

# KONSEP KOMUNIKASI SERIAL

## 3.1 Pendahuluan

Antarmuka Kanal serial lebih kompleks/sulit dibandingkan dengan antarmuka melalui kanal paralel (akan dibahas pada bab berikutnya), hal ini disebabkan karena:

1. Dari Segi perangkat keras: adanya proses konversi data paralel menjadi serial atau sebaliknya menggunakan piranti tambahan yang disebut UART (*Universal Asynchronous Receiver/Transmitter*); dan
2. Dari Segi perangkat lunak: lebih banyak register yang digunakan atau terlibat;

Namun di sisi lain antarmuka kanal serial menawarkan berapa kelebihan dibandingkan secara paralel, antara lain:

1. **Kabel untuk komunikasi serial bisa lebih panjang dibandingkan dengan paralel;** data-data dalam komunikasi serial dikirim-kan untuk logika '1' sebagai tegangan -3 s/d -25 volt dan untuk logika '0' sebagai tegangan +3 s/d +25 volt, dengan demikian tegangan dalam komunikasi serial memiliki ayunan tegangan maksimum 50 volt, sedangkan pada komunikasi paralel hanya 5 volt. Hal ini menyebabkan gangguan pada kabel-kabel panjang lebih mudah diatasi dibandingkan pada paralel;
2. **Jumlah kabel serial lebih sedikit;** Anda bisa menghubungkan dua perangkat komputer yang berjauhan dengan hanya 3 kabel untuk konfigurasi *null modem*, yaitu TXD (saluran kirim), RXD(saluran terima) dan Ground, bayangkan jika digunakan teknik paralel akan terdapat 20 - 25 kabel! Namun pada masing-masing komputer dengan komunikasi serial harus dibayar "biaya" antarmuka serial yang agak lebih mahal;
3. **Banyaknya piranti saat ini (*palmtop, organizer, hand-phone dan lain-lain*) menggunakan teknologi infra merah untuk komunikasi data;** dalam hal ini pengiriman datanya dilakukan secara serial. IrDA-1 (spesifikasi infra merah pertama) mampu mengirimkan data dengan laju 115,2 kbps dan

dibantu dengan piranti UART, hanya panjang pulsa berkurang menjadi 3/16 dari standar RS-232 untuk menghemat daya;

4. **Untuk teknologi *embedded system*, banyak mikrokontroler yang dilengkapi dengan komunikasi serial (baik seri RISC maupun CISC) atau *Serial Communication Interface* (SCI); dengan adanya SCI yang terpadu pada IC mikrokontroler akan mengurangi jumlah pin keluaran, sehingga hanya dibutuhkan 2 pin utama TxD dan RxD (di luar acuan *ground*).**

## 3.2 TINJAUAN PERANGKAT KERAS

### 3.2.1. Spesifikasi Perangkat Keras

Piranti-piranti yang menggunakan komunikasi serial meliputi:

- DTE = *Data Terminal Equipment*, yaitu komputer itu sendiri;
- DCE = *Data Communication Equipment*, misalnya modem, *plotter* dan lain-lain;

Beberapa parameter yang ditetapkan EIA (*Electronics Industry Association*) antara lain:

- Sebuah 'spasi' (logika 0) antara tegangan +3 s/d +25 volt;
- Sebuah 'tanda' (logika 1) antara tegangan -3 s/d -25 volt;
- Daerah tegangan antara +3 s/d -3 volt tidak didefinisikan (*undefined*);
- Tegangan rangkaian terbuka tidak boleh lebih dari 25 volt (dengan acuan *ground*);
- Arus hubung-singkat rangkaian tidak boleh lebih dari 500 mA. Sebuah penggerak (*driver*) harus mampu menangani arus ini tanpa mengalami kerusakan.

Tabel 3.1. Perbedaan Pinout Untuk DB-9 Dan DB-25 (Male)

Pin DB25	Pin DB9	Singkatan	Keterangan
Pin 2	Pin 3	TD	Transmit Data
Pin 3	Pin 2	RD	Receive Data
Pin 4	Pin 7	RTS	Request To Send
Pin 5	Pin 8	CTS	Clear To Send
Pin 6	Pin 6	DSR	Data Set Ready
Pin 7	Pin 5	SG	Signal Ground
Pin 8	Pin 1	CD	Carrier Detect
Pin 20	Pin 4	DTR	Data Terminal Ready
Pin 22	Pin 9	RI	Ring Indikator

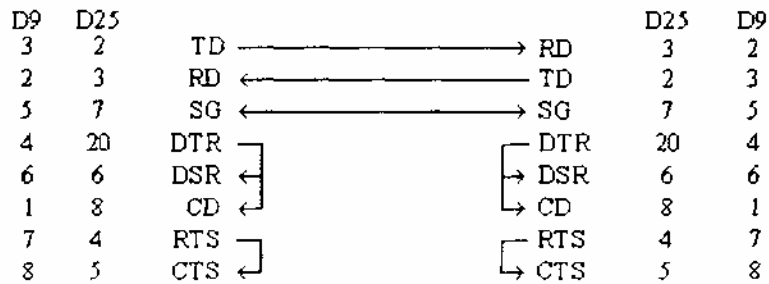
Tabel 3.2. Fungsi-fungsi Kaki (pena/pin)

Singkatan	Keterangan	Fungsi
TD	Tranmisit Data	Untuk pengiriman data serial (TDX)
RD	Receuve Data	Untuk penerimaan data serial (RDX)
CTS	Clear To Send	Digunakan untuk memberitahukan bahwa Modem siap untuk melakukan pertukaran data
DCD	Data Carrier Detect	Saat modem mendeteksi suatu `Carieer` Dari Modem lain (dari tempat lain) maka signal ini akan diaktifkan
DSR	Data Set Ready	Memberitahuakan UART bahwa Modem siap untuk melakukan komunikasi (Link)
DTR	Data Terminal Ready	Kebalikan dari DSR, untuk memberitahukan bahwa UAT siap melakukan hubungan komunikasi
RTS	Regues To Send	Signal untuk menginformasikan modem Bahwa UART siap melakukan pertukaran data
RI	Ring Indikator	Akan aktif jika modem mendeteksi adanya signal dering dari saluran telepon

### 3.2.2. Konfigurasi Null Modem

Konfigurasi *Null Modem* digunakan untuk menghubungkan dua DTE, dengan diagram pengkabelan yang dapat dilihat pada gambar 3.1. Dalam hal ini hanya dibutuhkan tiga kabel antar DTE, yaitu untuk **TxD**, **RxD** dan **Gnd**. Cara kerjanya cukup mudah: yaitu bagaimana membuat komputer agar

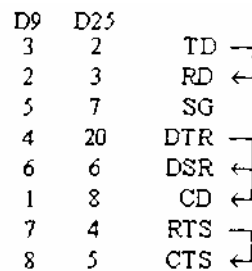
mengira dia berkomunikasi dengan modem (DCE) bukan dengan komputer lainnya.



Gambar 3.1. Diagram Pengkabelan Konfigurasi Null Modem

Pada gambar 3.1 terlihat bahwa kaki DTR (*Data Terminal Ready*) dihubungkan ke DSR (*Data Set Ready*) dan juga ke CD (*Carrier Detect*) pada masing-masing komputer, sehingga pada saat sinyal DTR diaktifkan maka sinyal DSR dan CD juga ikut aktif (konsep Modem Semu atau *Virtual Modem*). Karena komputer dalam hal ini melakukan pengiriman data dengan kecepatan yang sama, maka kontrol aliran (*flow control*) belum dibutuhkan sehingga RTS (*Request To Send*) dan CTS (*Clear To Send*) pada masing-masing komputer saling dihubungkan. Sedangkan untuk pengujian port serial bisa digunakan konfigurasi *Loopback Plug* sebagaimana ditunjukkan pada gambar 3.2.

#### LoopBack Plug



Gambar 3.2. Pengkabelan pada Konfiurasi Loopback

Jika anda memasang plug ini pada komputer dengan perangkat lunak terminal (misalnya Hyperterminal pada Sistem Operasi Windows), maka apa yang Anda ketikkan akan dimunculkan lagi (*echoing*).

### 3.2.3. Laju Kecepatan DTE/DCE

Laju kecepatan pengiriman data yang sering dibicarakan adalah Iaju kecepatan DTE ke DCE (antara PC dan modem atau disebut juga sebagai Iaju kecepatan terminal (*terminal speed*)) dan Iaju kecepatan DCE ke DCE (antar modem yang berkomunikasi atau disebut juga sebagai Iaju kecepatan jalur (*line speed*)).

Jika Anda menggunakan modem 28,8K atau 36,6K, maka artinya kecepatan ini mengacu pada Iaju kecepatan DCE ke DCE. Jika digunakan UART 16550a, maka Iaju kecepatan maksimumnya adalah 115.200 bps, sedangkan kebanyakan perangkat lunak yang digunakan saat ini digunakan untuk mengatur Iaju kecepatan DTE ke DCE.

Interupsiin itu banyak modem saat ini beredar di pasaran dilengkapi dengan fasilitas kompresi-dekompresi (pemampatan-penguraian) data. Biasanya rasionya sekitar 1:4 (untuk berkas teks), dengan demikian jika dilakukan transfer data dengan Iaju 28,8K (DCE ke DCE), dengan modem yang berfasilitas kompresi data, maka artinya Anda sebenarnya mengirimkan data tersebut dengan Iaju 115,2Kbps (DTE ke DCE). Hal ini yang menyebabkan Iaju DTE ke DCE bisa lebih besar dari DCE ke DCE. Namun ada juga modem yang bisa melakukan kompresi hingga rasionya mencapai 1:8, sehingga kecepatan Iaju DTE ke DCE bisa mencapai 168.800 bps (modem <--> UART). Jika digunakan 16550a, yang kecepatannya maksimum hanya 115.200 bps, akan sia-saia saja, sehingga sebaiknya digunakan UART 16550c yang kecepatan pengiriman datanya bisa mencapai 230.400 bps. Angka-angka yang disebutkan adalah angka-angka maksimum, kenyataannya bisa lebih rendah/kecil.

### 3.2.4. Kontrol Aliran (*Flow Control*)

Jika Iaju kecepatan DTE ke DCE lebih cepat dibandingkan dengan DCE ke DCE, lambat-laun akan menyebabkan kehilangan data atau isitilahnya terjadi *buffer overflow*, dengan demikian dibutuhkan kontrol aliran baik secara perangkat lunak maupun perangkat keras.

Jika anda memasang plug ini pada komputer dengan perangkat lunak terminal (misalnya Hyperterminal pada Sistem Operasi Windows), maka apa yang Anda ketikkan akan dimunculkan lagi (*echoing*)'.

### **3.2.3. Laju Kecepatan DTE/DCE**

Laju kecepatan pengiriman data yang sering dibicarakan adalah Iaju kecepatan DTE ke DCE (antara PC dan modem atau disebut juga sebagai Iaju kecepatan terminal (*terminal speed*)) dan Iaju kecepatan DCE ke DCE (antar modem yang berkomunikasi atau disebut juga sebagai Iaju kecepatan jalur (*line speed*)).

Jika Anda menggunakan modem 28,8K atau 36,6K, maka artinya kecepatan ini mengacu pada Iaju kecepatan DCE ke DCE. Jika digunakan UART 16550a, maka Iaju kecepatan maksimumnya adalah 115.200 bps, sedangkan kebanyakan perangkat lunak yang digunakan saat ini digunakan untuk mengatur Iaju kecepatan DTE ke DCE.

Interupsiin itu banyak modem saat ini beredar di pasaran dilengkapi dengan fasilitas kompresi-dekompresi (pemampatan-penguraian) data. Biasanya rasionya sekitar 1:4 (untuk berkas teks), dengan demikian jika dilakukan transfer data dengan Iaju 28,8K (DCE ke DCE), dengan modem yang berfasilitas kompresi data, maka artinya Anda sebenarnya mengirimkan data tersebut dengan Iaju 115,2Kbps (DTE ke DCE). Hal ini yang menyebabkan Iaju DTE ke DCE bisa lebih besar dari DCE ke DCE. Namun ada juga modem yang bisa melakukan kompresi hingga rasionya mencapai 1:8, sehingga kecepatan Iaju DTE ke DCE bisa mencapai 168.800 bps (modem <--> UART). Jika digunakan 16550a, yang kecepatannya maksimum hanya 115.200 bps, akan sia-saia saja, sehingga sebaiknya digunakan UART 16550c yang kecepatan pengiriman datanya bisa mencapai 230.400 bps. Angka-angka yang disebutkan adalah angka-angka maksimum, kenyataannya bisa lebih rendah/kecil.

### **3.2.4. Kontrol Aliran (*Flow Control*)**

Jika Iaju kecepatan DTE ke DCE lebih cepat dibandingkan dengan DCE ke DCE, lambat-laun akan menyebabkan kehilangan data atau isitilahnya terjadi

*buffer overflow*, dengan demikian dibutuhkan kontrol aliran baik secara perangkat lunak maupun perangkat keras.

Kontrol aliran melalui perangkat lunak yang biasa digunakan adalah **Xon/Xoff**, yaitu dengan cara mengirimkan karakter Xon (ascii 17) dan Xoff (ascii 19) yang masing-masing membutuhkan panjang data terkirim total 10 bit, sehingga akibatnya akan memperlambat laju kecepatan, namun dari sisi perangkat keras tidak menambah jumlah kabel serial. Karakter **Xon** digunakan sebagai tanda bahwa modem siap untuk menerima data berikutnya, sedangkan karakter **Xoff** digunakan sebagai sinyal untuk menghentikan pengiriman data dari komputer.

Sedangkan kontrol aliran melalui perangkat keras menggunakan sinyal RTS (*Request To Send*) dan CTS (*Clear To Send*), sehingga dalam hal ini perlu ditambahkan dua kabel lagi namun dari sisi perangkat lunak tidak dibutuhkan tambahan bit, sehingga tidak akan menurunkan laju kecepatan. Pada saat komputer ingin mengirimkan data maka akan diaktifkan sinyal RTS, jika modem masih memiliki ruang penyimpan sementara (*buffer*), maka modem akan mengirimkan jawaban berupa sinyal CTS.

### 3.2.5. UART (8250 dan Kompatibelnya)

UART merupakan kepanjangan dari *Universal Aysnchronous Receiver I Trasmmitter*. Seri8250, yang mencakup 16450, 16550, 16650 dan 16750, merupakan jenis UART yang banyak digunakan, pada gambar II.3 ditunjukkan diagram pin dari jenis UART ini. Ada kalanya UART ini terpadu dalam suatu chip bersama-sama dengan kontrol kanal paralel, kanal game, hard disk dan floppy drive.

Keping 16550 merupakan kompatibelnya 8250 dan 16450, perbedaannya terletak pada pin 24 dan 29:

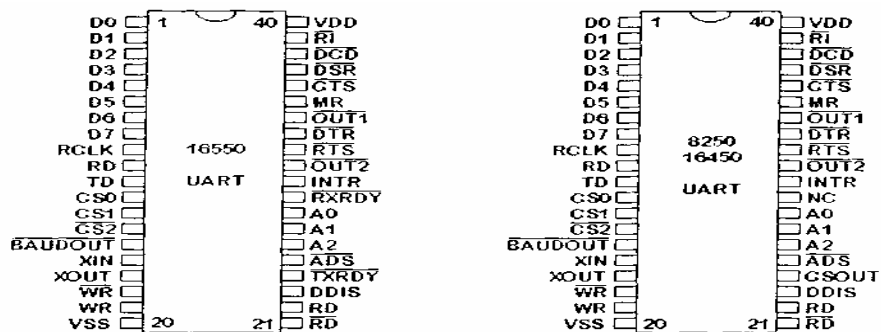
Kaki	16550	8250/16450
24	$\overline{TXRDY}$	CSOUT
29	$\overline{RXRDY}$	Tidak dihubungkan

Pada 16550 terdapat sinyal  $\overline{TXRDY}$  (*Transmit Ready*) dan  $\overline{RXRDY}$  (*Receive Ready*) yang dapat digunakan untuk implementasi DMA (*Direct Memory Access*) dengan dua mode kerja (operasional):

1. **Mode 0 - Single Transfer DMA:** lebih dikenal juga dengan mode 16450, mode ini diaktifkan dengan cara menon-aktifkan

FIFO (bit-0 FCR = 0) atau dengan mengaktifkan FIFO dan memilih mode DMA (bit-3 FCR = 1). Sinyal *RXRDY* akan aktif (rendah) jika ada (minimal) sebuah karakter pada penyangga penerima dan akan kembali non-aktif (tinggi) jika tidak ada satupun karakter pada penyangga penerima, sedangkan sinyal *TXRDY* akan aktif jika penyangga pengirim kosong sama sekali dan akan kembali non-aktif (tinggi) setelah karakter 1 byte pertama diisikan ke penyangga pengirim.

2. **Mode 1 - Multi Transfer DMA:** dipilih dengan syarat FCR bit-0 = 1 dan FCR bit-3 = 1. Pada mode ini, sinyal *RXRDY* akan aktif (rendah) jika telah tercapai tingkat picuan (*trigger level*) atau saat munculnya *time-out* 16550 dan akan kembali non-aktif jika sudah tidak ada satupun karakter yang tersimpan dalam FIFO. Sinyal *TXRDY* akan aktif (rendah) jika tidak ada karakterpun pada penyangga pengirim dan akan non-aktif jika penyangga pengirim FIFO sudah betul-betul penuh.



Gambar 3.3. Diagram pin UART 16550 dan 8250/16450

Semua chip UART kompatibel dengan TTL (termasuk sinyal TxD, RxD, RI, DCD, DTS, CTS, DTR dan RTS), dengan demikian diperlukan konverter tingkat RS232 (*RS232 level converter*) yang berfungsi untuk mengkonversi sinyal TTL menjadi logika tingkat RS232. Interupsiin itu UART juga membutuhkan *clock* untuk operasionalnya, biasanya dibutuhkan kristal eksternal dengan frekuensi 1,8432 MHz atau 18,432 MHz.



Tabel 3.3. Fungsi PinOut UART 16550 dan 8250/16450

Pin	Nama	Keterangan
Pin 1:8	D0:D7	Bus Data
Pin 9	RCLK	Masukan Clock penerima. Frekuensinya harus sama dengan baud-rate x26
Pin 10	RD	Terima Data
Pin 11	TD	Kirim Data
0Pin 12	CS0	Chip select 0 – Aktif Tinggi
Pin 13	CS1	Chip select 1 – Aktif Rendah
Pin 14	$\overline{CS2}$	Chip select 2 – Aktif Rendah
Pin 15	$\overline{BOUDOUT}$	Keluaran Baud – Keluaran dari Pembangkit Baud Rate Terprogram. Frekuensi = (baud rate x 16)
Pin 16	XIN	Masukan kristal eksternal –Digunakan untuk osilator pembangkit Boud Rate
Pin 17	XOUT	Keluran Kristal Eksternal
Pin 18	$\overline{WR}$	Jalur Tulis – Aktif Rendah
Pin 19	WR	Jalur Tulis – Aktif Tinggi
Pin 20	VSS	Dihubungkan ke ground
Pin 21	RD	Jalur Baca– Aktif Tinggi
Pin 22	$\overline{RD}$	Jalur Baca – Aktif Rendah
Pin 23	DDIS	Drive disable. Pin ini akan rendah saat CPU membaca dari UART. Dapat dihubungkan bus data kapasitas tinggi
Pin 24	$\overline{TXRDY}$	Transmit Ready – Siap kirim
Pin 25	$\overline{ADS}$	Address Store. Digunakan jika sinyal tidak stabil interupsima siklus baca atau tulis
Pin 26	A2	Bit alamat 2
Pin 27	A1	Bit alamat 1
Pin 28	A0	Bit alamat 0
Pin 29	$\overline{RXRDY}$	Receive Ready (siap terima data)
Pin 30	INTR	Intrrupt Output (keluaran interupsi)
Pin 31	$\overline{OUT2}$	User Output 2 (keluaran pengguna2)
Pin 32	$\overline{RTS}$	Reguest to Send (permintaan pengiriman)
Pin 33	$\overline{DTR}$	Dat Terminal Ready (Terminal data siap)
Pin 34	$\overline{OUT1}$	User Output 1
Pin 35	MR	Master Riset

Pin 36	$\overleftarrow{CTS}$	Clear To Send
Pin 37	$\overleftarrow{DSR}$	Data Set Ready
Pin 38	$\overleftarrow{DCD}$	Data Carrier Detect
Pin 39	$\overleftarrow{RI}$	Ring Indikator (indicator dering)
Pin 40	VDD	+ 5 Volt

### 3.2.6. Tipe-tipe UART

8250 UART pertama pada seri ini. Tidak memiliki register *scratch*, versi 8250A merupakan versi perbaikan dari 8250 yang mampu bekerja dengan lebih cepat;

8250A UART ini lebih cepat dibandingkan dengan 8250 pada sisi bus. Lebih mirip secara perangkat lunak dibanding 16450;

8250B Sangat mirip dengan 8250;

16450 Digunakan pada komputer AT dengan kecepatan 38,4 Kbps, masih banyak digunakan hingga sekarang;

16550 Generasi pertama UART yang memiliki penyangga, dengan panjang 16-byte, namun tidak bekerja (produk gagal) sehingga digantikan dengan 16550A;

16550A UART yang banyak digunakan pada komunikasi kecepatan tinggi, misalnya 14,4 Kbps atau 28,8 Kbps;

16650 UART baru, memiliki penyangga FIFO 32-byte, karakter Xon/Xoff terprogram dan mendukung manajemen sumber daya;

16750 Diproduksi oleh Texas Instrument, memiliki FIFO 64-byte!

## 2.3. TINJAUAN PERANGKAT LUNAK

### 2.3.1. Alamat-alamat Kanal dan IRQ

Tabel 3.4. Alamat Port serial dan nomor IRQ-nya

Nama	Alamat (Heksa)	IRQ
<b>COM1</b>	<b>3F8</b>	<b>4</b>
<b>COM2</b>	<b>2F8</b>	<b>3</b>
<b>COM3</b>	<b>3E8</b>	<b>4</b>
<b>COM4</b>	<b>2E8</b>	<b>3</b>

Tabel 3.5. Informasi pada memori berkaitan dengan Port serial

Alamat Awal (Heksa)	Fungsi
<b>0000:0400</b>	<b>Alamat Dasar COM 1</b>
<b>0000:0402</b>	<b>Alamat Dasar COM 2</b>
<b>0000:0404</b>	<b>Alamat Dasar COM 3</b>
<b>0000:0406</b>	<b>Alamat Dasar COM 4</b>

UART menyediakan pencacah pembagi 16 (*divide by 16*) yang akan membagi frekuensi masukan dengan 16. Dengan demikian, jika frekuensi kristalnya 1,8432 MHz, maka frekuensi kerjanya adalah 115.200 Hz (mampu mengirim dan menerima dengan laju kecepatan 115.200 bps). Laju kecepatan ini tidak interupsi cocok untuk berbagai macam alat, dengan demikian digunakan Pembangkit Baud Rate terprogram (terdiri dari 2 register).

Misalnya diinginkan kecepatan 2.400 bps maka 115.200 harus dibagi 48 supaya menghasilkan nilai 2.400 Hz. Angka 48 dianggap sebagai "Pembagi" atau "*Divisor*" dan disimpan dalam dua register yang dikontrol oleh bit DLAB (lihat tabel II.6 untuk DLAB=1), karena menggunakan register (total) 16-bit maka angkanya antara 0 s/d 65.535. Gunakan tabel II.6 untuk frekuensi kristal 1,8432 MHz, DLL merupakan *Divisor Latch Low Byte* dan DLH merupakan *Divisor Latch High Byte*.

Tabel 3.6. Daftar Pembagi (*divisor*) untuk berbagai kecepatan pengiriman data.

Laju (bps)	Pembagian (Desimal)	DLH - Heksa	DLL – Heksa
50	2304	09h	00h
300	384	01h	80h
600	192	00h	C0h
2400	48	00h	30h
4800	24	00h	18h
9600	12	00h	0Ch
19200	6	00h	06h
38400	3	00h	03h
57600	2	00h	02h
115200	1	00h	01h

Tabel 3.7. Daftar REGISTER-REGISTER

Alamat Dasar	DLAB	Baca/Tulis	Simbol	Nama Register
+0	=0	Tulis	-	Transmitter Holding Buffer
	=0	Baca	-	Receiver Buffer
	=1	Baca/Tulis	-	Divisor Latch Low Byte
+1	=0	Baca/Tulis	IER	Interrupt Enable Register
	=1	Baca/Tulis	-	Divisor Latch High Byte
+2	-	Baca	IIR	Interrupt Indification Register
	-	Tulis	FCR	FIFO Control Register
+3	-	Baca/Tulis	LCR	Line Control Register
+4	-	Baca/Tulis	MCR	Modem Control Register
+5	-	Baca	LSR	Line Status Register
+6	-	Baca	MSR	Modem Status Register
+7	-	Baca/Tulis	-	Scratch register

**IER = INTERRUPT ENABLE REGISTER**

Bit	Keterangan
Bit 7	Tercadang
Bit 6	Tercadang
Bit 5	Aktifasi Mode Daya Rendah (16750)
Bit 4	Aktifasi Mode Tidur/Sleep (16750)
Bit 3	Aktifasi Interupsi Modem Status (MS), Prioritas -3
Bit 2	Aktifasi Interupsi Receiver Line Status (rls), Prioritas -0
Bit 1	Aktifasi Interupsi Transmitter Holding Register Empty (THRE), PRIORITAS-2
Bit 0	Aktifasi Interupsi Received Data Available (RDA), prioritas-1

Register IER mungkin satu-satunya register yang mudah dipahami dan simpel. Interupsi RDA terjadi jika register penerima (FIFO) berisi data yang akan dibaca CPU. Interupsi THRE terjadi saat penyangga pengirim kosong, sedangkan interupsi RLS terjadi jika ada perubahan pada status jalur penerimaan (*receiver line status*), demikian juga untuk interupsi MS akan diaktifkan jika ada perubahan pada status modem.

### IIR = INTERRUPT IDENTIFICATION REGISTER

Bit	Keterangan		
Bit 6 dan 7	Bit 6	Bit 7	
	0	0	Tidak Pakai FIFO (8250/16450)
	0	1	FIFO aktif tapi Tak-dapat digunakan (16550)
	1	1	FIFO aktif
Bit 5	FIFO 64 Byte aktif (hanya untuk 16750)		
Bit 4	Tercadang		
Bit 3	0	Tercadang pada 8250,16450	
	1	Pada 16550 untuk Time-out Interrupt Pending	
Bit 1 dan 2	Bit 2	Bit 1	
	0	0	Interupsi Modem Status
	0	1	Interupsi Transmitter Holding Register Empty
	1	0	Interupsi Received Data Available
	1	1	Interupsi Receiver Line Status
Bit 0	0	Penantian Interupsi ( <i>Interrupt Pending</i> )	
	1	Tidak Ada Penantian Interrupsi ( <i>No Interrupt Pending</i> )	

Register IIR merupakan register hanya-baca (*read only*). Bit 6 dan 7 menginformasikan jenis penyangga FIFO, jika keduanya berisi 0, maka UART yang bersangkutan tidak memiliki penyangga FIFO. Ini adalah satu-satunya basil, jika menggunakan UART seri 8250 atau 16450. Jika Bit-7=1 dan bit-6=0, maka UART memiliki penyangga FIFO tetapi tdak dapat digunakan dan ini sebagai 'bug' pada seri 16550. Sedangkan jika kedua bit tersebut berisi 1, maka penyangga FIFO berhasil diaktifkan dan berfungsi penuh. Bit-0 digunakan untuk memberitahukan ada atau tidaknya interupsi, jika ada interupsi statusnya ditunjukkan pada bit 1 dan 2. Interupsi-interupsi ini bekerja berdasar prioritas, perhatikan tabel IER

### FCR = FIFO CONTROL REGISTER

Bit	Keterangan		
Bit 6 dan 7	Bit 7	Bit 6	Tingkat Pemicu Interupsi
	0	0	1 Byte
	0	1	4Byte
	1	0	8Byte
	1	1	14Byte
Bit 5	Aktifasi FIFO 64 Byte (Hanya Seri16750)		
Bit 4	Tercadang		
Bit 3	Pemilihan Mode DMA.Mengubah Status Pin RXRDY & TXRDY dari mode 0 ke mode 1		
Bit 2	Kosongkan FIFO Pengirim		
Bit 1	Kosongkan FIFO Penerima		
Bit 0	Aktifasi fifo		

Register FIFO ini merupakan register tulis-saja (*write only*). Register ini digunakan untuk mengontrol penyangga FIFO yang dapat ditemukan pada chip UART 16550 dan yang lebih tinggi. Bit-0 digunakan untuk mengaktifkan FIFO, jika Anda menon-aktifkan FIFO tersebut maka data yang tersimpan dalam FIFO juga akan hilang.

Bit-1 dan 2 mengontrol penghapusan FIFO pengirim atau pe-nerima. Dengan men-set 1, isi FIFO akan dibersihkan tanpa mem-pengaruhi register geser. Dua bit ini juga bersifat *self-resetting*, artinya Anda tidak perlu membuat bit ini kembali 0, karena akan dilakukan secara otomatis jika proses selesai dikcrjakan. Bit 3 digunakan untuk mengaktifkan pemilihan mode DMA yang dapat ditemukan pada UART 16550 dan yang lebih tinggi. Sedangkan bit 6 dan 7 digunakan untuk menset tingkat pemicuan pada FIFO penerima.

**LCR = LINE CONTROL REGISTER**

Bit	Keterangan			
Bit 7	1	Divisor Latch Access Bit (DLAB)		
	0	Akseske Penyangga Penerima (RX), Pengirim (TX) & register Aktivasi Interupsi (IER)		
Bit 6	Aktifasi Set Break			
Bit 3,4 dan 5	Bit 5	Bit 4	Bit 3	Pemilihan Paritas
	X	X	0	Tidak Pakai Paritas
	0	0	1	Paritas Ganjil
	0	1	1	Paritas Genap
	1	0	1	Paritas tnggi (Sticky), Paritas interupsilu 1
	1	1	1	Paritas tnggi (Sticky), Paritas interupsilu 0
Bit 2	Length of Stop Bit			
	0	Sati Bit Stop		
	1	2 bit stop untuk panjang word 6, 7 atau 8 bit atau 1, 5 bit stop untuk panjang Word 5 bit		
Bit 0 dan 1	Bit 1	Bit 0		Panjang Word
	0	0		5 Bit
	0	1		6 Bit
	1	0		7 Bit
	1	1		8 Bit

Register ini digunakan untuk men-set parameter-parameter dasar. Bit-7 sebagai *Divisor Latch Buffer* atau DLAB. Bit-6 digunakan untuk mengaktifkan *break*, jika diaktifkan, jalur TxD akan berkondisi "Penspasian" (*Spacing*) yang mengakibatkan pemutusan (*break*) hubungan dengan UART penerima. Sedangkan bit 3, 4 dan 5 digunakan untuk pemilihan paritas:

- Bit-3 digunakan untuk menentukan dipakai atau tidaknya bit paritas (jika 0 tidak ada bit paritas);
- Bit-5 menentukan penggunaan paritas *sticky*, yaitu jika bit-5 diaktifkan (=1) (menggunakan paritas *sticky*) dan jika bit-4=0 maka bit paritas akan interupsi dikirim dalam logika 1 dan atau dikirim 0 jika bit-4=1. Jika bit-5 ini tidak diaktifkan berarti menggunakan mode paritas normal.

Pemeriksaan kesalahan menggunakan bit paritas memang baik tetapi tidak sempurna: jika ada salah satu bit yang terbalik (dari 1 menjadi 0 misalnya) maka tetap akan terdeteksi terjadinya kesalahan paritas (*parity error*) tetapi jika ada dua bit yang berlawanan (1 dan 0) yang saling bertukar-tempat, maka tidak dapat dideteksi adanya kesalahan! Perhatikan ilustrasi pada gambar II.4. Sehingga banyak perangkat lunak yang kemudian menggunakan algoritma CRC-32.

Dikirim sbb:				Paritas Genap	
0	1	0	1	0	Awalnya
Diterima sbb:					
1 error	0 error	0	1	0	Ada error tetapi paritas tetap benar

Gambar 3.4. Logika 1 dan 0 yang bertukar tempat tidak menyebabkan error.

Bit-2 digunakan untuk menset panjang atau jumlah bit stop (tergantung dari panjang word) sedangkan bit 0 dan 1 digunakan untuk menentukan panjang word apakah 5, 6, 7 atau 8 byte.

#### MCR = MODEM CONTROL REGISTER

Bit	Keterangan
Bit 7	Tercadang
Bit 6	Tercadang
Bit 5	Aktifasi Auto flow Control (hanya seri 16750)
Bit 4	Mode Loopback
Bit 3	Keluaran AUX 2
Bit 2	Keluaran AUX 1
Bit 1	Force Request to Send
Bit 0	Force Data Terminal Ready

Register ini bersifat Baca/Tulis. Bit-4 digunakan untuk meng-aktifkan mode *loopback*. Dalam mode ini keluaran serial (pengiriman) dalam kondisi ditandai (*marked*). Masukan serial penerima diputus-kan. Keluaran pengirim dikirim-kembali (*loopbacked*) ke masukar penerima.

Sinyal-sinyal DSR, CTS, RI dan BCD diputuskan. DTR, RTS, OUT1 dan OUT2 dihubungkan ke masukan kontrol modem. Pin-pin

keluaran kontrol modem kemudian dinon-aktifkan. Dalam mode ini data apa saja yang ditempatkan pada register pengirim untuk keluaran akan diterima oleh rangkaian penerima dalam satu chip yang sama sehingga data tersebut akan siap dibaca melalui penyangga penerima.

Keluaran AUX 2 dapat dihubungkan dengan rangkaian eks-ternal yang digunakan untuk mengontrol proses interupsi UAKT dan CPU. Keluaran AUX 1 normalnya tak terhubung, namun untuk beberapa kartu digunakan untuk *men-swicth* antara penggunaan kristal 1,8432 MHz dan 4 MHz yang dijumpai pada peralatan MIDI (*Music Instrument Device Interface*).

#### LSR = LINE STATUS REGISTER

Bit	Keterangan
Bit 7	Ada Kesalahan Dalam FIFO Penerima
Bit 6	Data Holding Register kosong
Bit 5	Transmitter Holding Register kosong
Bit 4	Break Interrupt
Bit 3	Framing Error
Bit 2	Parity Error
Bit 1	Overrun Error
Bit 0	Data Siap (dibaca)

Register ini bersifat hanya-baca. Jika bit-6=1, baik register *Transmitter Holding* (THR) dan register geser kosong. Register pe-nahan (*holding*) UART menyimpan data byte berikutnya yang akan dikirim dengan cara paralel. Register geser digunakan untuk meng-ubah data byte, yang paralel tersebut, menjadi serial sehingga dapat dikirim melalui jalur serial. Sedangkan jika bit-5=1 hanya THR yang kosong. Lantas apa perbedaan keduanya? Jika bit-6=1, THR dan register geser kosong, maka tidak ada proses konversi serial, artinya tidak ada aktivitas sama sekali pada jalur pengiriman data. Sedangkan jika bit-5=1, hanya THR saja yang kosong, jadi data byte berikutnya dapat dikirim ke kanal data dan register geser masih tetap be-kerja.

Interupsi *break* (bit-4) muncul saat jalur penerimaan data ditahan pada ttondisi logika '0\* atau spasi interupsima lebih dari waktu yang dibutuhkan untuk mengirimkan satu word penuh. Beberapa kesalahan lain bisa muncul sesuai kondisi berikut:

- *Framing error* (bit-3) terjadi jika saat bit terakhir yang terkirim bukan suatu bit stop hal ini bisa disebabkan karena kesalahan pewaktuan (*timing error*);



- *Overrun error* (bit-1) terjadi saat program Anda (komputer) tidak cukup cepat untuk membaca dari kanal serial;

Sedangkan *Data ready* (bit-0) akan sama dengan 1 jika ada sebuah byte yang diterima oleh UART dan penyangga penerima siap untuk dibaca.

**MSR = MODEM STATUS REGISTER**

Bit	Keterangan
Bit 7	Karrier Detect (deteksi Carrier)
Bit 6	Ring Indicator (indicator dering)
Bit 5	Data Set Ready (Set data siap)
Bit 4	Clear To Send (siap untuk mengirim)
Bit 3	Delta Data Carrier Detect
Bit 2	Trailing Edge Ring Indicator
Bit 1	Delta Data Set Ready
Bit 0	Delta Clear to Send

Bit-0 pada register ini menunjukkan *delta clear to send*, delta artinya adanya perubahan dalam, dengan demikian artinya ada perubahan dalam jalur *clear to send*, sejak pembacaan terakhir dari register ini.

**SCRATCH REGISTER**

Register ini tidak digunakan untuk komunikasi namun sebagai tempat untuk meninggalkan sebuah byte data. Kenyataannya digunakan untuk menentukan apakah UART-nya seri 8250/8250B atau 8250A/16450 dan saat ini tidak digunakan karena 8250/8250B tidak pernah dibuat untuk AT serta dapat mengacaukan kecepatan bus!