mensimulasikan pembentukan format frame pada SDLC dengan menggunakan perangkat lunak, sehingga dapat terlihat dengan jelas isi setiap field pada frame yang dikirim oleh stasiun primer maupun oleh stasiun sekunder.
2. Mensimulasikan komunikasi data dalam multiframe SDLC antar 2 buah komputer
3. Memperagakan secara langsung proses komunikasi yang terjadi dengan multi frame yang dibentuk pada protokol SDLC.
4. Diharapkan hasil rancangan ini dapat digunakan sebagai salah satu modul percobaan semu pada Laboratorium Telekomunikasi Fakultas Teknik UHN

4. Kontribusi Penelitian

Penelitian ini merupakan perancangan simulasi komunikasi data multi frame antar dua komputer dengan protokol SDLC. Hasil rancangan ini nantinya diharapkan akan memberikan kontribusi diantaranya :

- a. Dapat sebagai modul percobaan semu pada laboratorium Telekomunikasi
- b. Penelitian berkelanjutan untuk menghasilkan yang lebih baik
- c. Sebagai alat bantu bahan ajar untuk mata kuliah komunikasi data
- d. Memberikan motivasi kepada mahasiswa untuk mencoba melakukan perancangan dalam bidang komunikasi data.
- e. Sebagai kewajiban dosen melakukan tridarma perguruan tinggi.
- f. Sebagai salah satu usaha untuk menaikkan akreditasi program studi

II. TINJAUAN PUSTAKA

II.1. Deskripsi SDLC

Pada SDLC, kode karakter yang digunakan adalah EBCDIC (*Extended Binary Coded Decimal Interchange Code*) yang ditransfer secara sinkron dan dalam satu group yang disebut frame. Secara umum, panjang karakter dalam satu frame sebanyak 256 karakter. Pada SDLC, dikenal dua tipe stasiun, yaitu stasiun primer (*primary*) dan stasiun sekunder (*secondary*). Stasiun primer mengontrol pertukaran data dalam kanal komunikasi dan memberikan perintah kepada stasiun sekunder, sedangkan stasiun sekunder menerima perintah dan meresponnya ke stasiun primer.

Ada tiga keadaan transmisi pada SDLC yaitu : transien, idle, dan aktif. Keadaan transien adalah keadaan sebelum dan sesudah inisialisasi transmisi, keadaan idle adalah keadaan setelah 15 atau lebih bit '1' diterima, dan keadaan aktif adalah keadaan dimana stasiun primer maupun stasiun sekunder sedang mentransmisi informasi atau sinyal kontrol. Gambar 1. adalah gambar format frame SDLC, dimana frame yang dikirim oleh stasiun primer maupun stasiun sekunder mempunyai format frame yang sama. Ada 5 field dalam satu frame SDLC, yaitu *flag field*, *address field*, *control field*, *text* atau *information field*, dan *frame check sequence field* ¹⁾

F Flag	A Address	C Control	I Information	FCS Frame Check Sequence	F Flag
01111110					01111110
8 bit	8 bit	8 bit	Variabel Kelipatan 8 bit	16 bit CRC - 16	8 bit

Gambar 1. Format Frame SDLC

1) Wayne Tomasi, 1994, *Advanced Electronic Communications Systems, Third Edition, Prentice Hall International, Inc., USA*

II.1.2. Field Informasi (*Information Field*)

Semua informasi yang ditransmisikan berada dalam Information field (I field), dan jumlah bit informasi dalam information field harus kelipatan 8 bit dengan menggunakan kode EBCDIC, Kode tersebut dipakai oleh IBM dan mesin-mesin yang kompatibel dengan IBM. Karakter yang dibentuk oleh kode EBCDIC terdiri dari 8 bit, sehingga jumlah karakter yang dapat dibentuk oleh kode EBCDIC sebanyak $2^8 = 256$ karakter. Pada kode EBCDIC yang menjadi bit LSB (*Low Significant Bit*) adalah b_7 dan yang menjadi bit MSB (*Most Significant Bit*) adalah b_0 , yang ditransmisikan lebih dahulu adalah bit b_7 dan bit yang ditransmisikan terakhir adalah bit b_0 . Kode EBCDIC tidak menggunakan bit paritas, semua bit ($b_0 - b_7$) membentuk sebuah karakter.

Contoh :

U								H								N							
b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
0	0	1	0	0	1	1	1	0	0	0	1	0	0	1	1	1	0	1	0	1	0	1	1
E4								C8								D5							

II.1.3. Field Flag (*Flag Field*)

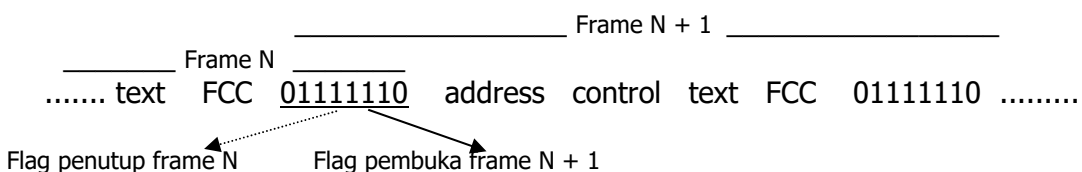
Ada dua flag field pada setiap frame, yaitu flag pembuka dan flag penutup. Flag pembuka berada diawal setiap frame dan flag penutup diakhir setiap frame. Jumlah bit pada flag pembuka maupun flag penutup berjumlah 8 bit. Flag juga sebagai pembatas antar frame, dan flag dapat dijadikan sebagai sinkronisasi karakter. Pola bit pada flag adalah 01111110 atau 7EH, yang pada kode EBCDIC merupakan karakter "=" . Ada beberapa variasi penggunaan flag, yaitu :

1. Satu flag pada pembuka frame dan satu flag pada diakhir setiap frame.

Flag pembuka
Flag penutup

..... 01111110 address control text FCC 01111110

2. Flag penutup dari satu frame dapat merupakan flag pembuka untuk frame berikutnya.



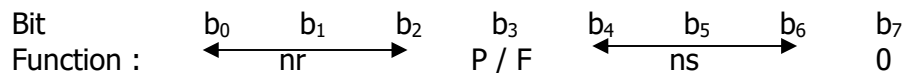
II.1.5. Field Kontrol (*Control Field*)

Jumlah bit pada *control field* sebanyak 8 bit yang merupakan identitas dari tipe frame tersebut. Control field digunakan untuk *polling*, konfirmasi terhadap frame informasi yang telah diterima, dan juga berfungsi sebagai manajemen link data. Ada tiga format frame pada SDLC, yaitu :

1. Informasi
2. Supervisory
3. Unnumbered

II.1.5 1. Frame Informasi

Frame informasi digunakan sebagai tempat data yang berupa informasi di dalam field informasi. Frame informasi digunakan untuk mentransmisikan urutan-urutan informasi. Pola bit field control pada frame informasi sebagai berikut :



nr = number receive, ns = number sent

Bit b₇ sebagai least significant bit di isi bit '0' yang berarti bahwa ini adalah frame informasi (information frame), sedangkan bit ke 4,5, dan 6 digunakan untuk penomoran frame yang ditransmisikan (ns). Jumlah bit pada ns sebanyak 3 bit, yang berarti dari biner 000 sampai dengan 111 (0-7), sehingga ada 8 penomoran frame yang dapat dilakukan, yaitu frame 000 untuk frame ke 1, 001 untuk frame ke 2, 010 untuk frame ke 3 dan seterusnya sampai 111 untuk frame ke 8. Penomoran ns untuk frame berikutnya (setelah 8 frame) adalah kembali ke 000 dan seterusnya.

Bit b₀, b₁, dan b₂ adalah sebagai nr, yaitu jumlah frame yang telah diterima dengan baik, dan secara otomatis meminta ditransmisikan kembali terhadap frame yang tidak diterima dengan baik, selain itu nr juga menyatakan nomor frame berikutnya yang akan dikirimkan oleh stasiun pengirim. nr pada stasiun penerima menyatakan bahwa frame yang telah diterimanya dengan baik dari stasiun pengirim adalah frame nr - 1. Frame yang telah dikirim oleh stasiun pengirim yang ns nya lebih besar dari frame nr-1 stasiun penerima akan dikirimkan kembali bersama sama dengan frame berikutnya,

karena frame tersebut belum / tidak diterima oleh stasiun penerima dengan baik. Berikut contoh pengiriman frame informasi antara stasiun primer dan sekunder,

Stasiun Primer	ns :	0 1 2	3 4	4 5 6 7 0
Stasiun Primer	nr :	0 0 0	2 2	5 5 5 5 5
Stasiun Sekunder	ns :	0 1	2 3 4	5
Stasiun Sekunder	nr :	3 3	4 4 4	1

Misalkan stasiun primer dan sekunder mulai melakukan pertukaran informasi yang diawali dengan mereset ns dan nr masing-masing menjadi 000. Stasiun primer mengirimkan 3 nomor frame informasi yaitu ns = 0,1, dan 2, dan dalam waktu yang bersamaan stasiun primer juga mengirimkan nr = 0, karena frame yang akan dikirimkan oleh stasiun sekunder adalah ns = 0. Stasiun sekunder memberikan respon dengan mengirimkan 2 frame informasi yaitu frame 0 dan 1 (ns= 0,1) dan juga mengirimkan nr = 3 yang berarti semua frame informasi yang dikirim oleh stasiun primer diterima dengan baik oleh stasiun penerima dan nomor ini juga menyatakan stasiun primer akan mengirimkan frame informasi ke 3. Stasiun primer mengirimkan 2 frame informasi yaitu ns= 3, dan 4 yang secara bersamaan juga mengirimkan nr=2 yang artinya bahwa stasiun primer telah menerima dengan baik frame informasi 0, dan 1 dari stasiun sekunder. Stasiun sekunder merespon dengan mengirimkan 3 frame informasi dengan ns = 2,3, dan 4 juga secara bersamaan mengirimkan nr = 4 yang artinya frame informasi yang diterimanya dengan baik adalah frame nr-1 = 4-1 = 3, dengan demikian frame informasi ke 4 belum atau tidak diterima dengan baik oleh stasiun sekunder. Stasiun primer mengirimkan ulang frame informasi yang tidak diterima dengan baik oleh stasiun sekunder yaitu frame 4 ditambah frame yang baru yaitu 5,6,7, dan 0 bersama sama dengan nr = 5 yang artinya bahwa frame ke 4 telah diterima oleh stasiun primer dengan baik dan akan menunggu frame ke 5 dari stasiun sekunder. Stasiun sekunder merespon dengan mengirimkan 1 frame informasi yaitu frame dengan ns= 5 dan secara bersamaan mengirimkan nr= 1 yang artinya stasiun sekunder telah menerima frame informasi 4,5,6,7, dan 0 dengan baik dan akan menunggu frame informasi berikutnya yaitu frame informasi 1. Pada kondisi ini semua frame yang telah ditransmisikan telah dikonfirmasi kecuali frame 5 dari stasiun sekunder.

Dengan SDLC , stasiun tidak dapat mengirimkan lebih dari 7 frame tanpa ada konfirmasi, sebagai contoh : bila stasiun primer mengirimkan 8 frame (ns =

0,1,2,3,4,5,6, dan 7) dan stasiun sekunder akan merespon dengan $nr = 0$, dan ini membingungkan karena menjadi tanda tanya frame mana yang telah dikonfirmasi, apakah ke delapan frame telah diterima dengan baik atau apakah frame 0 mengalami error (kesalahan) dan kedelapan frame harus ditransmisi kembali? (pada SDLC, semua frame yang telah ditransmisikan mulai dari frame $nr-1$ harus ditransmisikan kembali). Bit ke 3 pada frame informasi ini menyatakan Poll (P) atau Not a Poll (P) untuk frame yang dikirim oleh stasiun primer dan menyatakan Final (F) atau Not a Final (F) untuk frame yang dikirim oleh stasiun sekunder. Bila stasiun primer akan mengirimkan frame ke stasiun sekunder maka stasiun primer akan melakukan Polling ke stasiun sekunder dengan membuat bit P menjadi bit '1' sedangkan bila tidak melakukan Polling maka bit P di set menjadi bit '0'. Selama bit P diset menjadi bit '1' atau dengan kata lain pada saat stasiun primer melakukan Polling, maka stasiun sekunder tidak dapat mengirimkan frame ke stasiun primer. Bila frame yang dikirimkan oleh stasiun sekunder merupakan frame terakhir maka bit F di set menjadi bit '1' dan bila bukan merupakan frame terakhir maka F diset menjadi bit '0'.

Dengan frame informasi, stasiun primer dapat memilih stasiun sekunder yang akan dikirim frame, mengirimkan informasi yang telah diformat, mengkonfirmasi frame yang telah diterima, dan melakukan Poll dengan transmisi tunggal.

II.1.5. 2. Frame Supervisory

Frame Supervisory tidak dapat digunakan untuk mentransfer informasi, hanya digunakan sebagai bantuan dalam transfer informasi. Frame Supervisory digunakan untuk mengkonfirmasi frame yang telah diterima, menyatakan kondisi sibuk atau siap (ready), dan melaporkan frame yang mengalami error. Pola bit field control untuk Frame Supervisory adalah sebagai berikut :

Bit	:	b0	b1	b2	b3	b4	b5	b6	b7
Fungsi	:	-----	nr	-----	P or F	X	X	0	1

Bit b0, b1, dan b2 sebagai nr, bit b3 menyatakan Poll/Not a Poll atau Final/Not a Final, bit b6 dan b7 menyatakan frame supervisory, bit b4 dan b5 digunakan sebagai indikasi status penerimaan dari stasiun transmisi atau menyatakan permintaan transmisi atau transmisi ulang terhadap urutan frame informasi. Dengan kondisi dua bit tersebut (b4 dan b5) dapat menyatakan 4 fungsi, yaitu :

b4	b5	Status Penerima
0	0	Ready to Receive (RR)
0	1	Ready Not to Receive (RNR)
1	0	Reject (REJ)
1	1	Not Use with SDLC

Bila stasiun primer mengirimkan frame supervisory dengan P diset '1' dan status Ready to Receive, ini adalah ekivalen dengan Poll secara umum sama seperti pada protokol bisync. Frame supervisory digunakan oleh stasiun primer untuk melakukan polling dan untuk melakukan konfirmasi terhadap frame informasi yang telah diterima bila tidak ada informasi yang akan dikirimkan. Stasiun sekunder menggunakan format supervisory untuk mengkonfirmasi frame informasi yang telah diterima dan untuk melaporkan status penerimaan ke stasiun primer.

Bila stasiun sekunder mengirimkan frame supervisory dengan status RNR maka stasiun primer tidak dapat mengirimkan frame informasi sampai status RNR di clear. RNR akan clear bila stasiun sekunder mengirimkan frame informasi dengan bit F = '1' atau RR / REJ dengan F = '0'. Perintah atau respon REJ digunakan untuk mengkonfirmasi frame informasi sampai nr-1 dan juga digunakan untuk meminta agar dilakukan transmisi kembali frame informasi yang dimulai dengan nomor frame yang diidentifikasi oleh frame REJ. Perintah atau respon REJ hanya digunakan pada operasi transmisi Full-Duplex.

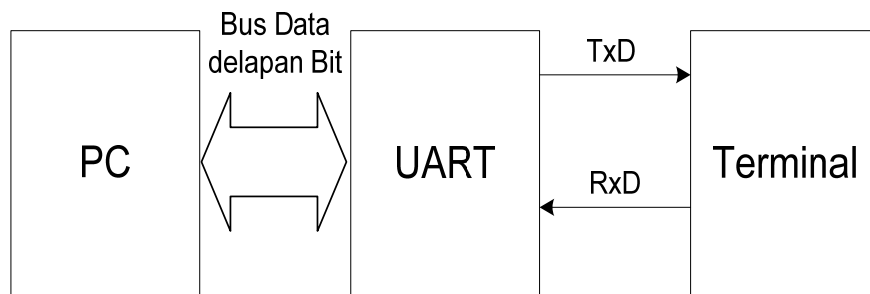
II.1.5. 3. Frame Unnumbered

Frame Unnumber tidak dilakuan pada penelitian ini karena terlalu banyak menggunakan perintah perintah.

II.2. Komunikasi Data Serial

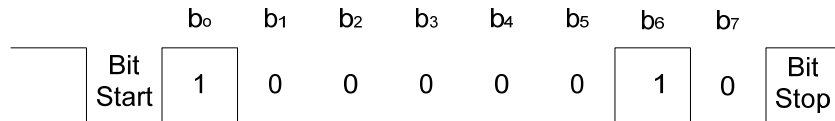
Pengiriman data serial biasanya digunakan untuk sambungan dengan jarak relatif lebih jauh. Gambar 2. menunjukkan konsep dasar pengiriman data serial. Data paralel internal dimasukan ke pengubah paralel ke seri (UART). Pengubah paralel ke seri biasanya dengan IC juga melakukan sejumlah fungsi yang lain dan dikenal sebagai UART, ACIA, PIA, dan lain-lain. Kanal serial mengirimkan setiap karkater per elemen

sehingga hanya diperlukan dua penghantar, yaitu kirim data (TxD), dan terima data (RxD). Masing-masing elemen isyarat ekuivalen dengan satu bit, dua atau tiga bit (disebut dibit atau tritbit), atau kurang dari satu bit (penyandian Manchester), tetapi dalam hal ini sebuah elemen dianggap sama dengan satu bit. Karena bit-bit dikirimkan secara berurutan dan tidak serempak, kecepatan pemindahan data lebih rendah dibanding pengiriman secara paralel. Pengiriman akan dimulai dari LSB (*least significant bit*), dan diakhiri dengan MSB (*most significant bit*). Setiap karakter yang dikirimkan, disajikan dengan suatu urutan bit tertentu sesuai dengan sandi yang digunakan. Penerima harus mencacah isyarat data yang sama, pada waktu yang tepat sebelum membentuk kembali karakter yang diterima.



Gambar 2. Pengiriman data Serial

Pada UART (*Universal Asynchronous Receiver / Transmitter*), kecepatan pengiriman data (baud rate) dan fase clocknya pada sisi transmitter dan pada sisi receiver harus sinkron. Untuk itu diperlukan sinkronisasi antara transmitter dan receiver. Hal ini dilakukan oleh bit 'Start' dan bit 'Stop'. Ketika saluran transmisi dalam keadaan idle, output UART adalah dalam keadaan logika '1'. Ketika transmitter ingin mengirimkan data, output UART akan diset lebih dahulu ke logika '0' untuk waktu satu bit. Sinyal ini pada receiver akan dikenali sebagai sinyal 'Start' yang digunakan untuk mensinkronkan fase clocknya sehingga sinkron dengan fase clock transmitter. Selanjutnya data akan dikirimkan secara serial dari bit paling rendah (bit 0) sampai bit tertinggi. Selanjutnya akan dikirim sinyal 'Stop' sebagai akhir dari pengiriman data serial. Cara pemberian kode data yang disalurkan tidak ditetapkan secara pasti. Berikut ini adalah contoh pengiriman huruf 'A' dalam format ASCII (41 heksa / 1000001 biner) tanpa bit paritas.



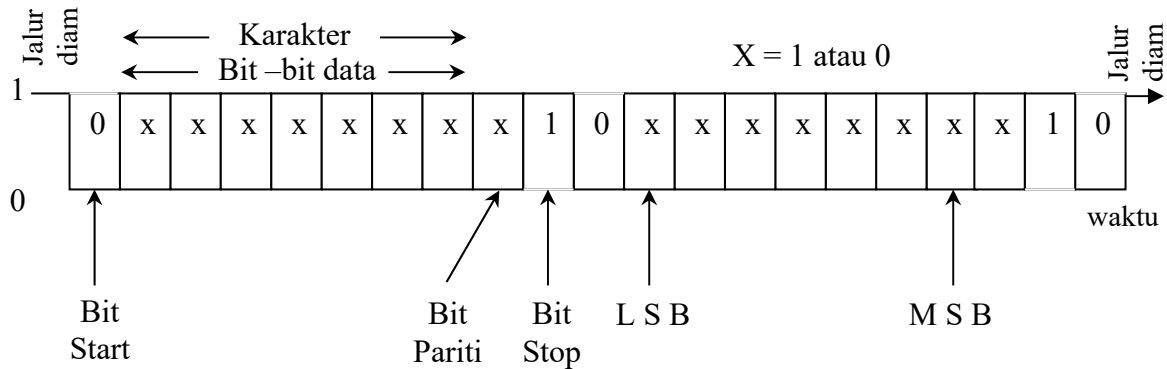
Gambar 3. Karakter 'A' tanpa bit paritas

Kecepatan transmisi (baud rate) dapat dipilih bebas dalam rentang tertentu. Baud rate yang umum dipakai adalah 110, 135, 150, 300, 600, 1200, 2400, dan 9600 (bit/detik). Dalam komunikasi data serial, baud rate dari kedua alat yang berhubungan harus diatur pada kecepatan yang sama. Selanjutnya, harus ditentukan panjang data (6,7 atau 8 bit), paritas (genap, ganjil atau tanpa paritas), dan jumlah bit 'Stop' (1,1½ atau 2 bit).

II.2.1. Pengiriman Data Serial Tak Sinkron (Asinkron)

Pada pengiriman data asinkron, setiap karakter dikirimkan sebagai satu kesatuan (*entity*) bebas, yang berarti bahwa waktu antara pengiriman bit terakhir dari sebuah karakter dan bit pertama dari karakter berikutnya tidak tetap. Pengiriman data asinkron lebih sederhana dibandingkan pengiriman sinkron, karena hanya isyarat data saja yang dikirimkan. Detak penerima dibangkitkan secara lokal di dalam penerima dan tetap dijaga agar sesuai dengan detak pengirim yang menggunakan bit awal (*Bit Start*) dan bit akhir (*Bit Stop*) yang dikirimkan dengan setiap karakter. Pada keadaan tidak berfungsi, pengirim akan mempertahankan tegangan jalur pada aras biner 1, dan detak penerima dihentikan. Pada saat pengirim mempunyai karakter untuk dikirimkan, pertama kali pengirim akan mengubah tegangan jalur menjadi aras biner 0, disebut bit awal, selama periode waktu satu bit setelah itu bit dari karakter tersebut dikirimkan. Detak penerima kemudian bekerja secara bebas untuk membangkitkan pulsa kira-kira 1 bit dan setelah itu setiap bit harus dicacah dengan interval waktu satu bit. Hal ini berarti detak penerima biasanya disesuaikan untuk meyakinkan bahwa waktu transisi detak terjadi kira-kira separuh dari waktu yang diperlukan untuk menerima sebuah bit. Dengan demikian, pencacahan setiap bit terjadi di tengah-tengahnya, dan inilah yang diinginkan agar kemungkinan terjadinya kesalahan dapat diperkecil. Pada akhir setiap karakter bit akhir dikirimkan, tegangan pada aras biner adalah 1 untuk menghentikan detak penerima. Detak penerima akan menunggu sampai bit awal berikutnya. Penyesuaian

antara detak pengirim dan penerima terjadi karakter per karakter. Hal ini berarti detak penerima tidak harus sangat stabil.



Gambar 4. Sinkronisasi awal akhir

Efisiensi tak sinkron tak begitu tinggi, karena hanya 7 dari 10 bit yang dikirimkan berisi informasi yang sesungguhnya. Istilah *asynchronous* dipakai untuk menunjuk ke suatu kanal yang mempunyai kemampuan untuk mengirimkan data tetapi tidak dapat melakukan isyarat pewaktuan (*timing signal*).

Jika detak penerima bekerja pada kecepatan yang berbeda dengan detak pengirim akan terjadi kemungkinan penerima tidak menerima setiap bit yang dikirim oleh pengirim. Jika detak penerima sedikit lebih cepat dari pengirim, pengirim akan mengambil sampel data yang datang lebih cepat. Setelah itu penerima akan mengambil sampel bit yang sama untuk kedua kalinya dan data yang diterima akan keluar dari sinkronisasi dengan data yang dikirim.

II.2.2. Flow Control

Jika kecepatan transfer data dari DTE ke DCE (misal komputer ke modem) lebih cepat dari pada transfer data dari DCE ke DCE (misal modem ke modem), cepat atau lambat kehilangan data akan terjadi karena buffer pada DCE akan mengalami overflow. Untuk itu diperlukan Flow Control untuk mengatasi masalah tersebut.

Dikenal dua macam Flow Control, yaitu secara software dan secara hardware. Flow Control secara software sering disebut Xon / Xoff. Flow Control menggunakan karakter Xon (tipikalnya karakter ASCII 17) dan karakter Xoff (tipikalnya ASCII 19) untuk

melakukan kontrol. DCE akan mengirimkan Xoff ke komputer untuk memberitahukan komputer agar menghentikan pengiriman data jika buffer pada DCE telah penuh. Jika buffer telah kembali siap menerima data, DCE akan mengirimkan karakter Xon ke komputer dan komputer akan mengirimkan data selanjutnya sampai data terkirim semua atau komputer menerima karakter Xoff lagi. Keuntungan Flow Control secara software ini adalah hanya diperlukan kabel sedikit karena karakter kontrol dikirimkan lewat saluran Tx / Rx. Akan tetapi kecepatan pengiriman data menjadi lambat.

Flow Control secara hardware sering disebut RTS / CTS. Flow Control menggunakan dua kabel untuk melakukan pengontrolan. Komputer akan menset saluran Request To Send jika akan mengirimkan data ke DCE. Jika buffer di DCE siap menerima data maka DCE akan membalas dengan menset saluran Clear To Send dan komputer akan mulai mengirimkan data. Jika buffer telah penuh, maka akan direset dan komputer akan menghentikan pengiriman data sampai saluran ini diset kembali.

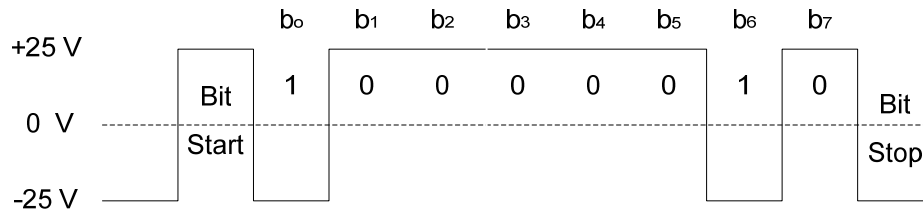
II.2.3. Karakteristik Sinyal Port Serial

Standar sinyal komunikasi serial yang banyak digunakan adalah RS 232 yang dikembangkan oleh Electronic Industry Association and Telecommunication Industry Association (EIA / TIA) yang pertama kali dipublikasikan pada tahun 1962. Ini terjadi jauh sebelum IC TTL populer sehingga sinyal ini tidak ada hubungan sama sekali dengan level tegangan IC TTL. Standar ini hanya menyangkut komunikasi data antara komputer (Data Terminal Equipment – DTE) dengan alat-alat pelengkap komputer (Data Circuit Terminating Equipment – DCE). Standar RS 232 inilah yang biasa digunakan pada port serial IBM PC kompatibel.

Standar sinyal serial RS 232 memiliki ketentuan level tegangan sebagai berikut :

1. Logika '1' disebut 'mark' terletak antara -3 Volt hingga -25 Volt.
2. Logika '0' disebut 'space' terletak antara +3 Volt hingga +25 Volt.
3. Daerah tegangan antara -3 Volt hingga +3 Volt adalah invalid level, yaitu daerah tegangan yang tidak memiliki level logika pasti sehingga harus dihindari. Demikian juga, level tegangan lebih negatif dari -25 Volt atau lebih positif dari +25 Volt juga harus dihindari karena tegangan tersebut dapat merusak line driver pada saluran RS 232.

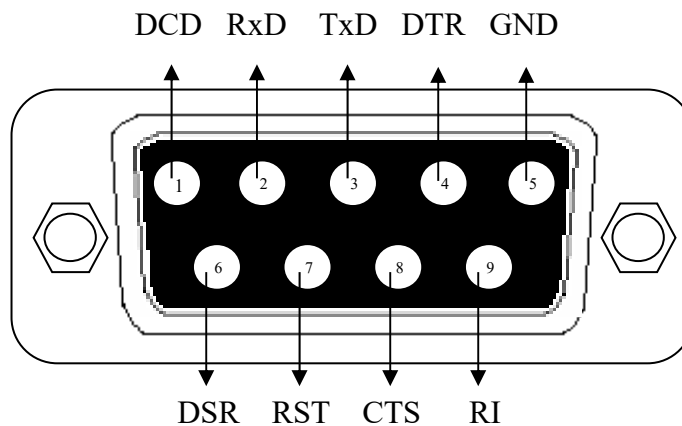
Gambar 5. berikut ini adalah contoh level tegangan RS 232 pada pengiriman karakter 'A' dalam format ASCII tanpa bit paritas.



Gambar 5. Karakter 'A' pada Level Tegangan RS 232

II.2.4. Konfigurasi Port Serial

Gambar 6. dibawah ini adalah gambar konektor port serial DB-9 pada bagian belakang CPU. Pada komputer IBM PC kompatibel biasanya dapat ditemukan dua buah konektor port serial DB-9 yang biasanya dinamai COM1 dan COM2.



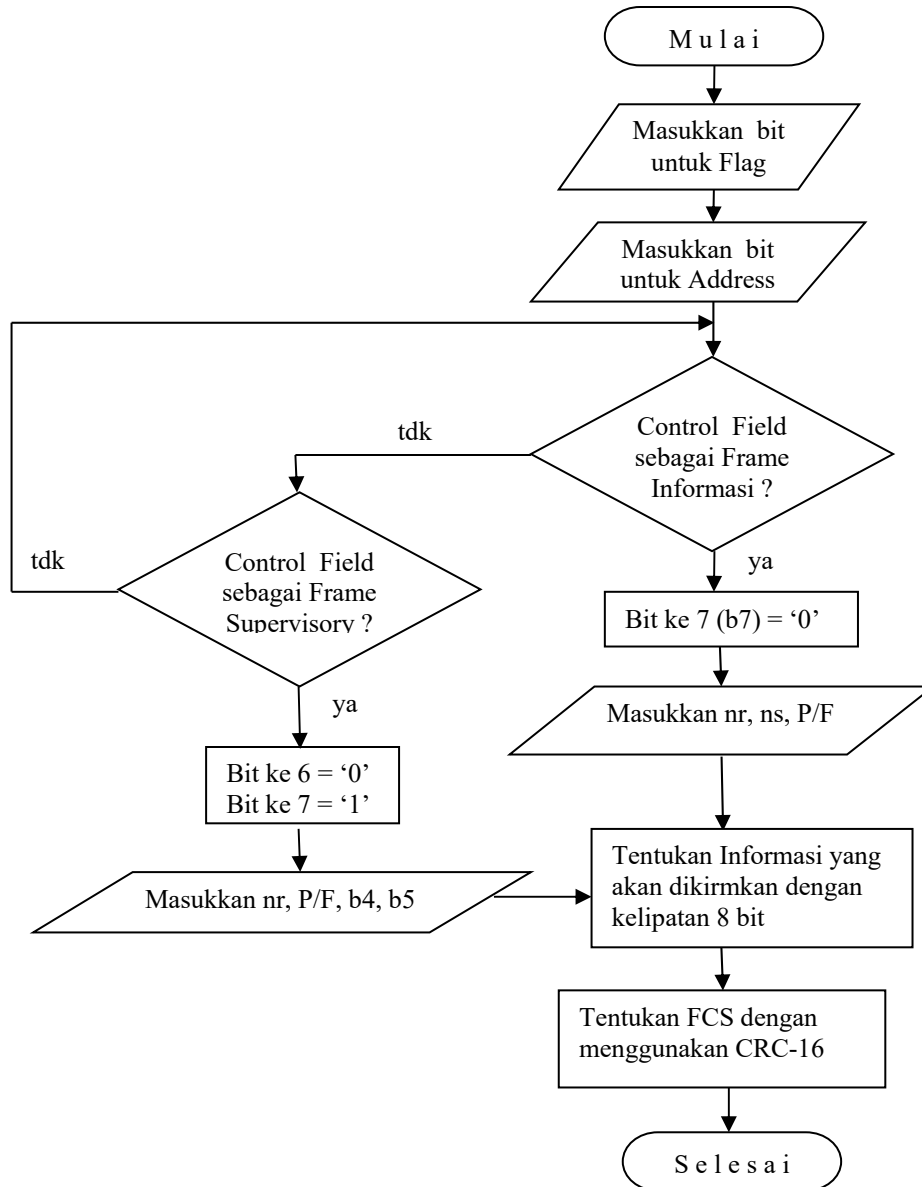
Gambar 6. Konektor port serial DB-9 pada bagian belakang CPU.

II.2.5. Pengaksesan Port Serial pada Visual Basic 6.0.

Untuk pengaksesan port serial pada Visual Basic kita dapat mengaksesnya secara langsung melalui register UART atau menggunakan kontrol MSComm yang telah disediakan Visual Basic 6.0.

III. METODOLOGI PENELITIAN

Perancangan simulasi pembentukan frame SDLC ini dilakukan dengan melakukan simulasi percobaan semu dengan bantuan perangkat lunak visual basic. Perancangan simulasi ini dilakukan dengan tahap-tahap proses seperti flow chart pada gambar 7.



Gambar 7. Flow Chart Pembentukan Format SDLC

III.1. Menentukan Flag

Pada saat start, diminta untuk menentukan dan memasukkan Flag pembuka / penutup. Pada keadaan ini ada pilihan untuk menentukan flag sebagai berikut :

1. Flag standard : 01111110
2. Flag pilihan sendiri

Pada pilihan 1, flag yang digunakan sebagai flag pembuka maupun flag penutup adalah flag standard yaitu '01111110' atau bilangan heksa 7EH, sedangkan pada pilihan 2. Susunan bit untuk flag dapat ditentukan sendiri dengan catatan jumlah bitnya harus 8 bit.

Apabila dipilih flag standard, maka langkah berikutnya adalah menentukan mode flag yang akan digunakan, yaitu :

1. Flag pembuka terpisah dengan flag penutup dengan keadaan idle bit '1'
2. Flag penutup untuk frame N merupakan flag pembuka untuk frame N+1
3. Bit '0' akhir dari flag penutup frame N merupakan awal bit '0' untuk flag pembuka untuk frame N+1.

Apabila flag yang digunakan dipilih sendiri, artinya tidak menggunakan flag standard, maka hanya berlaku flag pembuka terpisah dengan flag penutup.

III.2. Menentukan Alamat

Address atau alamat yang digunakan adalah 1 sampai dengan 254 dengan menggunakan 8 bit. Misalnya bila stasiun 7 yang diinginkan maka field address berisi bit 00000111, sedangkan stasiun primer tidak mempunyai alamat karena semua komunikasi menuju stasiun primer.

III.3. Menentukan Field Kontrol

Field control dapat menentukan jenis frame yang digunakan, pada penelitian ini ada dua pilihan yang diberikan, yaitu :

1. Sebagai frame informasi
2. Sebagai frame supervisory

Ketentuan dalam menggunakan field control sebagai frame informasi atau sebagai frame supervisory dapat dilihat pada penjelasan butir III.4.1. dan butir III.4.2. di atas.

III.4. Menentukan Field Informasi

Field informasi diisi dengan data karakter yang akan dikirimkan, dan kode yang digunakan adalah kode EBCDIC seperti yang dijelaskan pada butir II.2. Pada penelitian ini jumlah maksimum bit untuk data karakter sebanyak 32 bit, artinya jumlah maksimum karakter yang dikirimkan per satu frame adalah 4 karakter, ini disebabkan karena keterbatasan lebar penampilan untuk simulasi CRC nya agar dapat kelihatan secara keseluruhan.

III.5. Menentukan Field Frame Check Sequence (FCS)

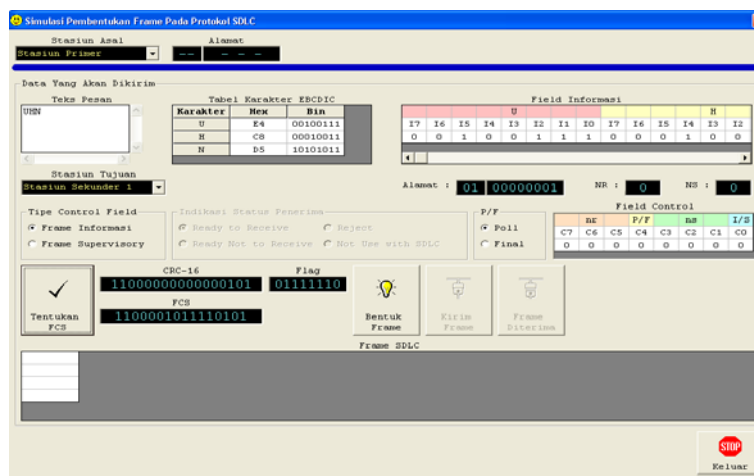
Frame Check Sequence merupakan sebuah kode untuk pengecekan kesalahan, kode FCS tersebut dihasilkan dengan metoda CRC-16 yang dihasilkan berdasarkan data karakter yang akan dikirimkan yang dalam hal ini panjang data karakter maksimum 32 bit. Cara dan metoda pengecekan kesalahan data dengan metoda CRC-16 juga ditampilkan pada hasil simulasi ini ²⁾.

2) *Sindak Hutauruk, 2010, Perancangan Simulasi Deteksi Kesalahan Data dengan Metoda CRC Menggunakan Perangkat Lunak Visual Basic, Hasil Penelitian, LP-UHN*

IV. HASIL RANCANGAN

IV.1. Rancangan Perangkat Lunak (Simulasi)

Perancangan dilakukan dengan berpedoman agar perangkat lunak simulasi yang dihasilkan harus dapat melihat seluruh tahapan proses pembentukan frame SDLC sehingga dapat dengan mudah dan jelas untuk dipahami dan dimengerti. Penjelasan tampilan masukan dan keluaran pada simulasi ini adalah sebagai berikut :



Gambar 8. Tampilan Simulasi Pembentukan Format Frame SDLC

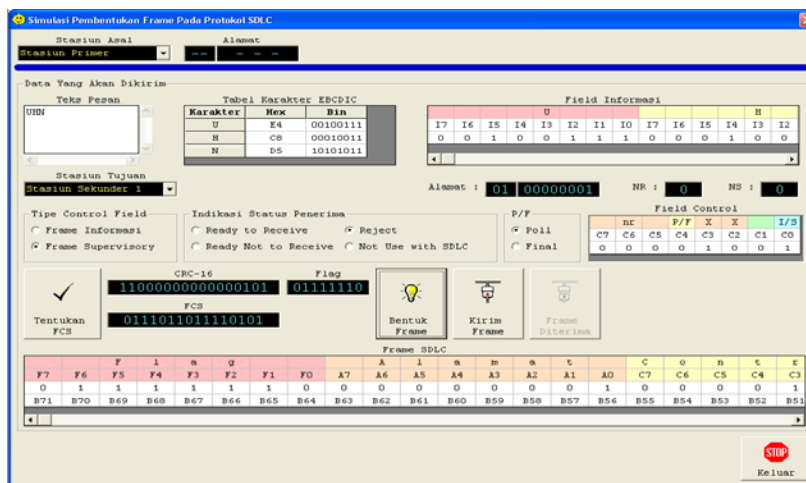
Gambar 8. Adalah tampilan awal dari eksekusi program simulasi, pada tampilan ini ditampilkan field masukan dan keluaran yaitu :

1. Field masukan stasiun asal, yaitu stasiun yang akan mengirimkan data teksnya. Pilihan stasiun asal tersebut dapat berupa stasiun primer atau stasiun sekunder, ada sebanyak 254 stasiun sekunder yang dapat dijadikan sebagai stasiun asal.
2. Field address atau alamat dari stasiun asal, karena yang digunakan untuk alamat sebanyak 8 bit maka jumlah stasiun yang dapat dialamati adalah sebanyak 256 stasiun dengan catatan alamat 00000000 dan 11111111 tidak digunakan sebagai alamat stasiun karena alamat tersebut digunakan untuk testing jaringan dan broadcast ke seluruh stasiun sekunder sedangkan alamat stasiun primer tidak diberikan karena semua data yang dikirim oleh stasiun sekunder menuju stasiun

- primer. Pada field alamat ditampilkan nomor alamat dalam bentuk heksadesimal dan biner, sebagai contoh alamat 15 diisi dengan heksa 0F dan biner 00001111.
3. Field teks pesan, panjang teks pesan yang digunakan maksimum 25 buah karakter. Pembatasan panjang karakter ini dilakukan agar tidak terlalu panjang pada penampilan dalam bentuk biner. Jenis kode karakter yang digunakan adalah kode EBCDIC, pesan karakter juga ditampilkan dalam bentuk tabel karakter EBCDIC dengan 3 kolom yaitu kolom dalam bentuk karakter, kolom dalam bentuk heksadesimal, dan kolom dalam bentuk biner. Keseluruhan pesan teks ini juga ditampilkan dalam bentuk biner pada field informasi.
 4. Field stasiun tujuan yang berisi stasiun yang akan dituju. Stasiun yang dituju dapat dipilih mulai dari stasiun 1 sampai dengan stasiun 254. Bila semua stasiun sekunder yang dituju maka dipilih alamat tujuan broadcast. Alamat stasiun tujuan ini juga ditampilkan dalam bentuk heksadesimal dan biner. Contoh apabila stasiun tujuan yang dituju adalah stasiun sekunder 7 maka akan ditampilkan juga alamat dalam bentuk heksadesimal 07 dan alamat dalam bentuk biner 00000111, dan bila keseluruhan stasiun sekunder yang dituju (broadcast) maka alamat yang ditampilkan dalam bentuk heksadesimal FF dan alamat dalam bentuk biner 11111111.
 5. Field Kontrol (Control Field) dapat dipilih sebagai frame informasi atau sebagai frame supervisory. Apabila dipilih sebagai frame informasi susunan bit pada field kontrol adalah 3 bit pertama C7, C6 dan C5 sebagai nr, C4 sebagai poll / final, C3, C2, dan C1 sebagai ns, dan C0 sebagai indikator frame informasi. C4 berisi bit 0 bila dipilih poll dan berisi bit 1 bila dipilih final. C0 berisi bit 0 sebagai indikator frame informasi. Sedangkan bila field kontrol dipilih sebagai frame supervisory maka akan muncul tampilan pilihan sebagai indikasi status penerima yaitu ready to receive, ready not to receive, reject, atau not use with SDLC. Tampilan susunan bit pada field kontrol adalah bit C7, C6, dan C5 sebagai nr, bit C4 sebagai poll atau final (sebagai poll berisi bit 0 dan sebagai final berisi bit 1), bit C3 dan C2 untuk menentukan indikasi status penerima, bila ready to receive maka C3 dan C2 berisi bit 0, bila ready not to receive maka C3 berisi bit 0 dan C2 berisi bit 1, bila reject maka C3 berisi bit 1 dan C2 berisi bit 0, dan bila not use

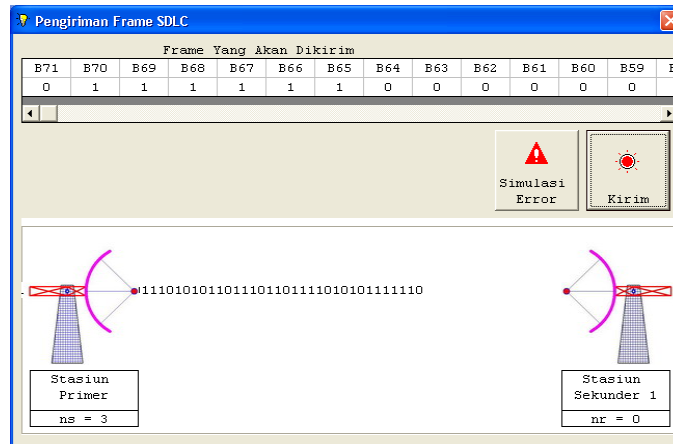
with SDLC maka C3 dan C2 berisi bit 1. Sedangkan C1 berisi bit 0 dan C0 berisi bit 1 digunakan sebagai indikator frame supervisory.

6. Field untuk CRC-16 yang digunakan dan field untuk flag ditampilkan. CRC-16 yang telah ditetapkan oleh CCITT mempunyai persamaan $P(X) = X^{16} + X^{15} + X^2 + X^0$ atau dengan susunan bit 1100000000000101 dan susuna bit flag adalah 01111110.
7. Field FCS ditentukan dengan menggunakan CRC-16 terhadap data karakter yang akan dikirim yang bila dieksekusi akan secara otomatis mengisi field FCS tersebut.
8. Setelah semuanya ditentukan maka bentuk frame dapat dieksekusi dengan menekan tombol pilihan 'bentuk frame' yang akan menampilkan frame SDLC seperti yang terlihat pada Gambar 9.



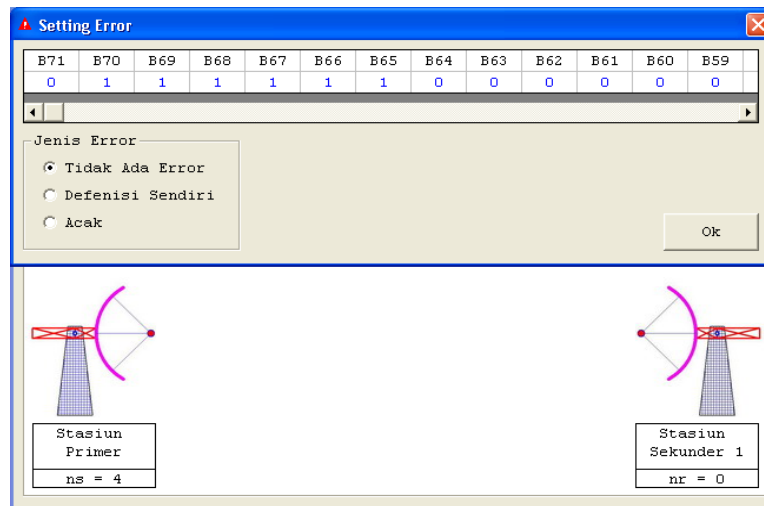
Gambar 9. Tampilan Simulasi Eksekusi Bentuk Frame SDLC

9. Selanjutnya untuk mengirimkan frame SDLC yang telah terbentuk, maka dieksekusi pilihan tombol 'kirim frame' yang akan menampilkan simulasi proses pengiriman frame dari stasiun pengirim ke stasiun penerima seperti yang diperlihatkan pada Gambar 10.

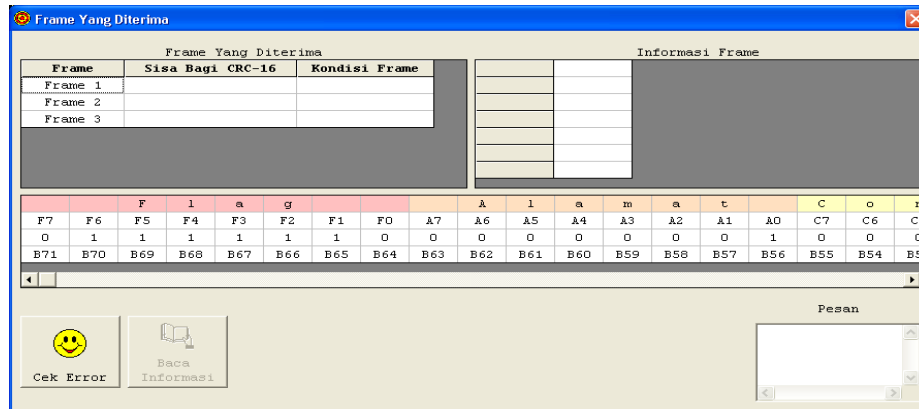


Gambar 10. Tampilan Simulasi Proses Pengiriman Frame SDLC

10. Bila kita anggap terjadi error atau kesalahan bit dalam pengiriman frame maka dipilih tombol pilihan 'simulasi error' yang akan memunculkan tampilan 3 pilihan yaitu tidak ada error, defenisi sendiri, atau acak. Seperti yang terlihat pada Gambar 11.
11. Untuk melihat frame SDLC yang diterima oleh penerima, maka pada field stasiun asal dipilih nama stasiun penerima, kemudian eksekusi tombol pilihan 'frame diterima' yang akan menampilkan frame yang diterima seperti pada Gambar 12.

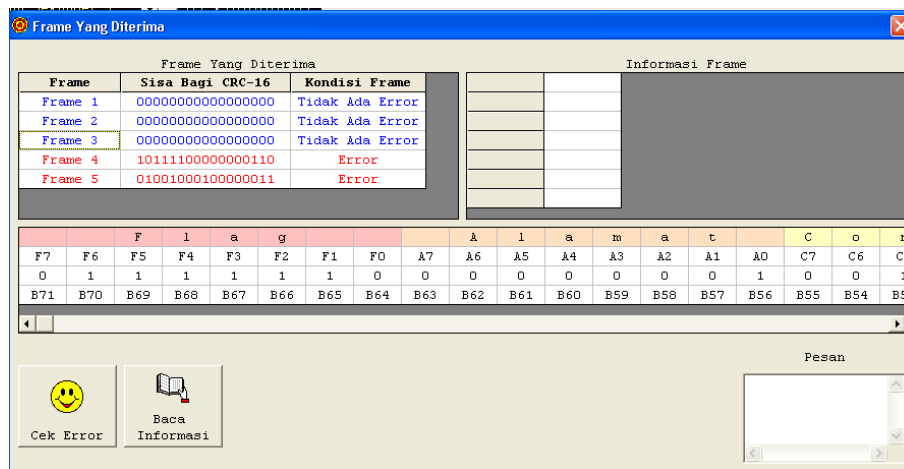


Gambar 11. Tampilan Simulasi untuk Pilihan Error yang Terjadi



Gambar 12. Tampilan Frame yang Diterima

12. Kesalahan atau error yang terjadi dapat di cek dengan mengeksekusi tombol pilihan 'cek error', hasil eksekusi ini seperti yang terlihat pada Gambar 13. Apabila tidak terjadi kesalahan maka sisa hasil bagi CRC-16 akan menghasilkan 0 dan ditampilkan bahwa kondisi frame tidak ada error atau kesalahan. Akan tetapi bila terjadi kesalahan maka sisa hasil bagi CRC-16 tidak menghasilkan 0 dan ditampilkan bahwa kondisi frame terjadi error atau kesalahan.



Gambar 13. Tampilan Ada Tidaknya Kesalahan yang Diterima

13. Untuk melihat informasi frame yang diterima maka eksekusi tombol pilihan 'baca informasi' sehingga informasi frame ditampilkan seperti yang terlihat pada Gambar 14.

V. KESIMPULAN DAN SARAN

V.1. Kesimpulan

Realisasi simulasi komunikasi data multiframe SDLC ini dapat dilakukan dengan baik pada komputer walaupun dengan beberapa keterbatasan dalam penampilan di layar monitor khususnya untuk jumlah bit pesan yang panjang. Semua proses pengiriman multiframe dapat dilakukan dan ditampilkan dengan rinci dan jelas sehingga sudah cukup baik untuk digunakan sebagai salah satu modul percobaan semu di laboratorium maupun pada mata kuliah komunikasi data.

V.2. Saran

Dari hasil perancangan ini, disarankan hal-hal berikut :

1. Disarankan agar hasil rancangan ini dapat digunakan sebagai modul percobaan semu pada laboratorium.
2. Disarankan agar metoda ini dapat juga disimulasikan dengan menggunakan komunikasi antar 2 buah komputer tanpa kabel (nirkabel).

DAFTAR PUSTAKA

- Adi Kurniadi, 2005, *Pemrograman Microsoft Visual Basic 6*, Elex Media Komputindo, Jakarta
- J. Dunlop and D.G. Smith, 1995, *Telecommunications Engineering*, Third Edition, Chapman & Hall, London
- Man Young Rhee, 1989, *Error Correcting Coding Theory*, McGraw-Hill International Editions, Singapore
- Michael Halvorson, 2001, *Step By Step Microsoft Visual Basic 6.0 Professional*, Elex Media Komputindo, Jakarta
- Sindak Hutauruk, 2010, *Perancangan Simulasi Deteksi Kesalahan Data dengan Metoda CRC Menggunakan Perangkat Lunak Visual Basic*, Hasil Penelitian, LP-UHN
- Sindak Hutauruk, 2010, *Perancangan Simulasi Pembentukan Frame Pada Protokol SDLC Dengan Perangkat Lunak Visual Basic*, Hasil Penelitian, LP-UHN
- Wahana Komputer, 2005, *Pemrograman Visual Basic.net 2005*, edisi 1, Penerbit Andi, Yogyakarta
- William Stalling, 2000, *Data & Computer Communication*, 6th Edition, Prentice Hall Inc., New Jersey
- Wayne Tomasi, 1994, *Advanced Electronic Communications Systems*, Third Edition, Prentice Hall International, Inc., USA