

MODE TRANSMISI DATA LAPISAN FISIK

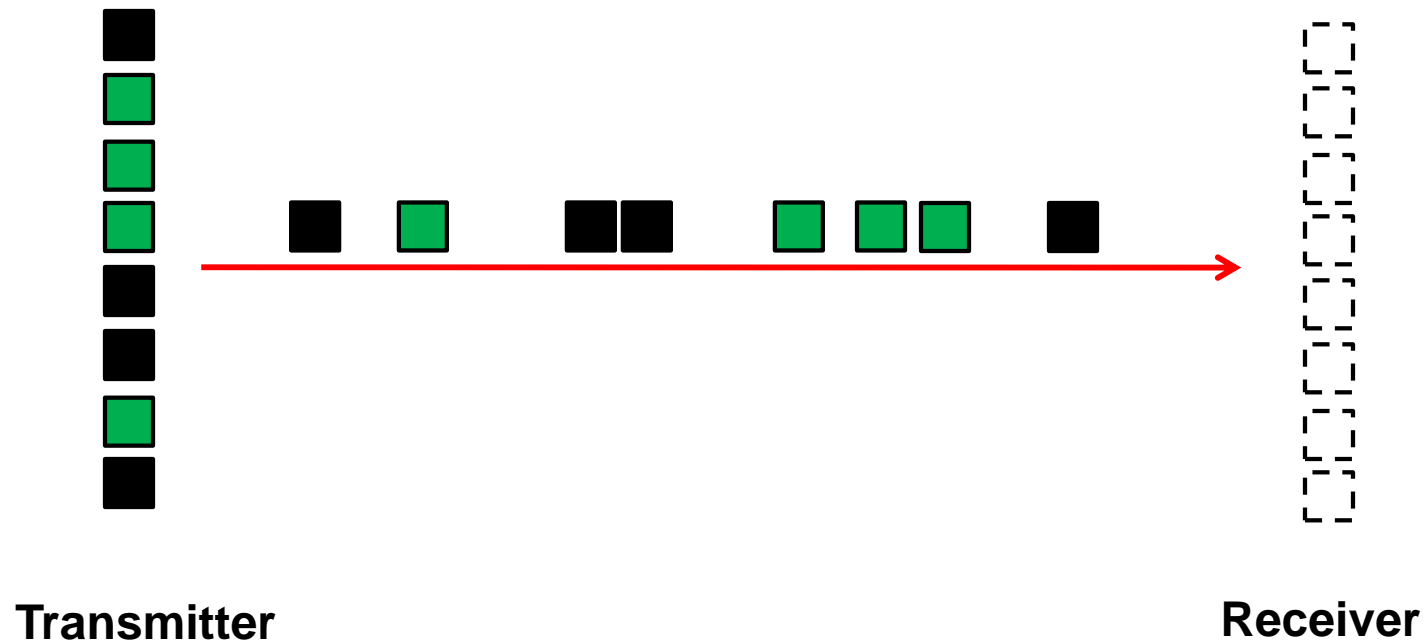


Mode Transmisi Data

- ☐ Mode Transmisi **Serial**
- ☐ Mode Transmisi **Paralel**

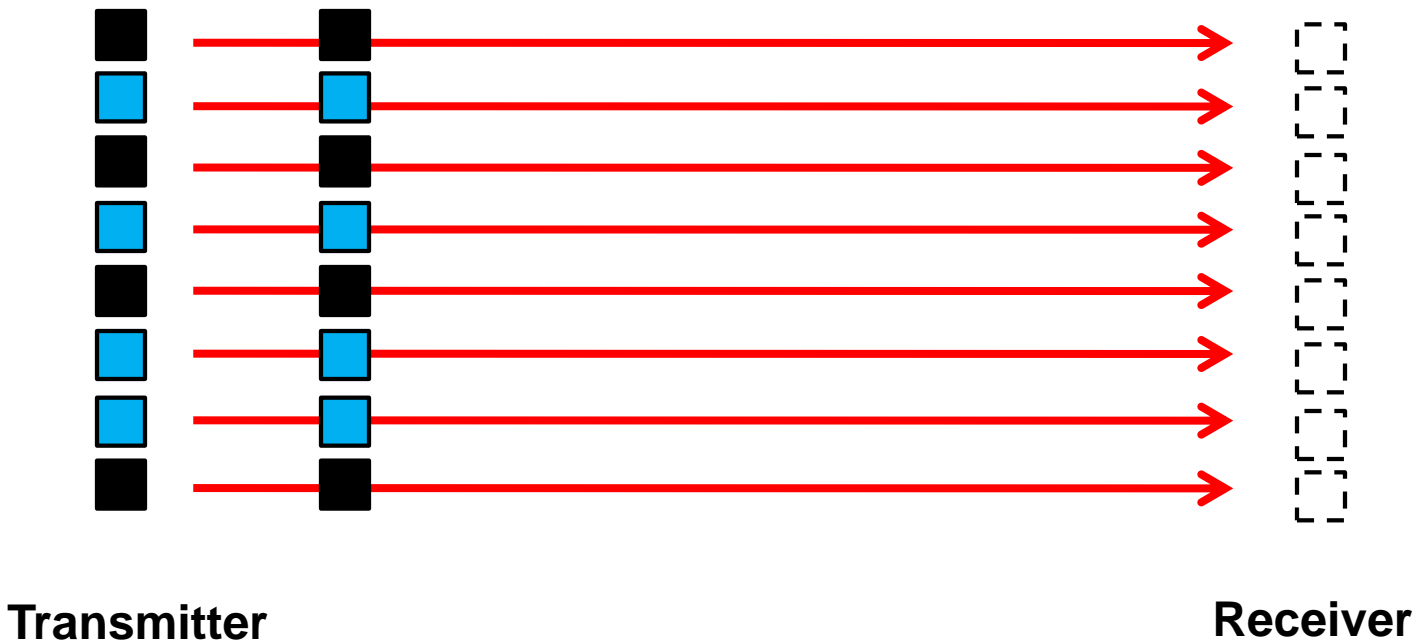
Mode Transmisi Serial

Proses pengiriman data pada *mode transmisi serial* adalah dilakukan *bit per bit*



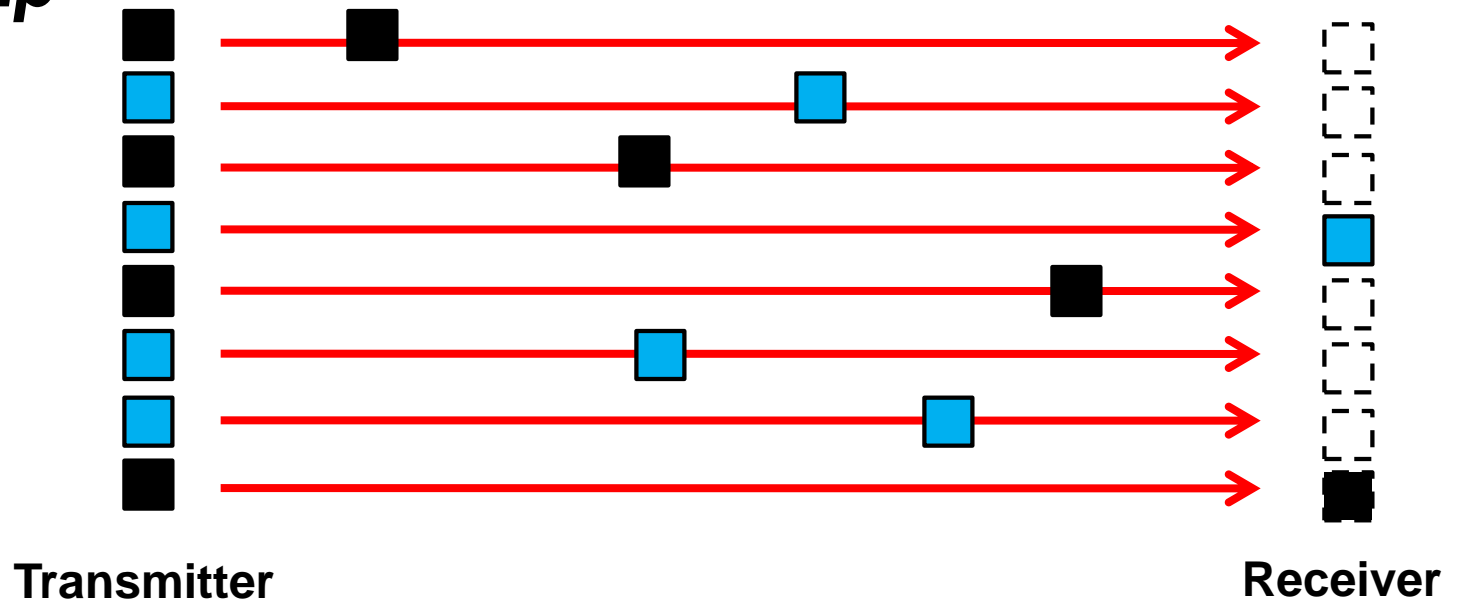
Mode Transmisi Paralel

Proses pengiriman data pada **mode transmisi paralel** adalah dilakukan *byte per Byte*



Mode Transmisi Paralel

SKEW EFFECT adalah Gejala pengiriman data *secara paralel* yang mana data dikirim secara bersamaan tapi *tiba tidak bersamaan* atau *tidak lengkap*



Metode Transmisi Data

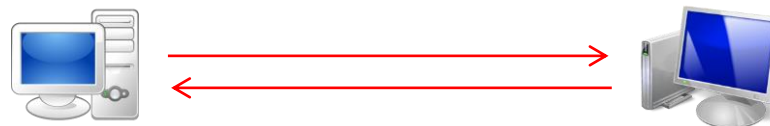
☐ Metode Simplex



☐ Metode Half Duplex



☐ Metode Full Duplex



Sinkronisasi

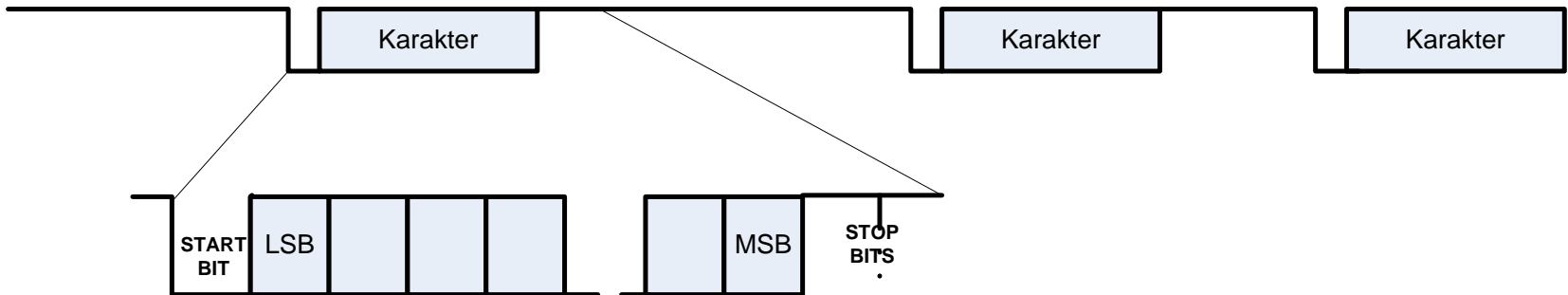
Pada transmisi data di *lapis fisik*, selain *modulasi* diperlukan kemampuan untuk *sinkronisasi*, yaitu *teknik mendapatkan bit* di *suatu sinyal* yang melibatkan masalah *waktu pengambilan sampel* dari *sinyal*, *format suatu karakter* dan *format paket*.

Jenis Sinkronisasi

Terdapat 3 jenis *teknik sinkronisasi* data yaitu :

- ☐ Asynchronous
- ☐ Synchronous
- ☐ Isochronous

Asynchronous



Data dikirim karakter demi karakter

Diawali start bit dan diakhiri stop bit

Jarak antar karakter bebas (tidak diatur)

Panjang karakter bisa : 5,6,7,8 bit

Panjang stop bit bisa : 1, 1.5 dan 2 bit

Bisa dibubuhkan pariti genap atau ganjil

Contoh : standar RS-232 (port serial pada komputer)

Umum digunakan untuk kondisi saluran berkualitas sedang

Asynchronous

- Misal format pengiriman [start bit][bit data][bit pariti][stop bit]
- 0101010101

Synchronous



Diawali dan diakhiri dengan karakter flag

Panjang karakter tetap dan berurutan

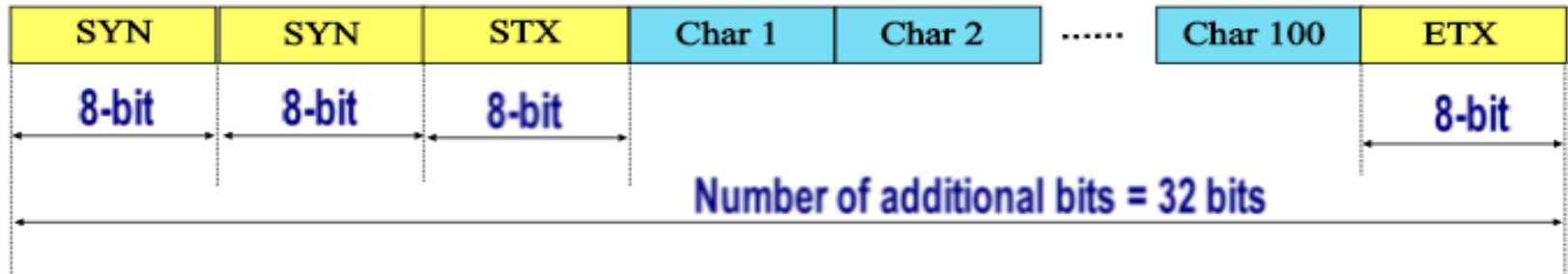
Tanpa pariti

Menuntut kualitas saluran sangat baik (error minimal, tidak terjadi pergeseran waktu dll)

Sangat efisien (rasio payload tinggi)

Contoh : X.25

Synchronous



Synchronous

Sistem Kode ASCII

Bit Positions				7	0	0	0	0	1	1	1	1
				6	0	0	1	1	0	0	1	1
				5	0	1	0	1	0	1	0	1
4	3	2	1									
0	0	0	0	NUL	DLE	SP	0	@	P	\	p	
0	0	0	1	SOH	DC1	!	1	A	Q	a	q	
0	0	1	0	STX	DC2	"	2	B	R	b	r	
0	0	1	1	ETX	DC3	#	3	C	S	c	s	
0	1	0	0	EOT	DC4	\$	4	D	T	d	t	
0	1	0	1	ENQ	NAK	%	5	E	U	e	u	
0	1	1	0	ACK	SYN	&	6	F	V	f	v	
0	1	1	1	BEL	ETB	'	7	G	W	g	w	
1	0	0	0	BS	CAN	(8	H	X	h	x	
1	0	0	1	HT	EM)	9	I	Y	i	y	
1	0	1	0	LF	SUB	*	:	J	Z	j	z	
1	0	1	1	VT	ESC	+	;	K	[k	{	
1	1	0	0	FF	FS	,	<	L	\	l		
1	1	0	1	CR	GS	-	=	M]	m	}	
1	1	1	0	SO	RS	.	>	N	^	n	~	
1	1	1	1	SI	US	/	?	O	_	o	DEL	

Isochronous



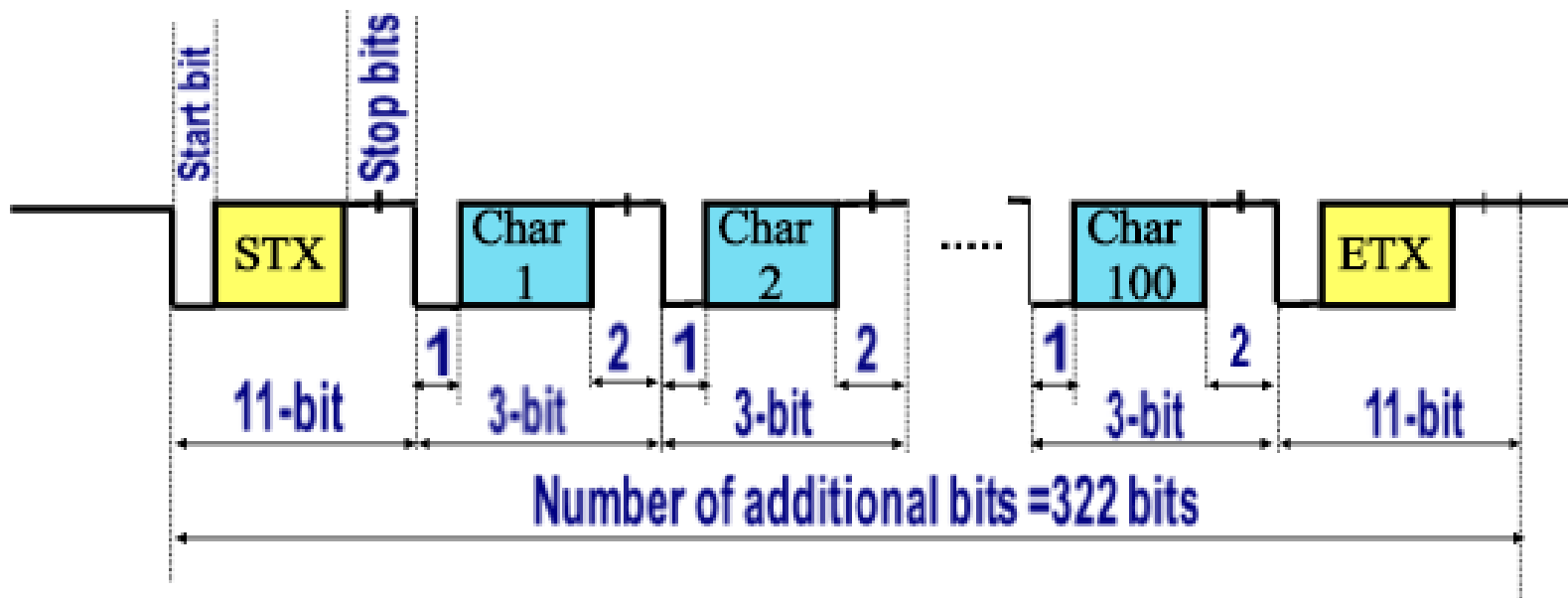
Gabungan dari asinkron dan sinkron

Diawali dengan karakter flag

Setiap karakter data diawali dengan start bit dan diakhiri dengan stop bit

Sudah jarang digunakan dikarenakan paling tidak efisien

Isochronous



Teknik Transmisi di Lapis Fisik

- ☐ Jika masalah pengkodean saluran sudah bisa dianggap selesai, maka urusan selanjutnya adalah bagaimana penerima ***mendapatkan data*** yang ditujukan kepadanya dari sinyal yang dikirim
- ☐ Pada dasarnya lapis fisik harus mampu memisahkan ***bit-demi-bit*** yang terkodekan di ***sinyal yang diterima***
- ☐ Proses ini disebut ***Sinkronisasi Bit***

Proses Sinkronisasi



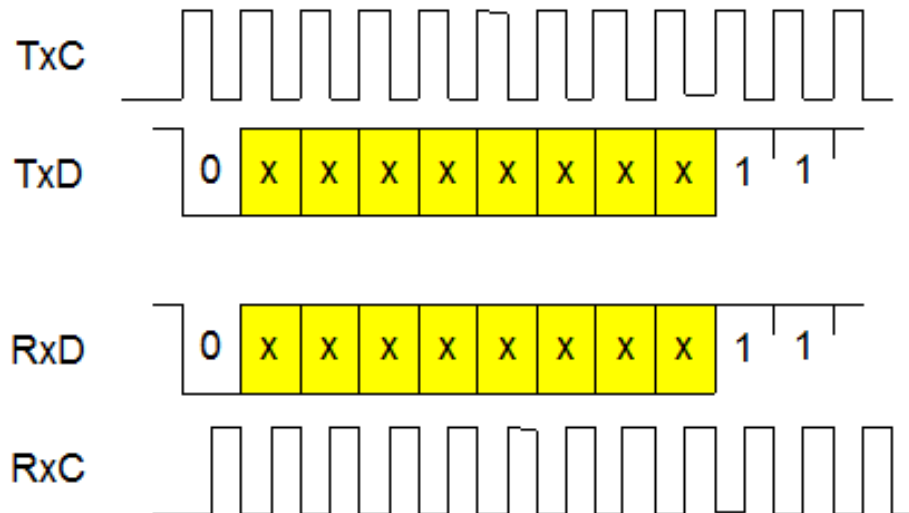
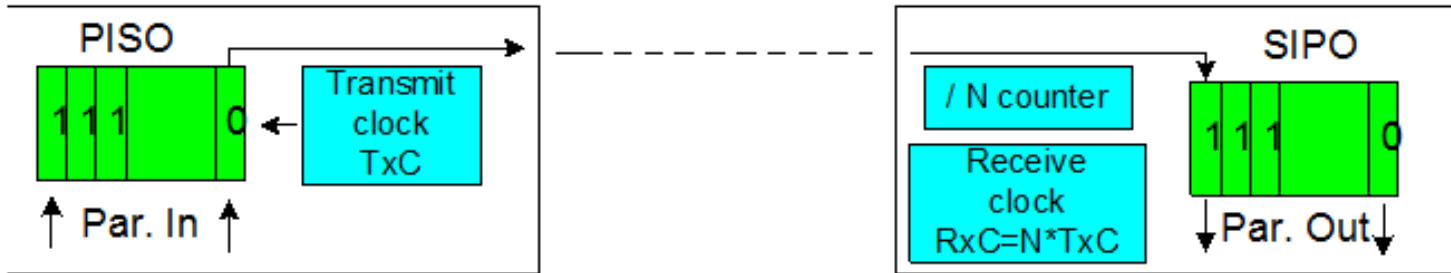
Sinkronisasi Bit

- ***Semakin banyak jumlah sample, maka akan semakin akurat prediksi bit yang didapat apakah bit '0' atau bit '1' dengan konsep sederhana 'mayoritas menentukan hasil', jika mayoritas bit di suatu perioda sampling (sepanjang slot pada laju bit pengirim) cenderung ke bit tertentu, maka dianggap bit tersebut yang diterima.***

Sinkronisasi Bit

- Untuk mendapatkan bit yang terdapat pada sinyal yang ***berubah-ubah*** dengan ***cepat***, dilakukan ***teknik sampling sinyal*** dengan jumlah sample ***beberapa kali dari laju data***.
- Pada ***sistem RS-232***, umum dilakukan sampling sebesar ***8x***, ***16x*** atau ***64x*** dari laju data pengirim

Sinkronisasi Bit



Sinkronisasi Karakter

- Setelah mendapatkan **bit-bit informasi**, maka tugas selanjutnya adalah mendapatkan **set bit** yang membentuk karakternya.
- Tugas ini sangat penting dikarenakan salah memilih **posisi bit** dalam **karakter** akan memberikan karakter lain yang **berbeda** artinya sama sekali

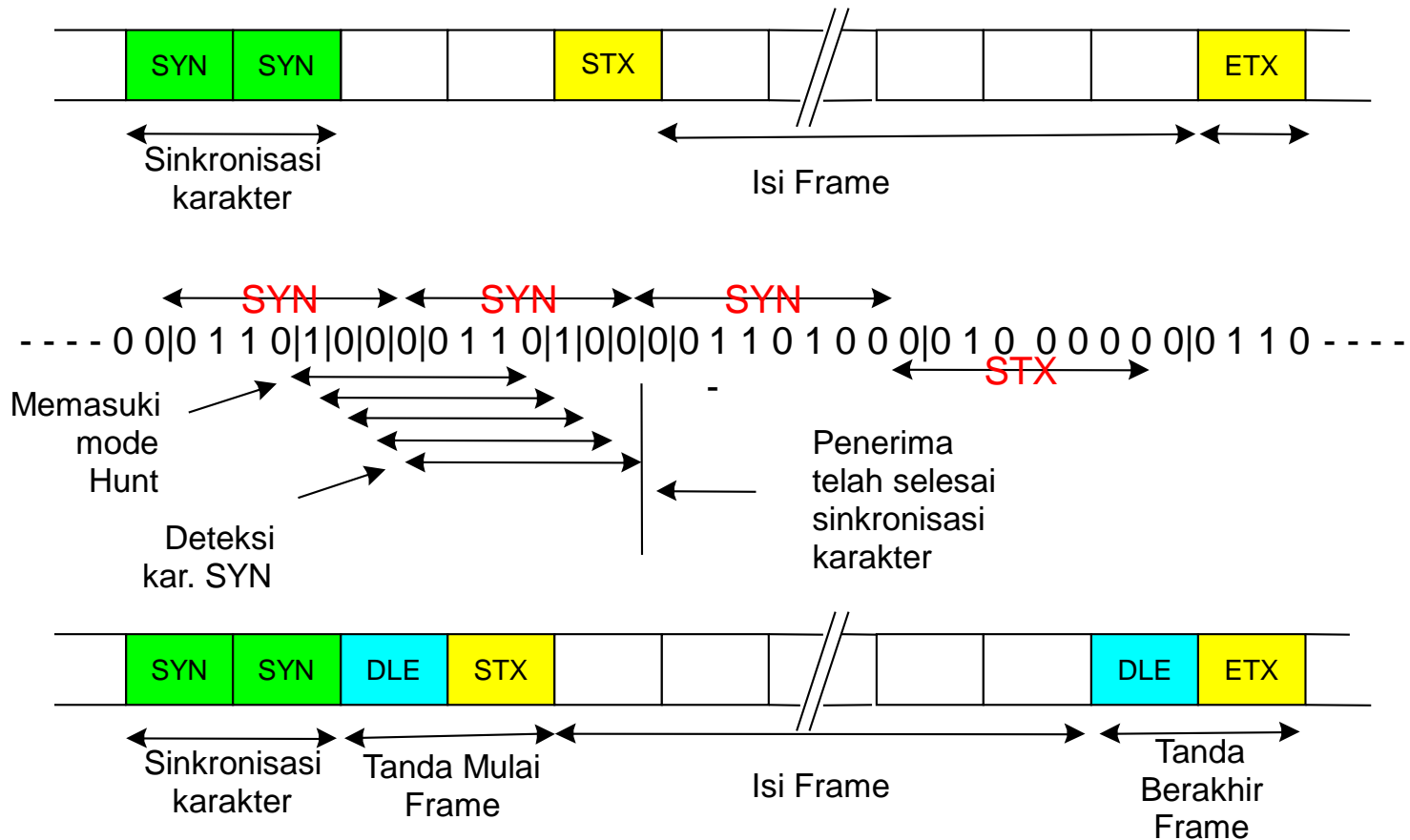
Sinkronisasi Karakter

□ Contoh : diterima **0011000101**

Jika dibaca sebagai MSB mulai dari bit paling kiri, maka akan didapatkan karakter **ASCII 31h (angka 1) → 00110001**

Jika dibaca sebagai MSB mulai dari bit kedua dari kiri, maka akan didapatkan karakter **ASCII 62h (huruf b) → 01100010**

Sinkronisasi Karakter



Sinkronisasi Karakter

- Digunakan karakter **SYN [0010110]** sebagai penanda mulainya bit dari suatu karakter
- Cara kerjanya relatif sederhana:
Penerima akan mencari (*hunting*) karakter **SYN [0010110]** dalam urutan bit yang diterimanya akan dicocokkan 8 bit pertama yang dimulai dari bit '0' (kondisi ini disebut memasuki *mode hunting*)

Sinkronisasi Karakter

- ☐ Jika cocok, maka **8 bit** tersebut ditetapkan sebagai karakter pertama
- ☐ Jika tidak cocok, maka akan mencari bit '0' berikutnya untuk selanjutnya melakukan *hunting* lagi
- ☐ Disediakan **2 atau 3 karakter SYN [0010110]** untuk berjaga-jaga jika terlewat menerima karakter **SYN [0010110]** pertama

Contoh

0101000101100010110001011000000101001111
100101110001010000011

	SYN	SYN	SYN	STX	O
01010	00101100	0101100	010110	0000010	1001111
1001011	1000101	0000011			
K	E	ETX			

Table ASCII

Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char
0	0x00	000	00000000	NUL	32	0x20	040	01000000	space	64	0x40	100	10000000	@	96	0x60	140	11000000	`
1	0x01	001	00000001	SOH	33	0x21	041	01000001	!	65	0x41	101	10000001	A	97	0x61	141	11000001	a
2	0x02	002	00000010	STX	34	0x22	042	01000010	"	66	0x42	102	10000010	B	98	0x62	142	11000010	b
3	0x03	003	00000011	ETX	35	0x23	043	01000011	#	67	0x43	103	10000011	C	99	0x63	143	11000011	c
4	0x04	004	00000100	EOT	36	0x24	044	01000100	\$	68	0x44	104	10000100	D	100	0x64	144	11000100	d
5	0x05	005	00000101	ENQ	37	0x25	045	01000101	%	69	0x45	105	10000101	E	101	0x65	145	11000101	e
6	0x06	006	00000110	ACK	38	0x26	046	01000110	&	70	0x46	106	10000110	F	102	0x66	146	11000110	f
7	0x07	007	00000111	BEL	39	0x27	047	01000111	'	71	0x47	107	10000111	G	103	0x67	147	11000111	g
8	0x08	010	00010000	BS	40	0x28	050	01010000	(72	0x48	110	10010000	H	104	0x68	150	11010000	h
9	0x09	011	00010001	TAB	41	0x29	051	01010001)	73	0x49	111	10010001	I	105	0x69	151	11010001	i
10	0x0A	012	00010010	LF	42	0x2A	052	01010010	*	74	0x4A	112	10010010	J	106	0x6A	152	11010010	j
11	0x0B	013	00010011	VT	43	0x2B	053	01010011	+	75	0x4B	113	10010011	K	107	0x6B	153	11010011	k
12	0x0C	014	00011000	FF	44	0x2C	054	01011000	,	76	0x4C	114	10011000	L	108	0x6C	154	11011000	l
13	0x0D	015	00011001	CR	45	0x2D	055	01011001	-	77	0x4D	115	10011001	M	109	0x6D	155	11011001	m
14	0x0E	016	00011010	SO	46	0x2E	056	01011010	.	78	0x4E	116	10011010	N	110	0x6E	156	11011010	n
15	0x0F	017	00011011	SI	47	0x2F	057	01011011	/	79	0x4F	117	10011011	O	111	0x6F	157	11011011	o
16	0x10	020	00100000	DLE	48	0x30	060	01100000	0	80	0x50	120	10100000	P	112	0x70	160	11100000	p
17	0x11	021	00100001	DC1	49	0x31	061	01100001	1	81	0x51	121	10100001	Q	113	0x71	161	11100001	q
18	0x12	022	00100010	DC2	50	0x32	062	01100010	2	82	0x52	122	10100010	R	114	0x72	162	11100010	r
19	0x13	023	00100011	DC3	51	0x33	063	01100011	3	83	0x53	123	10100011	S	115	0x73	163	11100011	s
20	0x14	024	00101000	DC4	52	0x34	064	01101000	4	84	0x54	124	10101000	T	116	0x74	164	11101000	t
21	0x15	025	00101001	NAK	53	0x35	065	01101001	5	85	0x55	125	10101001	U	117	0x75	165	11101001	u
22	0x16	026	00101010	SYN	54	0x36	066	01101010	6	86	0x56	126	10101010	V	118	0x76	166	11101010	v
23	0x17	027	00101011	ETB	55	0x37	067	01101011	7	87	0x57	127	10101011	W	119	0x77	167	11101011	w
24	0x18	030	00110000	CAN	56	0x38	070	01110000	8	88	0x58	130	10110000	X	120	0x78	170	11110000	x
25	0x19	031	00110001	EM	57	0x39	071	01110001	9	89	0x59	131	10110001	Y	121	0x79	171	11110001	y
26	0x1A	032	00110010	SUB	58	0x3A	072	01110010	:	90	0x5A	132	10110010	Z	122	0x7A	172	11110010	z
27	0x1B	033	00110011	ESC	59	0x3B	073	01110011	;	91	0x5B	133	10110011	[123	0x7B	173	11110011	{
28	0x1C	034	00111000	FS	60	0x3C	074	01111000	<	92	0x5C	134	10111000	\	124	0x7C	174	11111000	
29	0x1D	035	00111001	GS	61	0x3D	075	01111001	=	93	0x5D	135	10111001]	125	0x7D	175	11111001	}
30	0x1E	036	00111010	RS	62	0x3E	076	01111010	>	94	0x5E	136	10111010	^	126	0x7E	176	11111010	~
31	0x1F	037	00111011	US	63	0x3F	077	01111011	?	95	0x5F	137	10111011	_	127	0x7F	177	11111011	DEL

Sinkronisasi Frame

- Setelah mendapatkan karakter-karakter didapat masalah baru, yaitu ***karakter mana yang merupakan informasi (frame data)*** dan ***mana yang merupakan karakter random yang ditambahkan sistem transmisi*** (pada komunikasi sinkron) atau ***noise yang kebetulan memenuhi syarat untuk dibaca sebagai suatu karakter*** (pada komunikasi asinkron)

Sinkronisasi Frame

- ☐ Pada prinsipnya, suatu deretan karakter yang mengandung informasi diapit oleh karakter-karakter khusus sebagai penanda, karakter tersebut adalah **STX [0000010]** sebagai tanda awal frame dan **ETX [0000011]** sebagai tanda akhir frame
- ☐ Mekanisme ini disebut ***Sinkronisasi Frame***

Sinkronisasi Frame

Terdapat dua jenis sinkronisasi frame

- 1. Untuk yang berupa karakter (teks),** mengandung informasi yang hanya terdiri dari karakter-karakter huruf, angka dan karakter lain (umumnya merupakan karakter ASCII 00h s/d 7Fh) → cukup digunakan karakter **STX [0000010]** dan **ETX [0000011]**

Sinkronisasi Frame

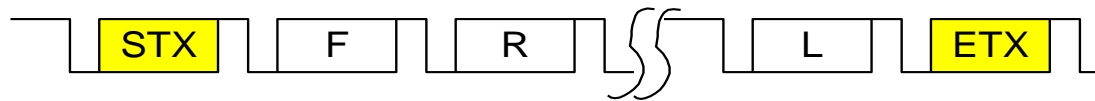
- 2. Untuk data biner**, mengandung informasi yang menggunakan semua kombinasi ASCII (**data gambar, suara dan data-data lain yang dikodekan dari 00h s/d FFh**) → menggunakan karakter **DLE [0010000]** yaitu **STX** dan **DLE ETX**

contoh

- Kirimkan dgn format sinkronisasi frame
 1. ada apa → **STX** a d a **space** a p a **ETX**
 2. ada apa → **DLE STX** a d a **ETX space** a p a **DLE ETX**

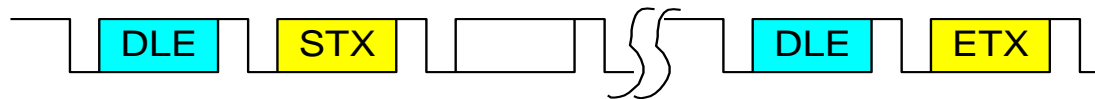
Sinkronisasi Frame

PRINTABLE CHARACTER



STX
F
R
L
ETX

BINARY DATA



DLE
STX
DLE
DLE
DLE
ETX