

PERCOBAAN IV Komunikasi Data MODEM

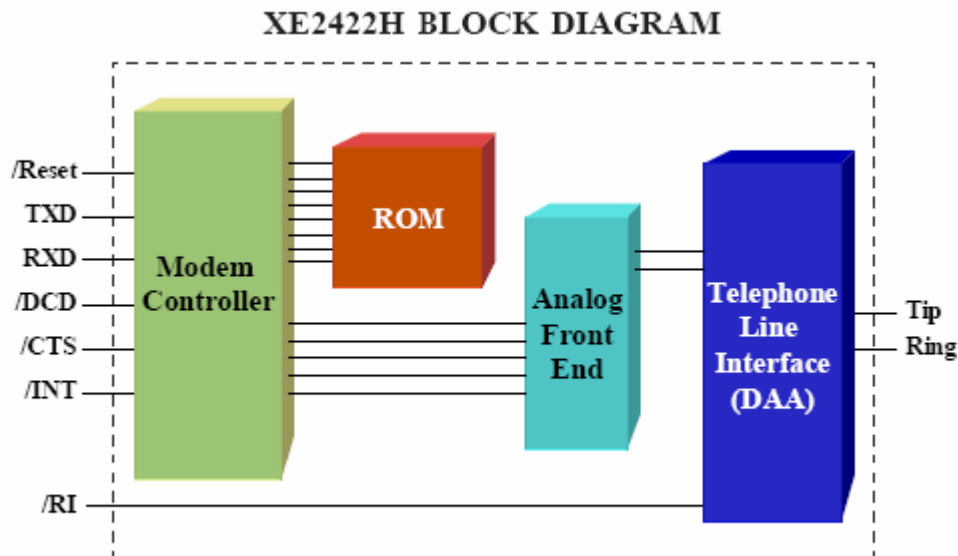
1. TUJUAN

Setelah melaksanakan praktikum ini mahasiswa diharapkan mampu :

- Melakukan koneksi antar 2 PC menggunakan dial up modem untuk kirim dan terima karakter dan file
- Memahami penggunaan protokol di modem
- Memahami bentuk sinyal dalam pengiriman data

2. DASAR TEORI

Sebuah modem adalah peralatan yang mentranslasikan data digital dari komputer ke sinyal analog yang akan dilewatkan jalur telepon. Modem di sisi yang lain, mengkonversikan sinyal analog yang diterima dari jalur telepon menjadi data yang akan dibaca oleh komputer. Kecepatan modem dinyatakan dalam Kbps (kilobit per second), yang merupakan ukuran dari data yang akan ditranslasikan dan dikirim. Blok diagram sebuah modem ditunjukkan seperti gambar 4.1.



Gambar 4.1. Blok Diagram Modem

Berdasarkan gambar di atas dapat ditunjukkan di sebelah kiri modem terhubung ke sisi UART (*Universal Asynchronous Receiver / Transmitter*) dari PC. Unit ini mengatur keluar masuknya data dari dan ke PC secara serial (melalui jalur TxD dan RxD) dan melakukan fungsi-fungsi control (*Clear to Send, Data Carrier Detect* dan *Ring Indicator*). Di sebelah kanan, modem terhubung dengan jalur telepon analog (two-wire, tip dan ring). Pada sisi yang terhubung ke jalur telepon ini data sudah mengalami modulasi (biasanya menggunakan modulasi FSK).



Gambar 4.2. Bentuk fisik Internal Modem

Proses translasi data dari pengirim dan penerima ditangani oleh protocol di masing-masing modem. Seluruh protocol data bekerja berdasarkan potongan blok-blok dari bit-bit data yang dibawa dan mengirimkan ke modem tujuan, yang selanjutnya akan mengecek error di masing-masing blok data, dan kemudian, berdasarkan hasilnya dan tipe protokolnya, modem penerima tadi akan mengembalikan sinyal positif Acknowledgement (ACK) maupun *negative acknowledgement* (NAK) ke modem pengirim. Jenis protocol modem dibedakan berdasarkan tipe pengecekannya, yaitu tipe pengecekan *Error-Correcting* dan *checksum*.

Metode *checksum* adalah metode yang menghitung jumlah bit dalam urutan transmisi yang bisa di-cek oleh receiver. Receiver akan mengecek jumlah bit yang diterima. Jika jumlah bit yang diterima sama dengan jumlah bit yang dikirim, maka dianggap bahwa data yang diterima sudah lengkap. Layer komunikasi TCP maupun UDP mengaplikasikan metode *checksum* ini dengan verifikasi sebagai salah satu layanannya.

Metode *Cyclic Redundancy Checking* adalah metode pengecekan untuk error-error data yang ditransmisikan dalam kanal komunikasi. Peralatan di sisi pengirim menggunakan sistem polynomial 16 bit atau 32 bit dalam satu blok data yang dikirim, dan menambahkan hasil *Cyclic Redundance Code* (CRC) nya pada blok tadi. Sisi penerima juga mengaplikasikan sistem polynomial yang sama, mengecek hasilnya dan membandingkan dengan kode CRC yang menyertai. Jika sama, maka data yang diterima dianggap benar (proses pengiriman sukses). Jika tidak, pengirim akan mengirim ulang blok data yang sama.

Beberapa protocol modem yang dikenal antara lain : XModem, YModem, ZModem V.24, V.90 dsb. Jenis protocol beserta tipe pengecekan dan jumlah bit per-blok ditunjukkan pada Tabel 4.1.

Tabel 4.1. Jenis Protocol Modem

Protocol	Block size	Frequency of response	Type of checking	Other information
Xmodem	128 bytes	Every block	checksum	Also called MODEM7
Xmodem CRC	128 bytes	Every block	cyclic redundancy checking	Cyclic redundancy checks detect errors that checksum won't
Xmodem-1K	1024 bytes	Every block	Cyclic redundancy check	Better for large files
WXmodem	128 bytes	Every block but doesn't hold up the sender before accepting the next block	Checksum	More efficient than Xmodem
Ymodem	1024 bytes	Every block	Cyclic redundancy check	Includes a batch mode that allows multiple files to

				be sent with one command
Ymodem-g	1024 bytes	Only when an error is detected to blocks sent as a stream and all must arrive successfully	Cyclic redundancy check	Supports batch transfers
Zmodem	512 bytes	When a block with an error is detected	Cyclic redundancy check	A transmission can be interrupted and resumed without retransmitting blocks already sent.
Kermit	Adjusts to the computer system	When a block with an error is detected	Checksum	Can resynchronize transmissions after a line interruption
V.42 (LAPM)	128 bytes	Up to 15 blocks (frames) can be sent before a response	Cyclic redundancy check	This takes precedence over V.42 MNP4.
V.42 (MNP4)	Varies	When a packet (block) with an error is detected	Cyclic redundancy check	Used when LAPM can't be used by both modems.

3. Peralatan :

- PABX
- 2 Laptop dilengkapi dengan internal modem untuk *client*

4. Tugas Pendahuluan

1. Jelaskan dan gambarkan blok diagram komunikasi menggunakan modem
2. Buatlah file dalam bentuk Microsoft word yang berukuran 50 Kbyte, 100 Kbyte dan 150 Kbyte

5. Langkah Percobaan

Send & Receive Data Text

1. Buka Hyperterminal : **START | Accessories | Communication | Hyperterminal.**
2. Pada sisi local : **New Connection | Enter a name for the connection ...** (ketik sembarang)
3. **Enter detail for the phone number to dial ...** (ketik nomor telepon yang akan dihubungi)
-> Dialing.



Gambar 4.3 Koneksi dengan modem

NB:

Phone number: Isikan dengan no tujuan PABX yang akan dihubungi
Connect using: Isikan dengan modem yang terinstall di laptop

4. Pada sisi remote: Buka Hyperterminal | Pilih **Call | wait for a call.**
5. Tunggu sampai ada pesan **Connected** pada sisi bawah.
6. Lakukan proses *Transmit Receive Karakter* antar dua PC, lakukan sebanyak 10 baris. Perhatikan, apakah semua karakter yang dikirim bisa diterima dengan baik. Jika ada yang salah, berapa persen kesalahan terhadap total seluruh karakter yang dikirim.

NB :

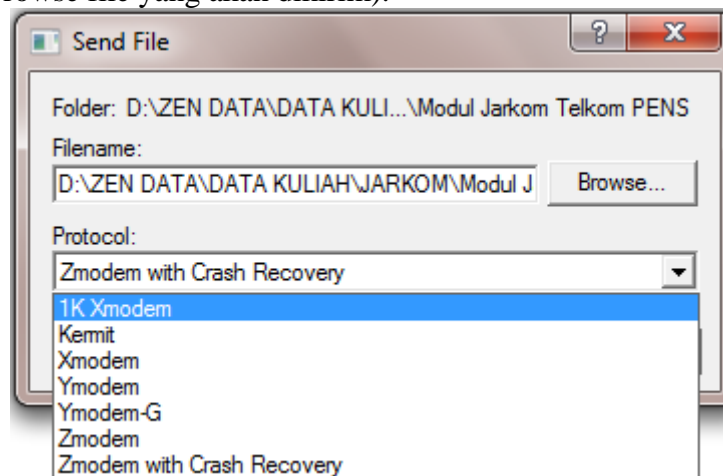
Untuk komunikasi modem, juga bisa menggunakan **at** command, sbb :

- a. Sisi pengirim
atd802 => 802 adalah no telp yang dihubungi

- b. Sisi penerima
ata

Send & Receive Data File

1. Pada sisi Transmit (Tx) : Masih membuka Hyperterminal ⌚ Pilih **Send File** ⌚ **File name** (browse file yang akan dikirim).



Gambar 4.4 Parameter pada sisi TX

2. Lakukan secara bergantian untuk pilihan protocol berikut ini :

- 1K XModem
- Kermit
- XModem
- YModem
- ZModem
- ZModem with Crash Recovery

NB:

Jika dengan ZModem with Crash Recovery, lakukan 2x karena jika sudah pernah mengirim file dengan nama yang sama, maka yang kedua akan jauh lebih cepat.

3. Dengan file yang sama, lakukan pengiriman menggunakan protocol-protocol di atas.
4. Catat waktu yang diperlukan untuk mengirim sebuah file (*time elapsed*) dan *throughput* nya.
5. Pada sisi *Receive (Rx)* : Dengan Hyperterminal pilih *Transfer / Receive File*
6. Jenis Protocol, harus sesuai dengan protocol yang dipilih sisi Transmit.

7. Isikan data pada table berikut, gunakan ukuran file yang sama yaitu 100Kbyte

Ukuran file : 100Kbyte ; Parameter :								
Protocol	TX			RX			Banyak paket	Error checking
	Waktu	Th (cps)	Th (bps)	Waktu	Th (cps)	Th (bps)		
1K XModem								
Kermit								
XModem								
YModem								
ZModem								
ZModem with CR								

NB:

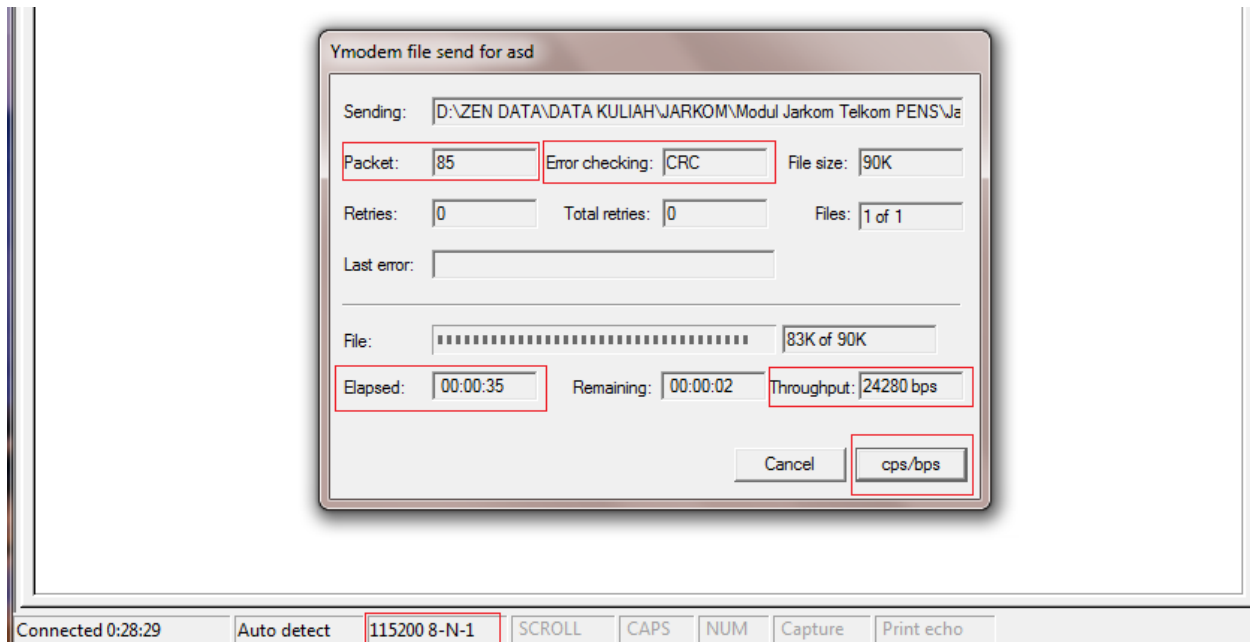
Th : Throughput

Secara teori, bila kita mengirim file 100Kbyte dengan kecepatan 300 bps, maka

Throughput = (data bps) / 10 ; satuan cps, 10 dari 8N1

Waktu = (ukuran file) / throughput ; satuan detik

Data di table diatas, diambil dari gambar berikut :



Gambar 4.5 Parameter yang diambil datanya

Untuk Packet dan Error Checking, ada beberapa protocol yang tidak muncul, tulis apa adanya.

8. Ulangi langkah diatas dengan merubah-rubah ukuran file, dan isikan table berikut ini :

Protocol	RX (50Kbyte)		RX (100Kbyte)		RX (150Kbyte)	
	Waktu	Th (cps)	Waktu	Th (cps)	Waktu	Th (cps)
	1K XModem					
Kermit						
XModem						
YModem						
ZModem						
ZModem with CR						

NB:

Ambil data di bagian RX saja, untuk Th (Throughput) ambil sebagai cps.

Untuk data yang 100Kbyte, gunakan data pada poin 7.

9. Gambarkan grafik :

- Pada poin 7, gunakan grafik batang untuk membandingkan perbedaan waktu dan throughput masing-masing protocol.
- Pada poin 8, gunakan grafik garis untuk membandingkan perbedaan throughput masing-masing protocol jika ukuran file dinaikkan.
- Pada poin 8, gunakan grafik garis untuk membandingkan perbedaan waktu masing-masing protocol jika ukuran file dinaikkan.

6. Pertanyaan & Tugas :

- Dari ukuran file yang sudah diterima pada proses send & receive file, bandingkan antara *throughput* yang diterima untuk masing-masing protocol dengan *blok size* dari setiap protocol pada Tabel diatas. Beri komentar. Dibagi menjadi berapa blok datakah satu file tersebut ?
- Apa perbedaan protokol modem yang digunakan praktikum diatas ?