

KATA PENGANTAR

Alhamdulillah, segala puji bagi Allah yang telah memberikan segala bimbingan-Nya kepada penulis untuk menyelesaikan buku praktikum Komunikasi Data ini.

Buku ini dipergunakan sebagai modul ajar praktikum Komunikasi Data program D3/D4 di Politeknik Elektronika Negeri Surabaya. Sasaran dari praktikum Komunikasi Data ini adalah memberikan pengetahuan kepada mahasiswa tentang Sistem Komunikasi Data antar dua atau lebih PC, selain itu juga mempelajari tentang Model referensi OSI Layer dan protokol TCP/IP. Komunikasi Data merupakan dasar dari Jaringan Komputer. Buku praktikum Komunikasi Data ini dapat digunakan sebagai panduan bagi mahasiswa saat melaksanakan praktikum tersebut.

Penulis menyadari bahwa buku ini jauh dari sempurna, oleh karena itu penulis akan memperbaikinya secara berkala. Saran dan kritik untuk perbaikan buku ini sangat kami harapkan.

Akhir kata, semoga buku ini bermanfaat bagi mahasiswa dalam mempelajari mata kuliah Komunikasi Data. Amin.

Surabaya, 24 Januari 2007

Hormat kami,

Penulis

DAFTAR ISI

Kata Pengantar.....	1
Daftar Isi.....	2
<u>Praktikum 1 : Perakitan Kabel Null Modem DB9, DB25, RJ45</u>	<u>3</u>
<u>Praktikum 2 : HyperTerminal dan Minicom</u>	<u>12</u>
<u>Praktikum 3 : Pengukuran Komunikasi Serial</u>	<u>22</u>
<u>Praktikum 4 : Socket Programming User Datagram Protocol (UDP).....</u>	<u>27</u>
<u>Praktikum 5 : Socket ProgrammingTransport Control Protocol (TCP).....</u>	<u>35</u>
<u>Praktikum 6 : Wireless – LAN (Indoor).....</u>	<u>43</u>
<u>Praktikum 7 : Wireless – LAN (Outdoor).....</u>	<u>49</u>

PERCOBAAN I

PERAKITAN KABEL NULL MODEM DB9,DB25,RJ45

TUJUAN

- 1 Mahasiswa mampu memahami kegunaan kabel/konektor DB9, DB25, RJ45.
- 2 Mahasiswa mampu memahami fungsi dari masing-masing pin dari konektor DB9, DB25 dan RJ45.
- 3 Mahasiswa mampu merakit kabel dengan menggunakan DB9, DB25, RJ45.

DASAR TEORI

RS232/EIA232 - Null-Modem

RS232 adalah standard komunikasi serial antar peripheral-peripheral. Contoh paling sering kita pakai adalah antara komputer dengan modem, atau komputer dengan komputer.

Standar ini menggunakan beberapa piranti dalam implementasinya. Paling umum yang dipakai adalah plug DB9 atau DB25.

Untuk rs232 dengan DB9, biasanya dipakai untuk serial port pada komputer pribadi. Dipakai untuk port mouse dan modem. Fungsi dari masing-masing pin ditunjukkan pada gambar 1.

RS232 Pin Assignments DB9 PC Signal Set		
Pin Number	Signal Name	Abbreviation
1	Carrier Detect	CD
2	Receive Data	RxD
3	Transmit Data	TxD
4	Data Terminal Ready	DTR
5	System Ground	SG
6	Data Set Ready	DSR
7	Request To Send	RTS
8	Clear To Send	CTS
9	Ring Indicator	RI

Gambar 1. Fungsi pin-pin DB9 standar rs232.

Untuk melakukan komunikasi antar komputer dengan menggunakan standar rs232, bisa kita gunakan dua cara:

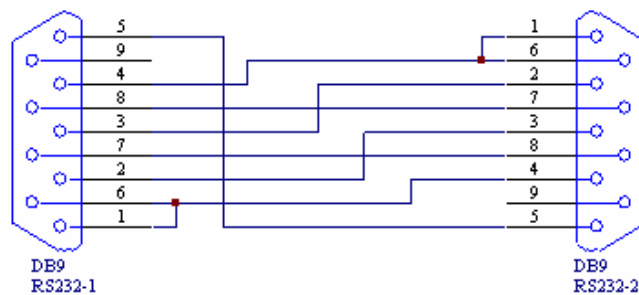
1. dua modem yang dipasang pada serial port, atau

2. dengan kabel konektor serial null-modem.

RS232 Null Modem Configuration			
Side 1		Side 2	
Signal Name	Pin Number	Pin Number	Signal Name
RxD	2	3	TxD
TxD	3	2	RxD
DTR	4	6+1	DSR+CD
SG	5	5	SG
DSR+CD	6+1	4	DTR
RTS	7	8	CTS
CTS	8	7	RTS

Gambar 2. Koneksi pin rs232 null-modem.

Untuk mengetahui nomor-nomor pin ini bisa dilihat pada plugnya langsung.



Gambar 3. Skema pin rs232 null-modem untuk komunikasi antar komputer.

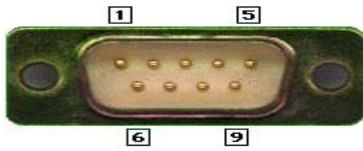
Suatu komunikasi data antar PC dapat dilakukan dengan menggunakan berbagai macam interface I/O. Pada PC terdapat interface:

- 1 serial
- 2 parallel
- 3 network, yang biasa disebut Ethernet

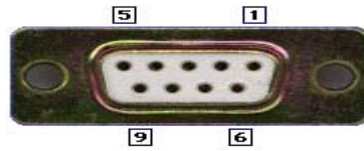
PORT

1. serial port Serial port bersifat asinkron dimana dapat mengirimkan data sebanyak 1 bit dalam tiap satu waktu. Port yang digunakan biasanya menggunakan konektor

DB9. DB9 mempunyai 9 pin yaitu:



DB9 male



DB9 female

keterangan:

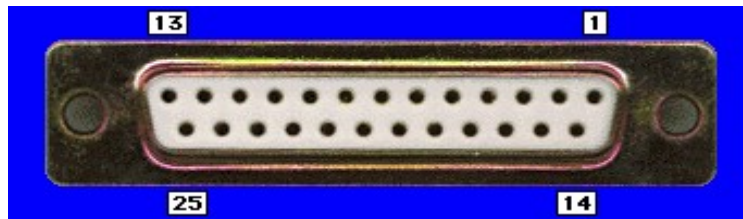
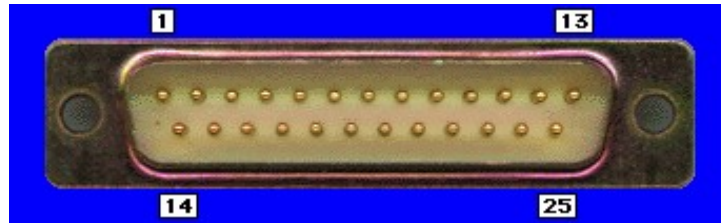
- pin 1 = Data Carrier Detect (DCD)
- pin 2 = Received Data (RxD)
- pin 3 = Transmitted Data (TxD)
- pin 4 = Data Terminal Ready (DTR)
- pin 5 = Signal Ground (common)
- pin 6 = Data Set Ready (DSR)
- pin 7 = Request To Send (RTS)
- pin 8 = Clear To Send (CTS)
- pin 9 = Ring Indicator (RI)

Tabel keterangan pin DB9

Function	Signal	Pin	DTE	DCE
Data	TxD	3	O	I
	RxD	2	I	O
Handshake	RTS	7	O	I
	CTS	8	I	O
	DSR	6	I	O
	DCD	1	I	O
	DTR	4	O	I
Common	Com	5	-	-
Other	RI	9	I	O

2. parallel port

Paralel port dapat mengirimkan 8 bit data sekaligus dalam satu waktu. Paralel port ini menggunakan konektor DB25. Panjang kabel maksimum yang diperlukan / diperbolehkan adalah 15 feet. Contoh peralatan yang menggunakan parallel port adalah: printer, scanner, external driver dsb. DB25 memiliki 25 buah pin dengan gambaran sbb:



Keterangan:

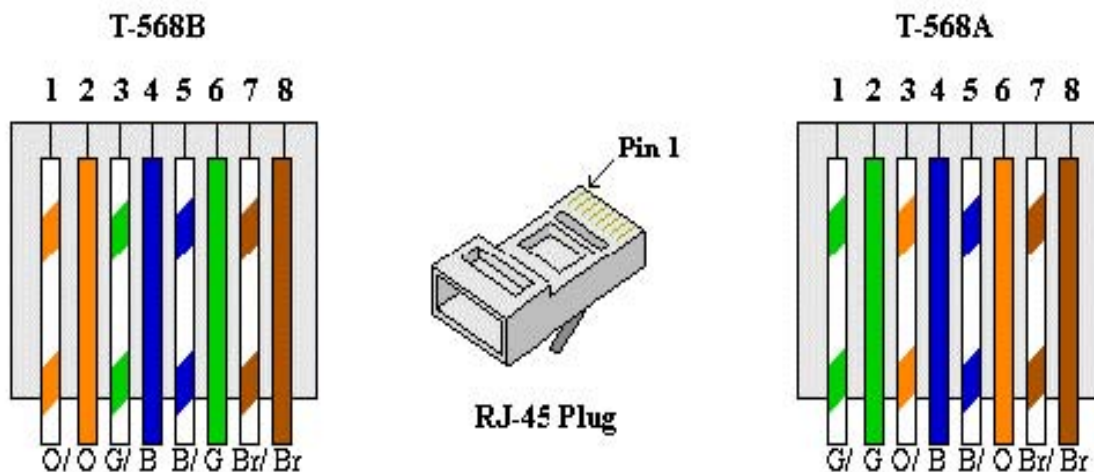
- Control pins
 - o Pin 4 = Request To Send
 - o Pin 5 = Clear to send
 - o Pin 6 = DCE Ready
 - o Pin 8 = received line signal detector
 - o Pin 12 = secondary received line signal detector
 - o Pin 13 = secondary clear to send
 - o Pin 19 = secondary request to send
 - o Pin 20 = DTE ready
- Timing pins
 - o Pin 15 = transmitter signal element timing (DCE-DTE)
 - o Pin 17 = receiver signal element timing (DCE-DTE)
 - o Pin 24 = transmitter signal element timing (DTE-DCT)
- Other pins
 - o Pin 1 = shield
 - o Pin 7 = signal ground / common return
 - o Pin 9 = reserved (testing)
 - o Pin 10 = reserved (testing)
 - o Pin 11 = unassigned
 - o Pin 18 = local loopback
 - o Pin 21 = remote loopback & signal quality detector
 - o Pin 22 = ring indicator
 - o Pin 23 = data signal rate select
 - o Pin 25 = test mode
 - o Pin 2 = transmit data
 - o Pin 3 = receive data

ETHERNET CABLE

Untuk menghubungkan jaringan diperlukan kabel Ethernet yaitu kabel yang digunakan disebut kabel UTP (Unshielded Twisted Pair) dengan menggunakan konektor RJ45. Kabel UTP mempunyai delapan pin (4 pasang).

- Pin1 dengan warna hijau-putih (TD+)
- Pin2 dengan warna hijau (TD-)
- Pin3 dengan warna orange-putih (RD+)
- Pin4 dengan warna biru (NC)
- Pin5 dengan warna biru-putih (NC)
- Pin6 dengan warna orange (RD-)
- Pin7 dengan warna coklat-putih (NC)
- Pin8 dengan warna coklat (NC)

Konfigurasi pin kabel UTP adalah sbb:

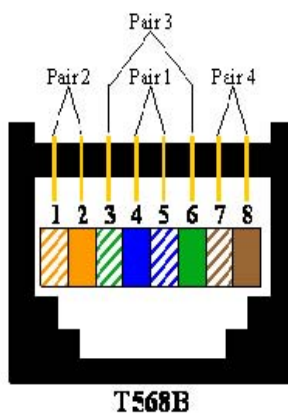


Ada tiga cara pemasangan kabel UTP:

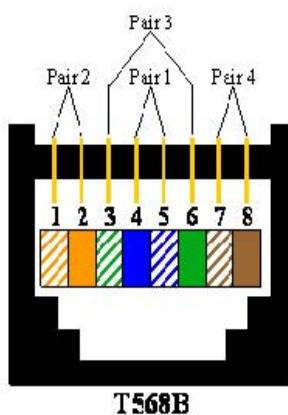
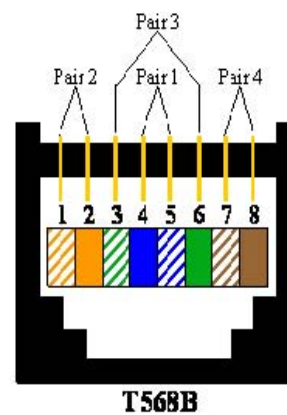
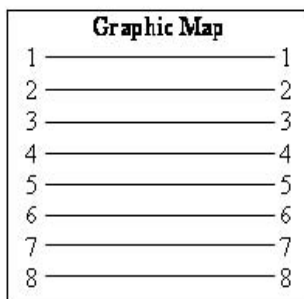
- 1 Straight Through Pengkabelan jenis ini biasanya diperuntukkan untuk menghubungkan peralatan yang berbeda jenis. Misal untuk menghubungkan PC dengan hub, switch dan router, switch dan PC dan sebagainya.
- 2 Cross Over Pengkabelan jenis ini biasanya digunakan untuk menghubungkan peralatan sejenis. Misal untuk menghubungkan PC dengan PC, hub dengan hub dan sebagainya. Pin up kabel cross over sbb:
- 3 Rollover

Pengkabelan jenis ini merupakan pengkabelan khusus. Misalnya untuk menghubungkan antar switch.

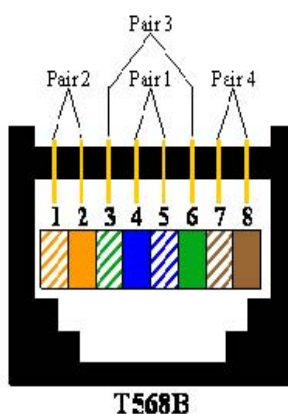
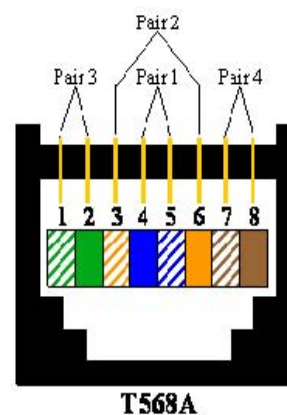
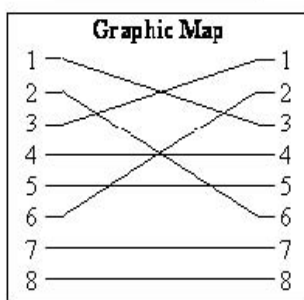
Skema dari tiga jenis kabel di atas adalah sbb: Straight Through, Cross Over, Rollover



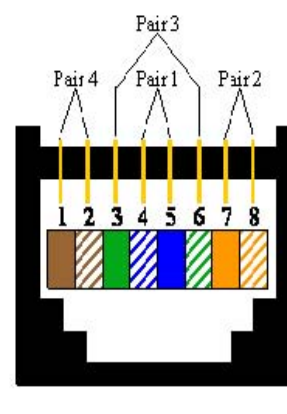
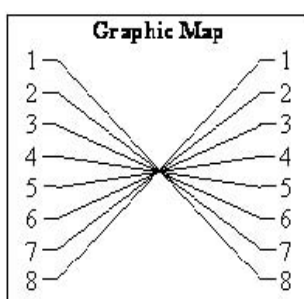
FLUKES
12364573
12364573



FLUKES
12364573
36124573



FLUKES
12635473
87364521



PERALATAN

- 1 kabel UTP
- 2 konektor : DB9, DB25, RJ45
- 3 tang, crimping tool
- 4 FLUKE
- 5 AVOMETER

LANGKAH-LANGKAH PERCOBAAN

- 1 Buat kabel serial dengan DB9 dan DB9.
- 2 Buat kabel serial dengan DB9 dan DB25.
- 3 Buat kabel serial dengan DB25 dan DB25.
- 4 buat kabel serial dengan RJ45 dan DB9.
- 5 Buat kabel Ethernt straight through, cross over dan rollover
- 6 Catat langkah-langkah pembuatannya dan hasil pengukuran dengan alat ukurnya.
7. Operasi fluke DSP 4000
8. Siapkan kabel yang akan dites
9. Hubungkan fluke ke remote (pasangan dari fluke) dengan kabel tersebut
10. Tekan test pada fluke
11. Untuk mengetahui parameter-parameter lain tekan tanda panah
12. Diagram fluke



TUGAS

- 1 Cari datasheet tentang DB9, DB25, RJ45
- 2 Cari datasheet tentang FLUKE yang digunakan

REFERENSI

- 1 http://www.zytrax.com/tech/layer_1/cables/tech_rs232
- 2 <http://www.nullmodem.com/DB-25>
- 3 http://salman.or.id/?page_id=9
- 4 <http://www.arcelect.com>
- 5 <http://www.camiresearch.com>

LEMBAR ANALISA

Tanggal Praktikum

Nama : : : :

NRP

Kelas

SerialDB9–DB9

no	Parameter	Status
1	TX	
2	RX	

DB9–DB25

no	Parameter	Status
1	TX	
2	RX	

DB25DB25

no	Parameter	Status
1	TX	
2	RX	

UTP–DB9

no	Parameter	Status
1	TX	
2	RX	

UTP

no	Parameter	Status
1	TDXAnalyzer	
2	TDR	
3	WireMap	
4	Length	
5	PropagationDelay	
6	DelaySkew	
7	Impedance	
8	Attenuation	
9	Resistance	
10	NEXT	

PERCOBAAN II

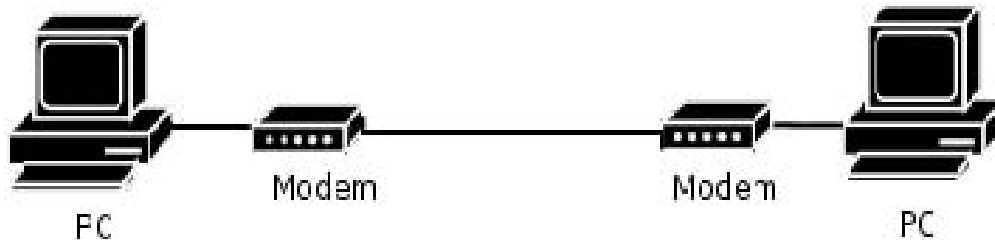
HyperTerminal dan Minicom (Komunikasi Serial dengan Null Modem)

TUJUAN

- 1 Mahasiswa dapat melakukan transfer data dengan menggunakan kabel null modem.
Mahasiswa dapat menghubungkan dua PC untuk dapat berkomunikasi lewat port serial RS-232.
- 2 Mahasiswa dapat menggunakan aplikasi hyperterminal dan minicom.

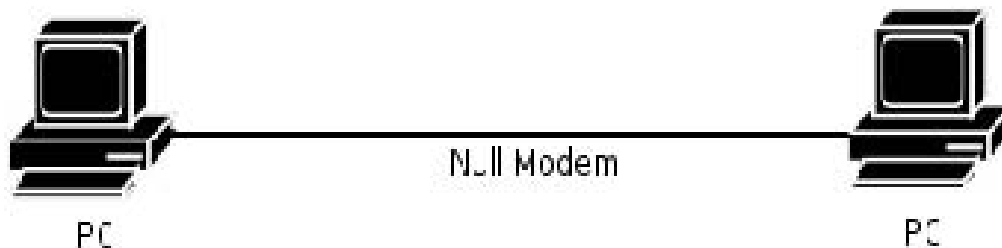
DASAR TEORI

Dasar komunikasi data menggunakan PC dapat dilakukan dengan cara menyambungkan suatu PC dengan modem, seperti Gb. 1.



Gb 1. Komunikasi antar PC dengan Modem

Selain itu dapat juga dilakukan komunikasi data antar 2 PC tanpa menggunakan modem, tetapi menggunakan kabel *nullmodem* seperti pada Gb. 2.



Gb 2: Komunikasi antar PC tanpa Modem (nullmodem)

Ada beberapa parameter untuk melakukan suatu komunikasi data serial, antara lain:

- Bit Rate
- Parity
- Data bit
- Stop bit

Dimana antara 2 PC tersebut harus memiliki kesamaan parameter.

Aplikasi yang digunakan untuk komunikasi serial antara lain :

- Hyper Terminal (OS windows)
- Minicom (OS Linux)

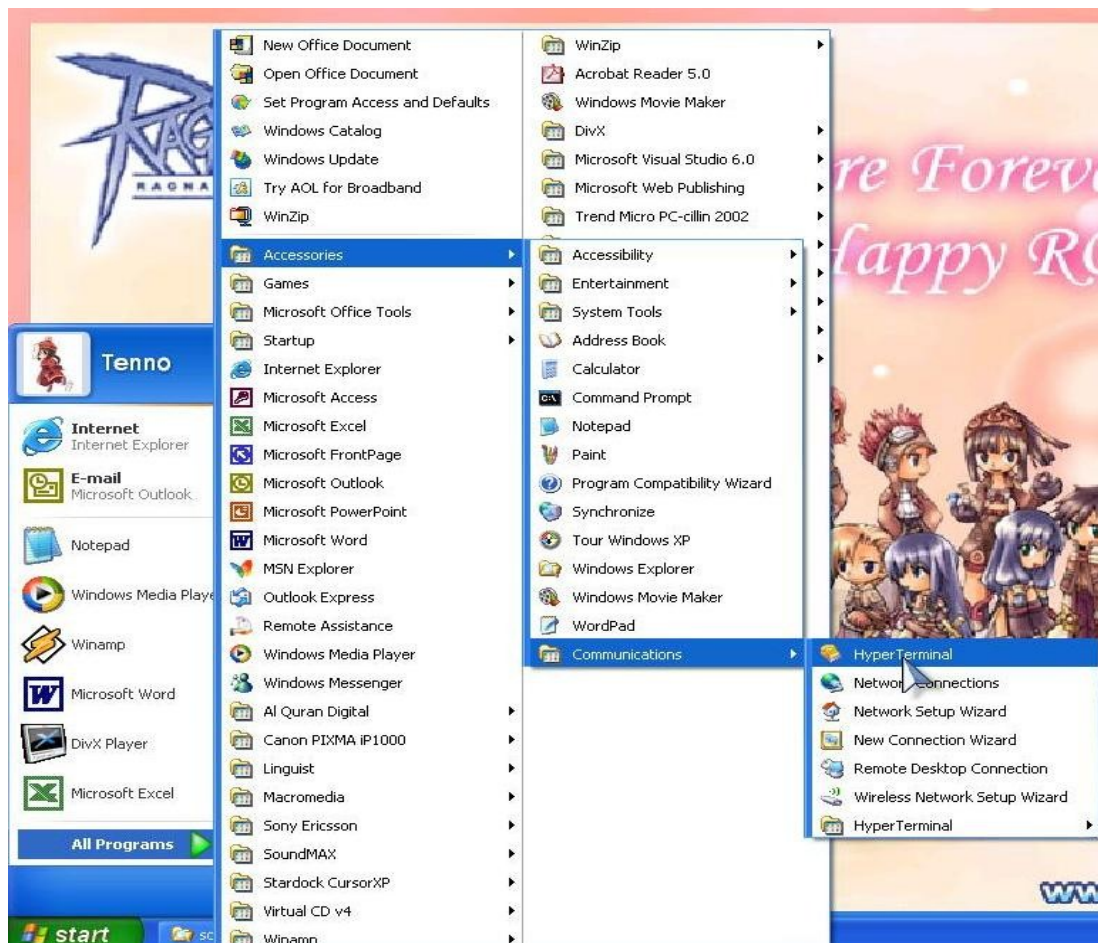
PERALATAN

- 1 Personal Computer (Windows dan Linux)
- 2 Kabel null modem DB9, DB25

LANGKAH-LANGKAH PERCOBAAN

a. Percobaan HyperTerminal

- 1 Hubungkan kedua PC dengan kabel nullmodem, pada port RS-232
- 2 Nyalakan PC
- 3 Klik start-program-accessories-communication-Hyper Terminal. Kemudian ikuti langkah-langkah perintah di Hyper Terminal. (Gb. 3)



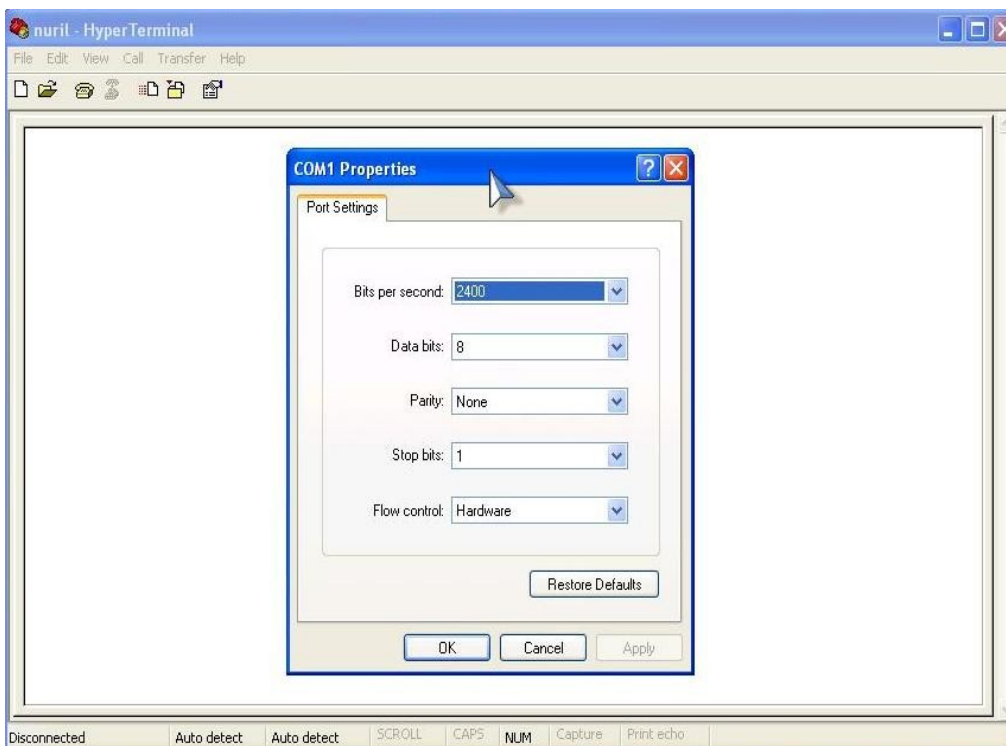
Gb 3: Menjalankan HyperTerminal

4. Beri nama pada koneksi hyperterminal (Gb. 4)



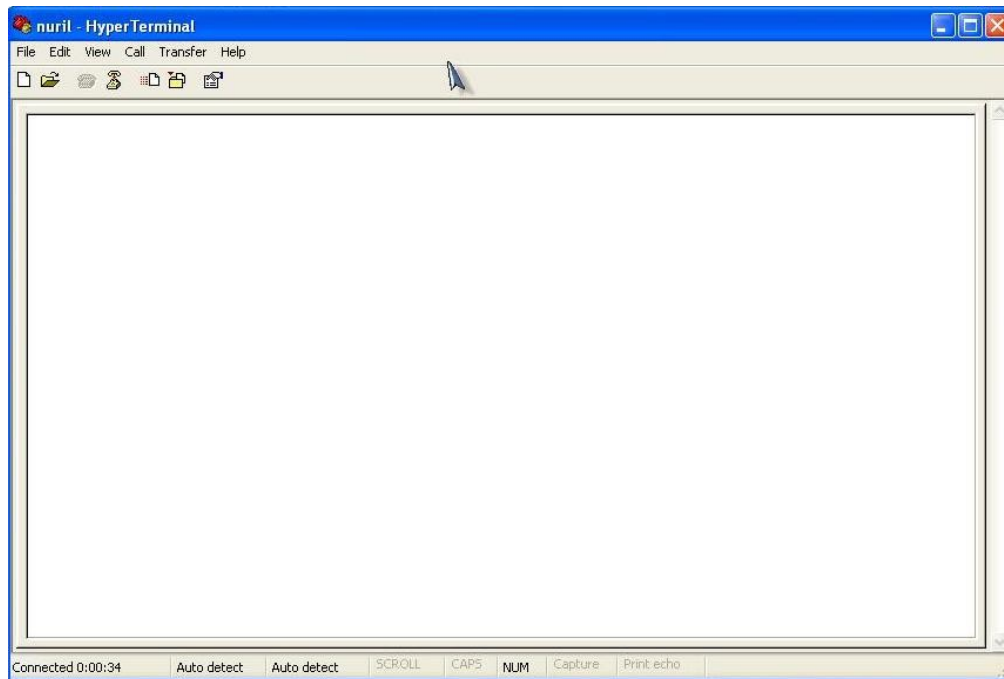
Gb 4: Memberi nama pada koneksi

5. Set parameter pada hyperterminal (Gb. 5)



Gb 5: Parameter pada hyperterminal

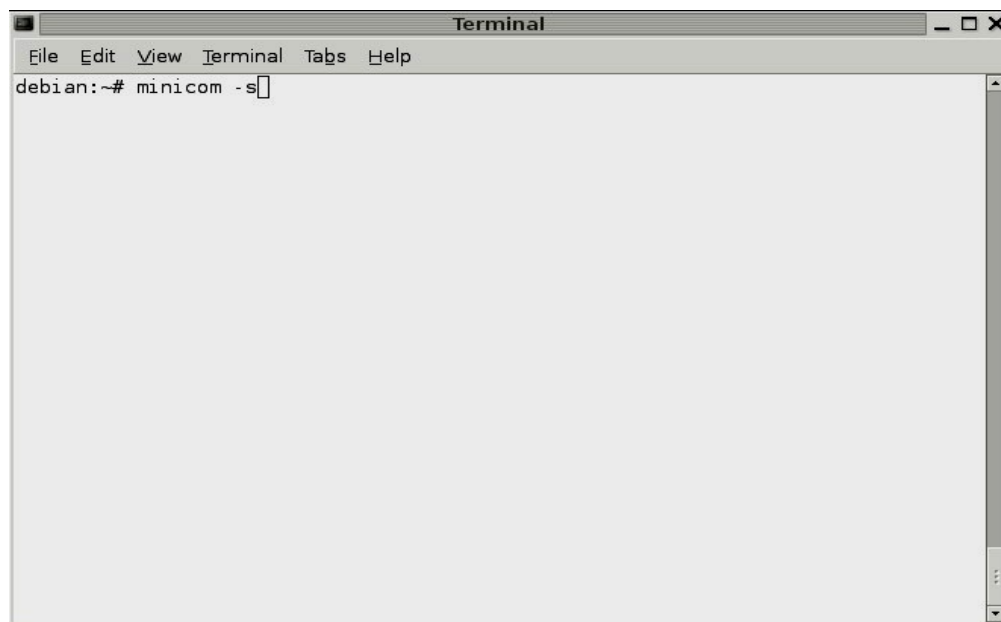
6. Connect !!! (Gb. 6)



Gb 6: Connected

b. Percobaan Minicom

1. Jalankan aplikasi Minicom (Gb. 7)



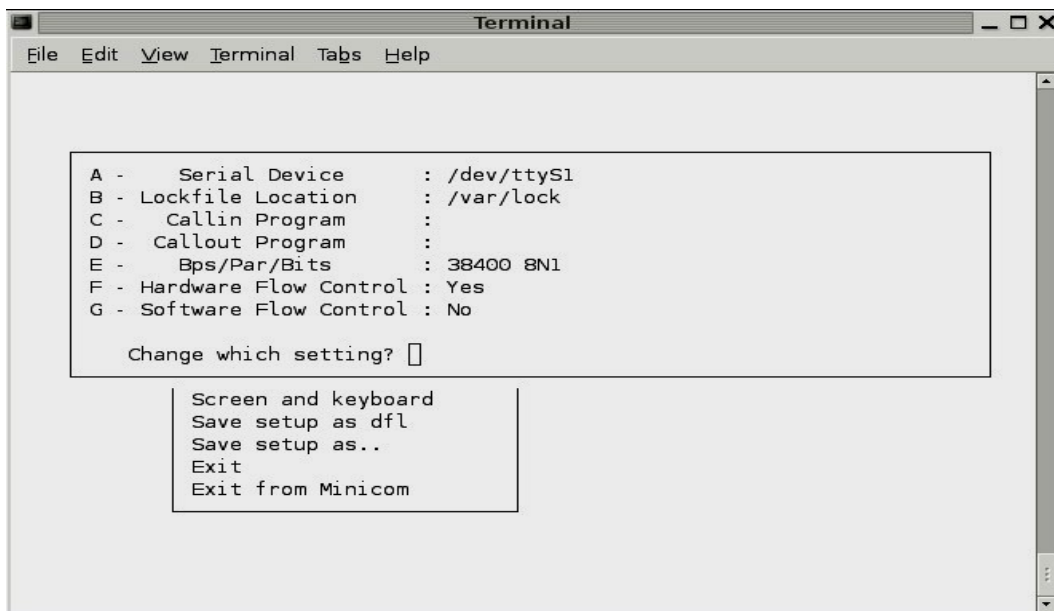
Gb 7: # minicom -s

2. Pilih Serial Port pada menu (Gb. 8)



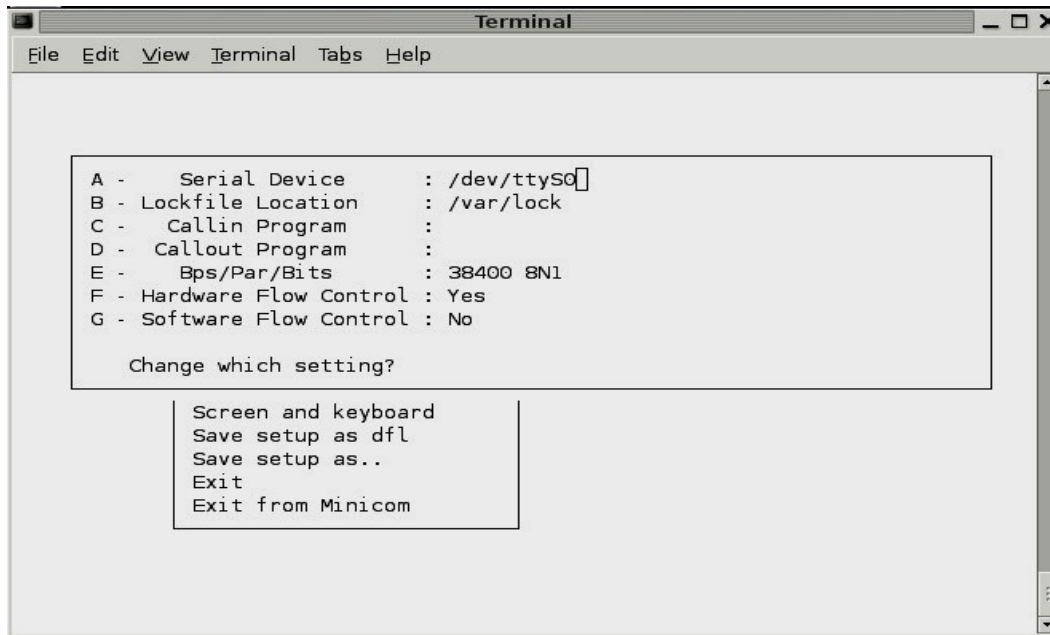
Gb 8: Serial Port Setup

3. Maka akan muncul menu seperti pada Gb. 9



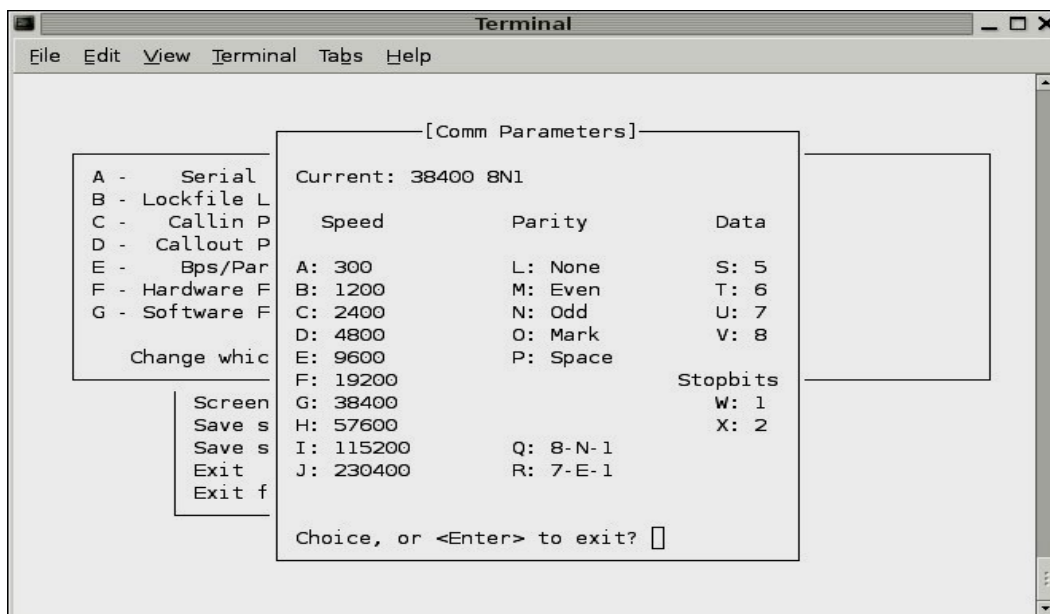
Gb 9: Menu serial port

4. Ganti perangkat serial menjadi /dev/ttyS0 dengan menekan huruf “A”



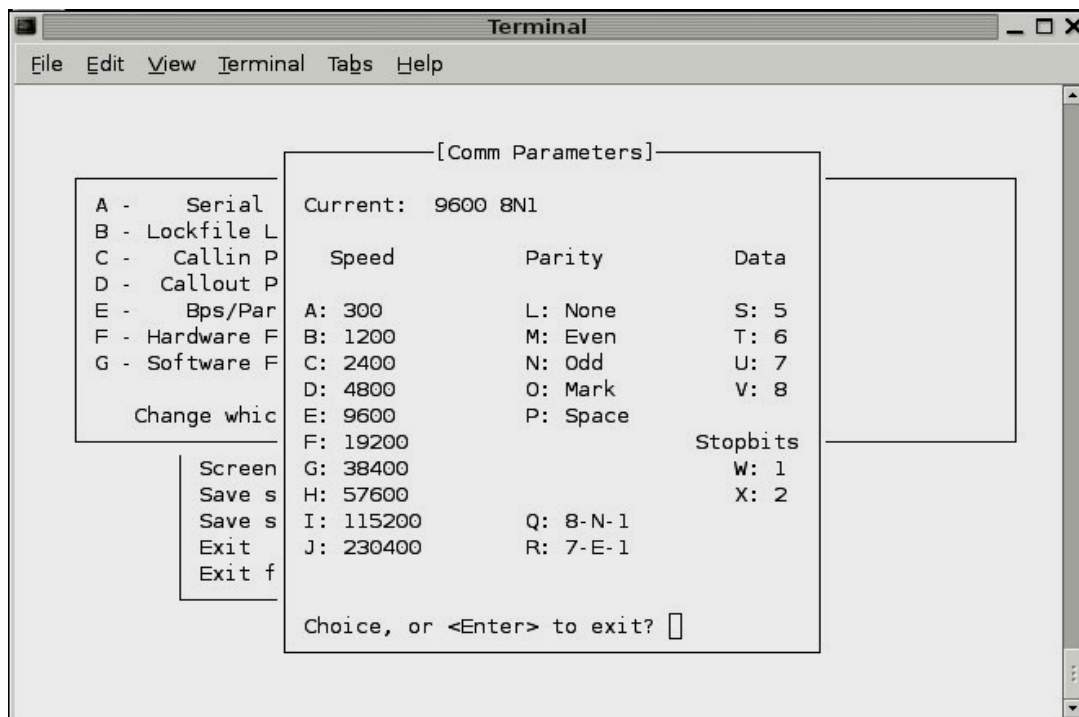
Gb 10: Tekan tombol A, ganti menjadi /dev/ttyS0

5. Masuk pada menu “E” Bps/Par/Bits (Gb.11)



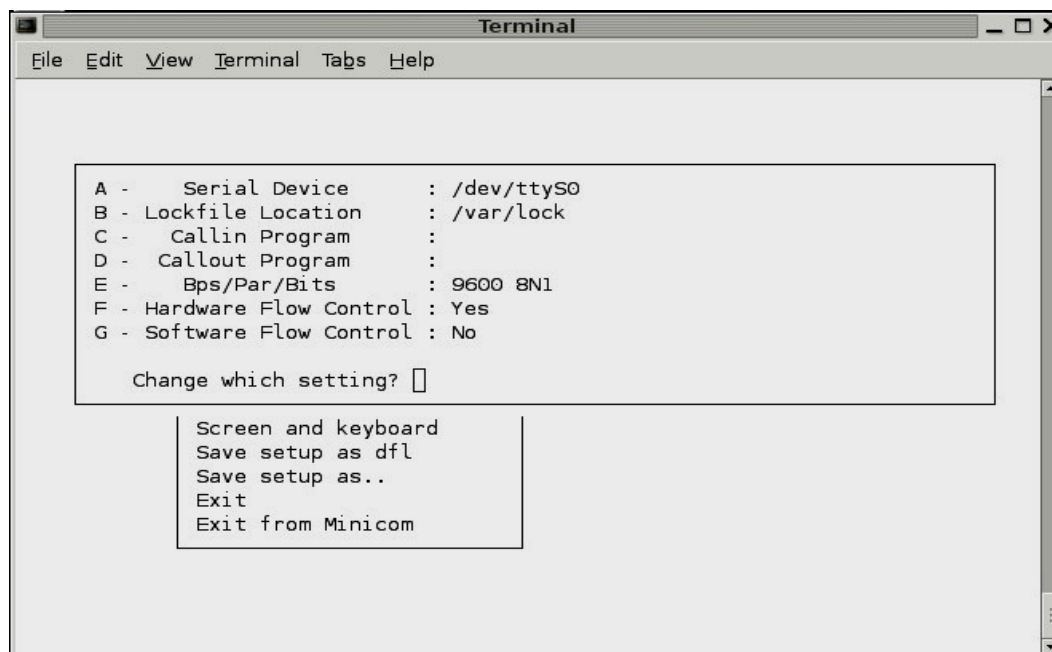
Gb 11: Tekan "E" untuk mengganti parameter

6. Ganti nilai “Current 38400 8N1” menjadi “9600 8N1” dengan menekan tombol “E”



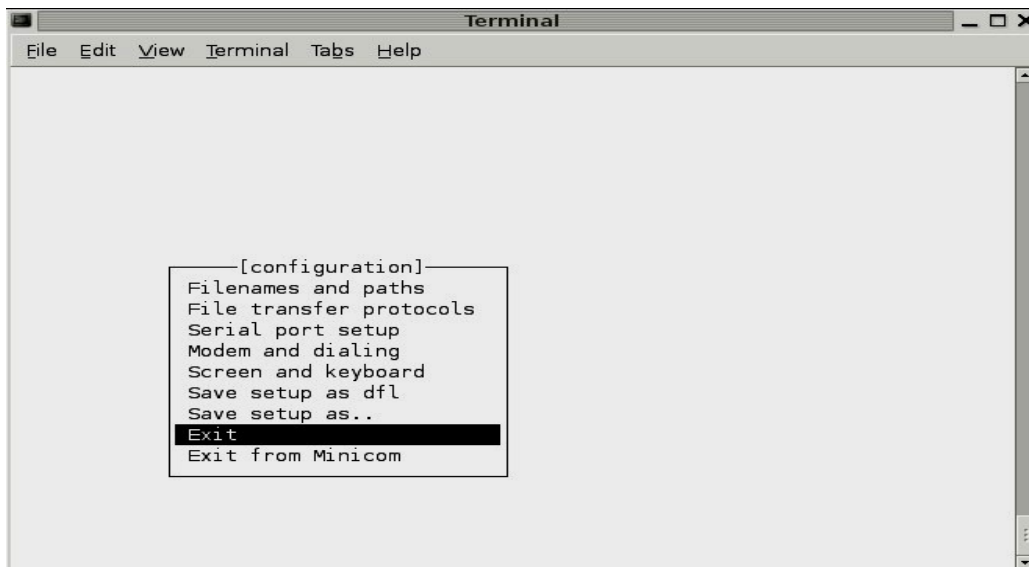
Gb 12: Parameter baru

7. Tekan Enter untuk kembali ke menu awal



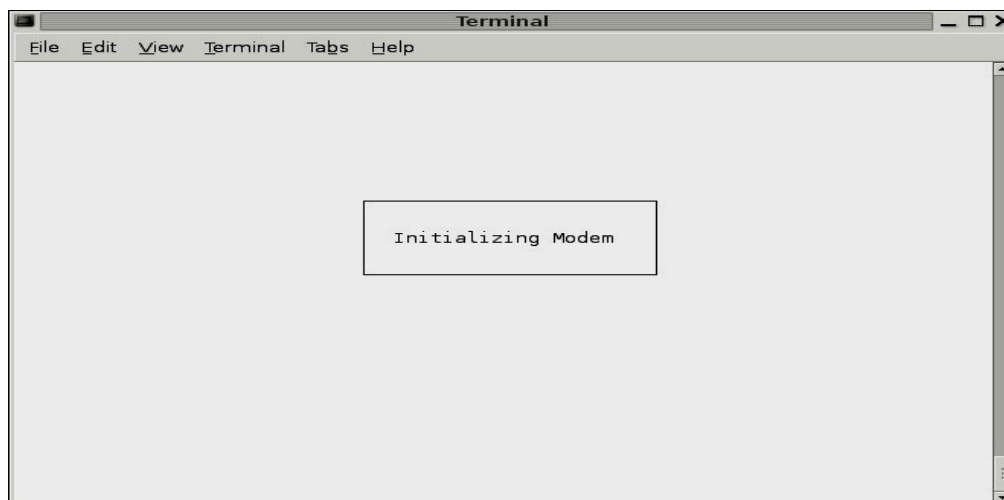
Gb 13: Kembali ke menu awal

8. Pilih "Exit" untuk memulai komunikasi



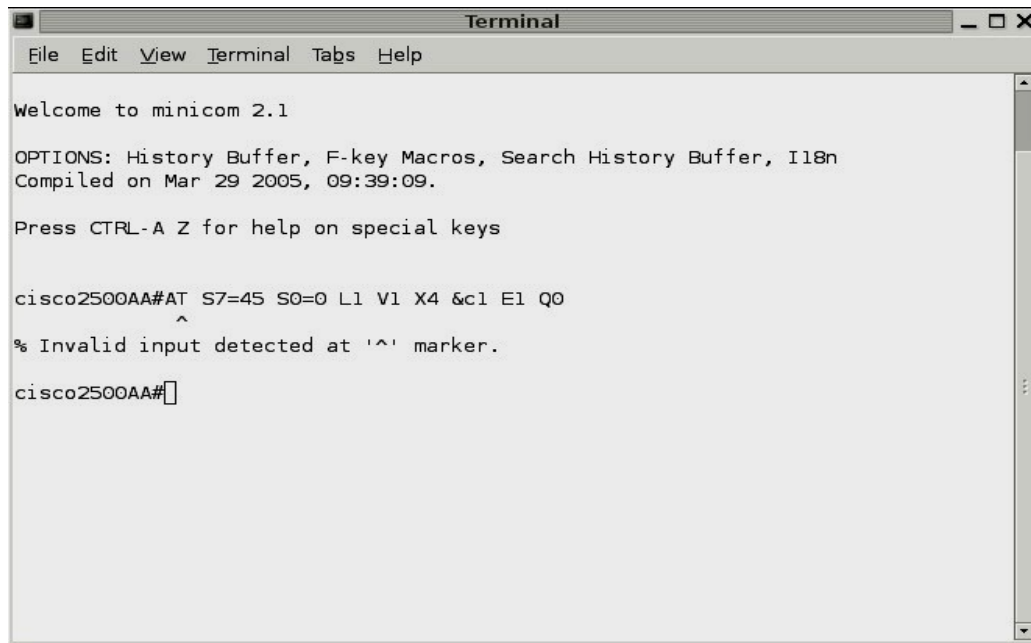
Gb 14: Keluar dari setting parameter

9. Minicom melakukan “Inisialisasi”



Gb 15: Inisialisasi

10. Melakukan komunikasi



Gb 16: Memulai komunikasi

11. Untuk masuk ke menu Tekan “Ctrl+A” kemudian “Z”

c. Percobaan pengiriman file

1. Ambil 2 buah file, catat ukuran file tersebut
2. Lakukan pengiriman file dengan parameter sebagai berikut
 - 2400 8N1
 - 9600 8N1
 - 115200 8N1
3. Catat waktu yang dibutuhkan !

TUGAS

1. Artikan parameter-parameter dari komunikasi serial
2. Apa yang dimaksud dengan LAPLINK

Referensi

- Minicom (1) , Linux Manpage
- <http://www.microsoft.com>

LEMBAR ANALISA

Praktikum Komunikasi Data – 2

Tanggal Praktikum :
Nama :
NRP :
Kelas :

A. Percobaan Hyperterminal

B. Percobaan Minicom

C. Percobaan Pengiriman File

UkuranFile	Byte
Parameter	Waktu
24008N1	
96008N1	
1152008N1	

Percobaan III

Pengukuran Komunikasi Serial

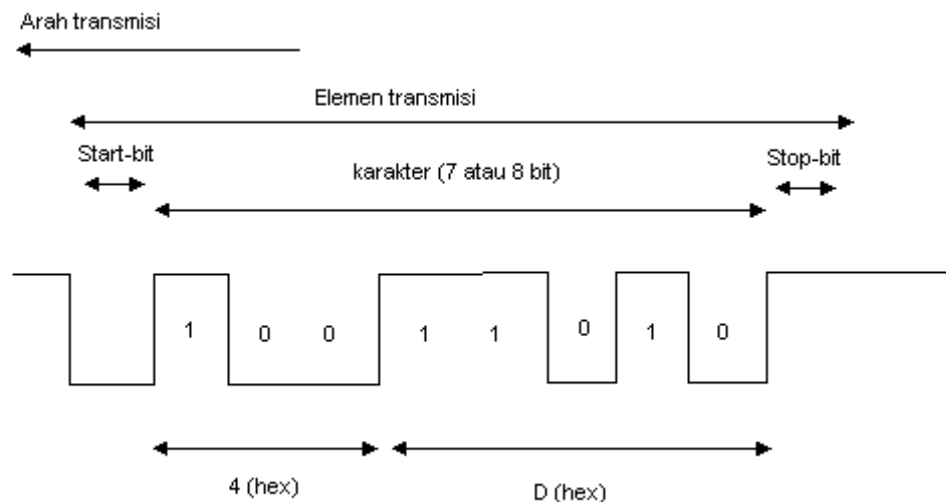
1. Tujuan

- Mampu menghubungkan 2 PC untuk dapat berkomunikasi lewat port serial RS-232
- Mengetahui system pengkabelan untuk menghubungkan 2 PC lewat port serial
- Mengetahui cara pengukuran sinyal digital pada saluran komunikasi data serial 2 PC
- Mengetahui adanya pulsa start bit, data bit (word), stop bit, even parity, dan odd parity
- Dapat membedakan setting port ada pengiriman kode ASCII secara sinkron dengan membaca karakteristik susunan kode binary dan dikirimkan

2 Dasar Teori

Metode transmisi ini dipergunakan bila data ingin dikirimkan ternyata dibangkitkan dengan interval yang random (contohnya adalah mengirim karakter), maka setiap sinyal dari tiap karakter dalam jalur transmisi akan diberi idle (marking) dengan interval yang panjang diantara beberapa karakter.

Pada komunikasi asinkron ini maka receiver harus dapat mensinkronkan kembali permulaan dari tiap karakter yang diterima karena karakter yang dikirim telah terbungkus oleh satu start bit dan satu atau dua stopbit serta kemungkinan ditambahkan nyaparity, tetapi unstuck percobaan ini tidak dilakukan penambahan parity.



Gb 1. Karakter M pada parameter 7E1

Pada Gb.1 ditunjukkan bahwa polaritas antara start bit dan stopbit adalah berbeda, perbedaan ini untuk meyakinkan bahwa minimum selalu ada transisi (101)

di antara masing-masing urutan karakter, dengan mengabaikan urutan bit dalam karakter yang dikirimkan. Pertama kali transisi 10 sesudah periode adanya idle akan dipergunakan receiver untuk menentukan permulaan karakter baru.

Kode ASCII merupakan kode alpha numeric yang paling populer yang dipakai dalam teknik telekomunikasi. Untuk mencari kode biner dari tiap karakter ASCII dapat dipergunakan table ASCII, sedangkan cara membaca table yaitu pada grafik kita ubah dulu menjadi bilangan biner, kemudian diubah menjadi bilangan hexadecimal dan kita bandingkan bilangan hexadecimal tersebut dengan table dibawah ini

Contoh:

1001101 (biner) 4D (hex) karakter M (ASCII)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	#32; Space	64	40	100	#64; @	96	60	140	#96; `			
1	1	001	SOH (start of heading)	33	21	041	#33; !	65	41	101	#65; A	97	61	141	#97; a			
2	2	002	STX (start of text)	34	22	042	#34; "	66	42	102	#66; B	98	62	142	#98; b			
3	3	003	ETX (end of text)	35	23	043	#35; #	67	43	103	#67; C	99	63	143	#99; c			
4	4	004	ECT (end of transmission)	36	24	044	#36; \$	68	44	104	#68; D	100	64	144	#100; d			
5	5	005	ENQ (enquiry)	37	25	045	#37; %	69	45	105	#69; E	101	65	145	#101; e			
6	6	006	ACK (acknowledge)	38	26	046	#38; &	70	46	106	#70; F	102	66	146	#102; f			
7	7	007	BEL (bell)	39	27	047	#39; '	71	47	107	#71; G	103	67	147	#103; g			
8	8	010	BS (backspace)	40	28	050	#40; (72	48	110	#72; H	104	68	150	#104; h			
9	9	011	TAB (horizontal tab)	41	29	051	#41;)	73	49	111	#73; I	105	69	151	#105; i			
10	A	012	LF (NL line feed, new line)	42	2A	052	#42; *	74	4A	112	#74; J	106	6A	152	#106; j			
11	B	013	VT (vertical tab)	43	2B	053	#43; +	75	4B	113	#75; K	107	6B	153	#107; k			
12	C	014	FF (NP form feed, new page)	44	2C	054	#44; ,	76	4C	114	#76; L	108	6C	154	#108; l			
13	D	015	CR (carriage return)	45	2D	055	#45; -	77	4D	115	#77; M	109	6D	155	#109; m			
14	E	016	SO (shift out)	46	2E	056	#46; .	78	4E	116	#78; N	110	6E	156	#110; n			
15	F	017	SI (shift in)	47	2F	057	#47; /	79	4F	117	#79; O	111	6F	157	#111; o			
16	10	020	DLE (data link escape)	48	30	060	#48; 0	80	50	120	#80; P	112	70	160	#112; p			
17	11	021	DC1 (device control 1)	49	31	061	#49; 1	81	51	121	#81; Q	113	71	161	#113; q			
18	12	022	DC2 (device control 2)	50	32	062	#50; 2	82	52	122	#82; R	114	72	162	#114; r			
19	13	023	DC3 (device control 3)	51	33	063	#51; 3	83	53	123	#83; S	115	73	163	#115; s			
20	14	024	DC4 (device control 4)	52	34	064	#52; 4	84	54	124	#84; T	116	74	164	#116; t			
21	15	025	NAK (negative acknowledge)	53	35	065	#53; 5	85	55	125	#85; U	117	75	165	#117; u			
22	16	026	SYN (synchronous idle)	54	36	066	#54; 6	86	56	126	#86; V	118	76	166	#118; v			
23	17	027	ETB (end of trans. block)	55	37	067	#55; 7	87	57	127	#87; W	119	77	167	#119; w			
24	18	030	CAN (cancel)	56	38	070	#56; 8	88	58	130	#88; X	120	78	170	#120; x			
25	19	031	EM (end of medium)	57	39	071	#57; 9	89	59	131	#89; Y	121	79	171	#121; y			
26	1A	032	SUB (substitute)	58	3A	072	#58; :	90	5A	132	#90; Z	122	7A	172	#122; z			
27	1B	033	ESC (escape)	59	3B	073	#59; ;	91	5B	133	#91; [123	7B	173	#123; {			
28	1C	034	FS (file separator)	60	3C	074	#60; <	92	5C	134	#92; \	124	7C	174	#124;			
29	1D	035	GS (group separator)	61	3D	075	#61; =	93	5D	135	#93;]	125	7D	175	#125; }			
30	1E	036	RS (record separator)	62	3E	076	#62; >	94	5E	136	#94; ^	126	7E	176	#126; ~			
31	1F	037	US (unit separator)	63	3F	077	#63; ?	95	5F	137	#95; _	127	7F	177	#127; DEL			

Source: www.LookupTables.com

Gb 2. Table ASCII

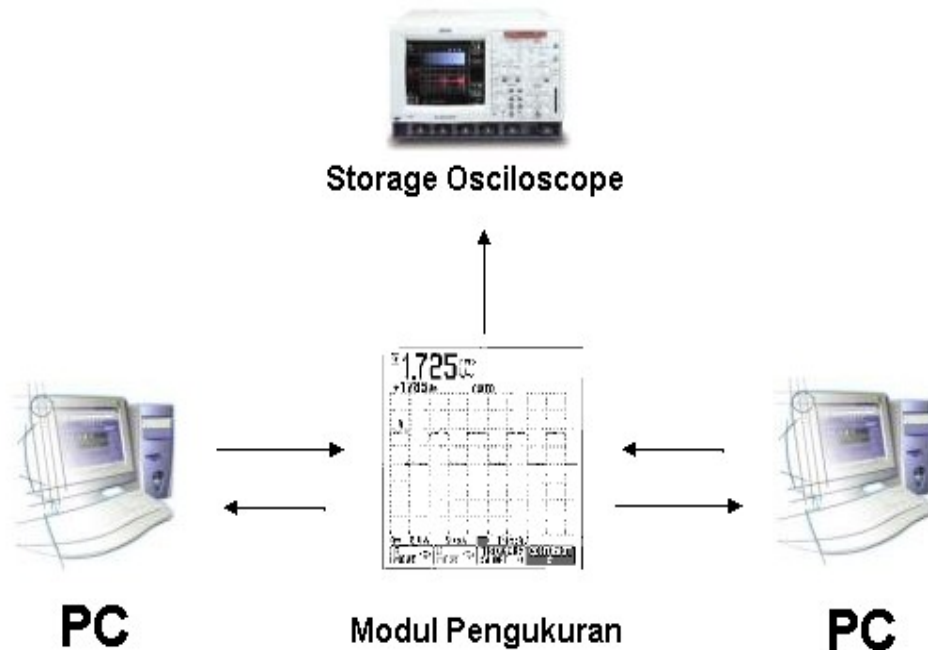
3. Alat-alat

- Personal Komputer
- Kabel serial
- Modul pengukuran
- StorageOscilloscope
- Hyperterminal
- Minicom

4. Langkah Percobaan

1. Persiapan

- Sambungkan kabel serial pada PC 1 dan PC 2 (serial port).
- Pada computer ke 1 terminal transmit data (txD) yang berada pada pin no 2 dihubungkan ke computer ke 2 pada terminal receiver data (RxD) yang berada pada pin no 3.
- Pada computer ke1 terminal receiver data (RxD) yang berada pada pin no 3 dihubungkan ke computer ke 2 pada terminal transmit data (TxD) yang berada pada pin no 2
- Pada computer ke 1 terminal ground yang berada pada pin no 7 dihubungkan ke computer ke 2 pada terminal ground yang berada pada pin no 7.



Gb 3: Rangkaian percobaan

5. Petunjuk praktikum

- Setting Hyperterminal/minicom pada kedua PC supaya ada pattern koneksi seperti praktikum sebelumnya
- Setting Hyperterminal: Bits per second= 2400 Data bits= 8 Parity= none Stop bits= 1 Flow Control= none
- Ketik sebuah karakter "M", kemudian pause pada DSO (Digital Storage Oscilloscope) dan gambar pada kertas Milimeter block-
Atur display pada DSO dengan mengatur [VOLT/DIV], [TIME/DIV] dan [DELAY]-Tunjukkan bagian mana yang disebut startbit, data, stopbit.-

- Kemudian lakukan hal yang sama untuk karakter “123”, “ABC” dan “a
b c”-Kemudian ganti parameter untuk kecepatan yang lainnya, parity lainnya.-
Buat laporan sementara.

6. Tugas

Lampirkan table ASCII dan extended ASCII

7. Referensi

- [Http://www.lookuptables.com](http://www.lookuptables.com)
- William Stallings, “Data and Computer Communications 7th edition”, 2003

LEMBAR ANALISA

Praktikum Komunikasi Data – 3

Tanggal Praktikum :

Nama :

NRP :

Kelas :

1 Parameter : 2400 8N1
Karakter M

2 Parameter : 2400 8N1

Karakter “A B C”

Karakter “1 2 3”

Karakter “a b c”

3. Parameter :-

Karakter M

Karakter “A B C”

Karakter “1 2 3”

Karakter “a b c”

PERCOBAAN IV

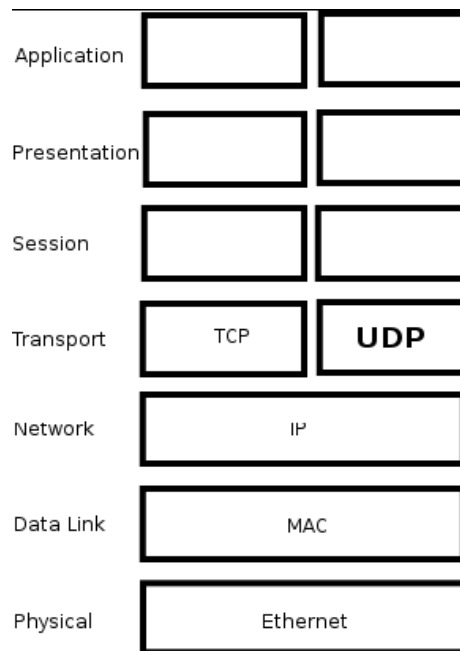
Socket Programming User Datagram Protocol (UDP)

1. TUJUAN

- Mahasiswa dapat memahami cara kerja protokol UDP
- Mahasiswa dapat membuat aplikasi clientserver

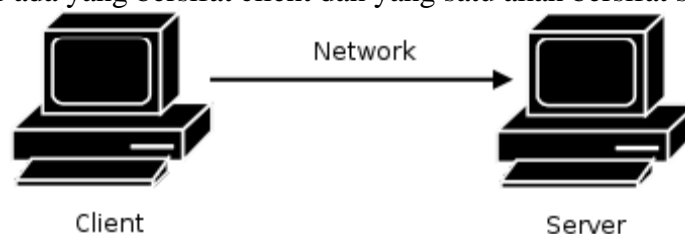
2. DASAR TEORI

UDP adalah suatu protocol pengiriman data yang berbasis Internet Protocol (IP) dan bersifat *connection-less oriented*. Pada OSI layer UDP berada pada layer transport yang fungsinya mengatur pengiriman suatu data dari client ke server.



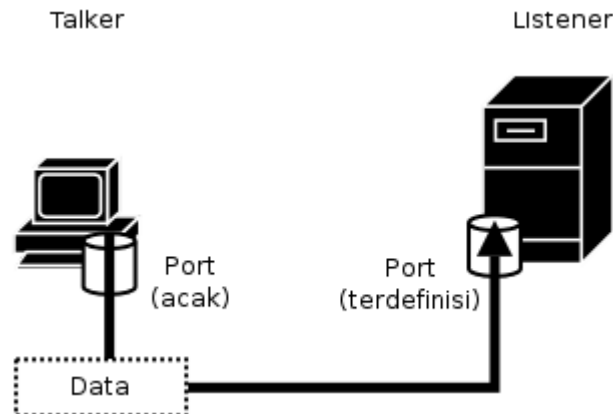
Gb 1. UDP pada OSI Layer

Model komunikasi data dengan client server artinya pada saat pengiriman data, salah satu komputer ada yang bersifat client dan yang satu akan bersifat sebagai server.



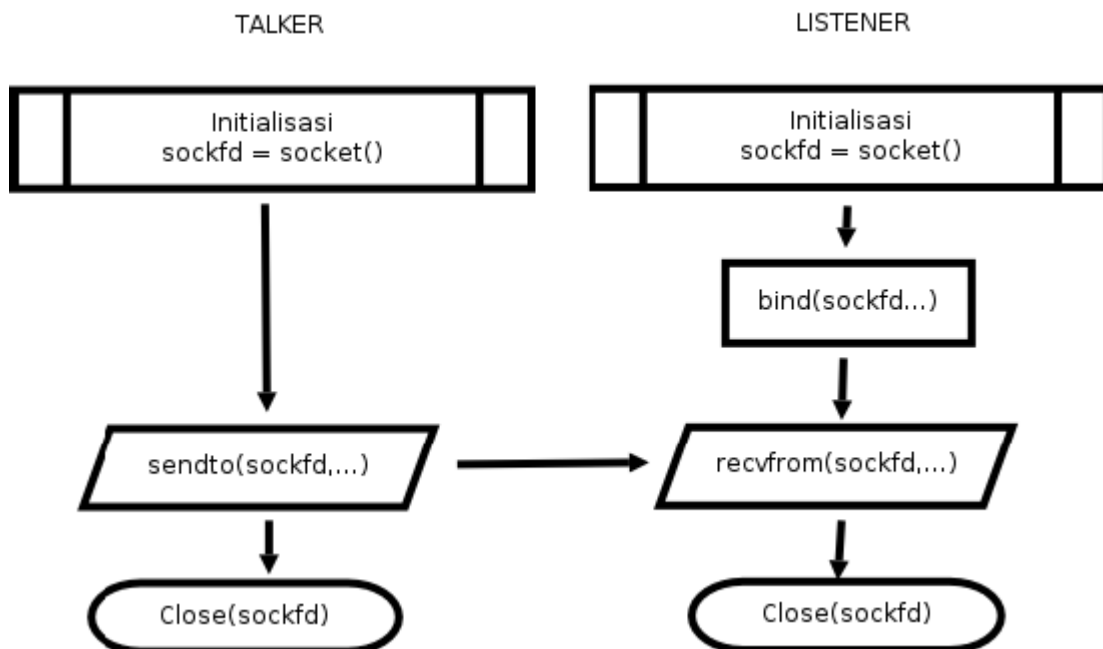
Gb 2. Client Server

Untuk pengiriman datanya, pada masing-masing computer (client-server) akan Menggunakan *port* dengan pendefinisian terlebih dahulu. Kemudian dari client akan mengirimkan data dari port pada PCnya kearah port pada PC servernya. Apabila port tersebut sudah digunakan oleh aplikasi lainnya maka akan terjadi error apabila aplikasi yang kita jalankan menggunakan port yang sama. Jumlah port yang ada 65535 digunakan sesuai dengan aplikasi yang sudah distandarkan.



Gb 3. Pengiriman data melalui PORT

Alur penggunaan socket programming untuk UDP seperti pada Gb. 4.



Gb 4. Alur socket programming pada UDP

3. PERALATAN

- PC (Linux OS)
- GCC
- UTP Cable
- Hub / Switch (optional)

4. LANGKAH PRAKTIKUM

1. Sebelum PC menyala, Sambungkan PC ke jaringan, apabila dihubungkan dengan switch/hub gunakan kabel UTP straigh through apabila dengan PC langsung gunakan kabel crossover
2. Nyalakan PC hingga proses booting sempurna.
3. Login dengan user “*root*” dan password “*root*” (isikan tanpa tanda petik)
4. jalankan perintah *ifconfig eth0* , kemudian catat IP addressnya.

contoh :

```
highway:~# ifconfig eth0
eth0      Link encap:Ethernet HWaddr 00:13:D4:CC:4E:2A
          inet addr:192.168.0.167 Bcast:192.168.0.255 Mask:255.255.255.0
          inet6 addr: fe80::213:d4ff:fecc:4e2a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:5630764 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2730184 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2746138798 (2.5 GiB) TX bytes:404795862 (386.0 MiB)
          Base address:0xdc00 Memory:febc0000febe0000
```

IP address kita adalah 192.168.0.167

• Tips untuk asisten :

Apabila PC belum memiliki IP, asisten diharapkan memberikan IP dengan perintah:

```
# ifconfig eth0 192.168.0.*
```

Dimana * digantikan dengan angka 1 – 254, dengan syarat tidak ada IP yang sama antar PC

5. Buat direktori dengan nama kelas – group, contoh : 2D4TA1

```
# cd
```

```
# mkdir 2D4TA1
```

Tanda # tidak perlu ikut diketik !!!

6. Masuk ke direktori tersebut ...

```
# cd 2D4TA1
```

7. Untuk memulai pengetikan program di linux, peserta menggunakan program “VIM”. Ketikkan source program *lister.c* atau *talker.c* dengan perintah :

```
# vim listerner.c
```

atau

```
# vim talker.c
```

• Tips untuk asisten:

Peserta praktikum dibagi menjadi beberapa kelompok dan ditunjuk supaya ada yang memprogram *lister* dan ada yang memprogram *talker*

8. Tekan tombol “Ins” / “Insert” pada keyboard untuk memulai pengetikan hingga terlihat indikasi “INSERT” di layar bagian bawah kiri. Ketik sesuai dengan source pada lampiran.
9. Setelah semua source di ketik, simpan source tersebut dengan menekan tombol “ESC”, hingga indikator “-INSERT” hilang, dilanjutkan dengan “:wq” (tanpa tanda petik).
10. Lakukan kompilasi program dengan cara :
 Untuk program listener.c
 # gcc -o listener listener.c
 Untuk program talker.c
 # gcc -o talker talker.c
 Apabila terjadi error, lakukan pengecekan dengan membuka file source seperti pada langkah ke7.
11. Jalankan program dengan perintah, sebagai berikut :
 Untuk listener :
 # ./listener
 Untuk talker :
 # ./talker 192.168.0.25 “percobaan pesan”
 Dimana 192.168.0.25 adalah IP dari komputer yang melakukan pemrograman *listener*. Pesan yang dikirim adalah *percobaan pesan*. Pada komputer yang menjalankan program listener akan tampil data text tersebut.
 * Tips : Untuk mematikan program lakukan dengan menekan “Ctrl + C”
12. Lakukan pengiriman text tersebut dengan kondisi sebagai berikut, kemudian amati pada komputer tersebut dan apabila muncul error catat di laporan sementara!
 1. Program listener dijalankan di komputer A, pada komputer B kirim pesan dengan program talker ke komputer A.
 2. Matikan program listener pada komputer A, pada komputer B kirim pesan dengan program talker ke komputer A.

5. TUGAS

1. Lampirkan RFC yang berhubungan dengan protokol UDP (RFC 768)

6. REFERENSI

- RFC 768
- man socket, bind, sendto, recvfrom

LEMBAR ANALISA

Praktikum Komunikasi Data – 4

Tanggal praktikum:

Nama:

NRP:

Kelas:

IP Listener :IP Talker:

<i>No</i>	<i>Listener</i>	<i>Talker</i>	<i>Pesan Error</i>
1.	Dijalankan	Dijalankan	
2.	Dimatikan	Dijalankan	

LAMPIRAN

```
/*
** Praktikum Komunikasi Data
** talker.c – a datagram "client" demo
** by Sritrusta Sukaridhoto, ST
*/

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>

#define MYPORT 4950                                // the port users will be connecting to

int main(int argc, char *argv[])
{
    int sockfd;
    struct sockaddr_in their_addr;                  // connector's address information
    struct hostent *he;
    int numbytes;

    if (argc != 3) {
        fprintf(stderr, "usage: talker hostname message\n");
        exit(1);
    }

    if ((he=gethostbyname(argv[1])) == NULL) {      // get the host info
        perror("gethostbyname");
        exit(1);
    }

    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == 1){
        perror("socket");
        exit(1);
    }

    their_addr.sin_family = AF_INET;                // host byte order
    their_addr.sin_port = htons(MYPORT);            // short, network byte order
    their_addr.sin_addr = *((struct in_addr *)he->h_addr);
    memset(&(their_addr.sin_zero), '\0', 8);        // zero the rest of the struct

    if ((numbytes=sendto(sockfd, argv[2], strlen(argv[2]), 0, (struct sockaddr *)&their_addr,
        sizeof(struct sockaddr))) == 1)
    {
        perror("sendto");
        exit(1);
    }

    printf("sent %d bytes to %s\n", numbytes, inet_ntoa(their_addr.sin_addr));
}
```



```

        close(sockfd);

        return 0;
    }

/*
** Praktikum Komunikasi Data
** listener.c – a datagram sockets "server" demo
** by Sritrusta Sukaridhoto, ST
*/

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#define MYPORT 4950                // the port users will be connecting to

#define MAXBUFLEN 100

int main(void)
{
    int sockfd;
    struct sockaddr_in my_addr;           // my address information
    struct sockaddr_in their_addr;        // connector's address information
    int addr_len, numbytes;
    char buf[MAXBUFLEN];

    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == 1) {
        perror("socket");
        exit(1);
    }

    my_addr.sin_family = AF_INET;         // host byte order
    my_addr.sin_port = htons(MYPORT);     // short, network byte order
    my_addr.sin_addr.s_addr = INADDR_ANY; // automatically fill with my IP
    memset(&(my_addr.sin_zero), '\0', 8); // zero the rest of the struct

    if (bind(sockfd, (struct sockaddr *)&my_addr, sizeof(struct sockaddr)) == 1) {
        perror("bind");
        exit(1);
    }

    addr_len = sizeof(struct sockaddr);
    if ((numbytes=recvfrom(sockfd, buf, MAXBUFLEN, 0, (struct sockaddr *)&their_addr,
        &addr_len)) == 1) {
        perror("recvfrom");
        exit(1);
    }

    printf("got packet from %s\n", inet_ntoa(their_addr.sin_addr));
}

```

```
printf("packet is %d bytes long\n",numbytes);
buf[numbytes] = '\0';
printf("packet contains \"%s\"\n",buf);

close(sockfd);

return 0;
}
```

PERCOBAAN V

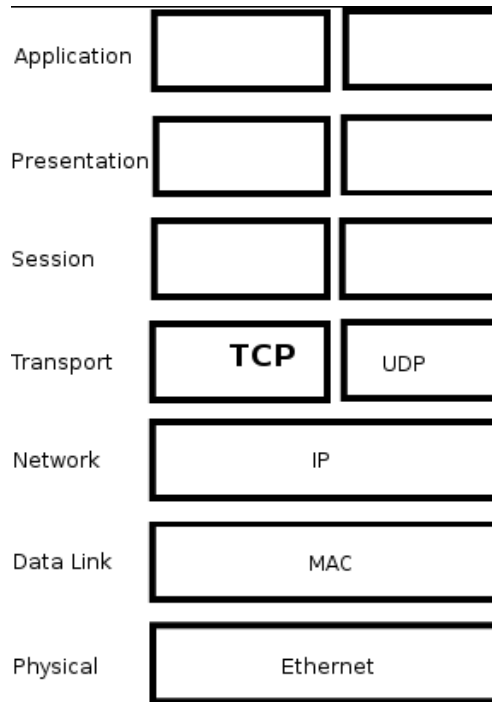
Socket Programming Transmission Control Protocol (TCP)

1. TUJUAN

- Mahasiswa dapat memahami cara kerja protokol TCP
- Mahasiswa dapat membuat aplikasi clientserver

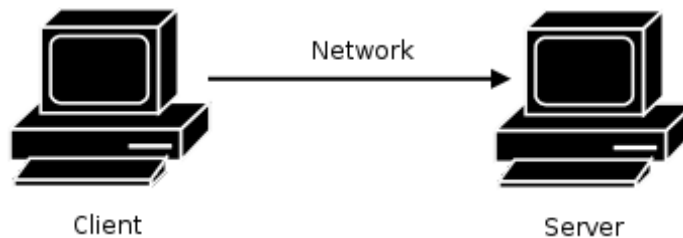
2. DASAR TEORI

TCP adalah suatu protocol pengiriman data yang berbasis Internet Protocol (IP) dan bersifat *connection oriented*. Pada OSI layer TCP berada pada layer transport yang fungsinya mengatur pengiriman suatu data dari client ke server.



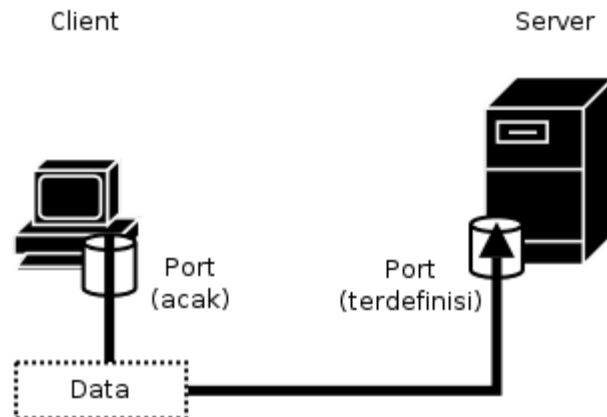
Gb 1. TCP pada OSI Layer

Model komunikasi data dengan client server artinya pada saat pengiriman data, salah satu komputer ada yang bersifat client dan yang satu akan bersifat sebagai server.



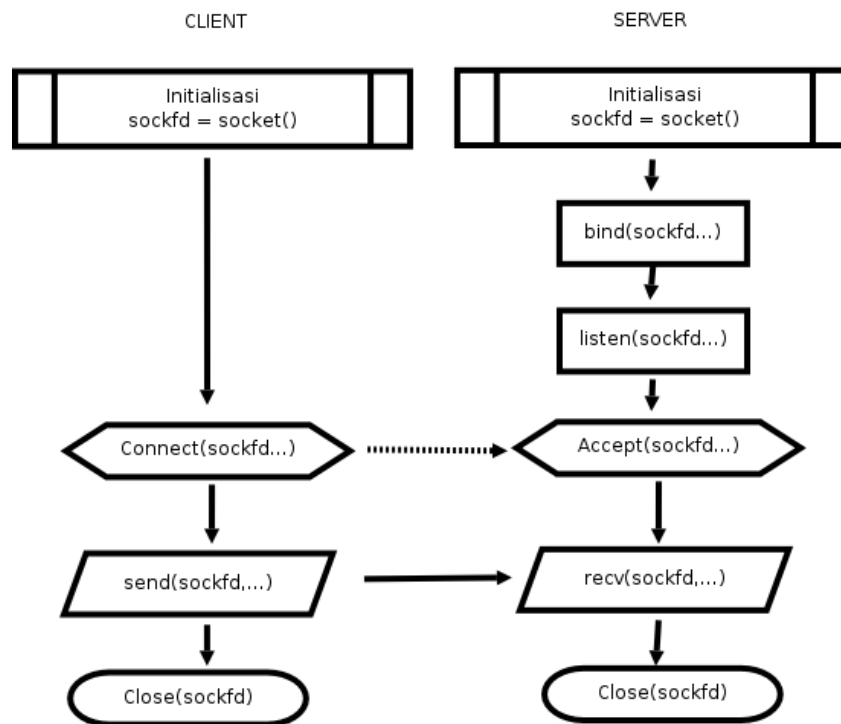
Gb 2. Client Server

Untuk pengiriman datanya, pada masing-masing computer (client-server) akan menggunakan *port* dengan pendefinisian terlebih dahulu. Kemudian dari client akan mengirimkan data dari port pada PCnya ke arah port pada PC servernya. Apabila port tersebut sudah digunakan oleh aplikasi lainnya maka akan terjadi error apabila aplikasi yang kita jalankan menggunakan port yang sama. Jumlah port yang ada 65535 digunakan sesuai dengan aplikasi yang sudah distandarkan.



Gb 3. Pengiriman data melalui PORT

Alur penggunaan socket programming untuk TCP seperti pada Gb. 4.



Gb 4. Alur socket programming pada TCP

3. PERALATAN

- PC (Linux OS)
- GCC
- UTP Cable
- Hub / Switch (optional)

4. LANGKAH PRAKTIKUM

1. Sebelum PC menyala, Sambungkan PC ke jaringan, apabila dihubungkan dengan switch/hub gunakan kabel UTP straight through apabila dengan PC langsung gunakan kabel crossover
2. Nyalakan PC hingga proses booting sempurna.
3. Login dengan user “*root*” dan password “*root*” (isikan tanpa tanda petik)
4. Jalankan perintah *ifconfig eth0* , kemudian catat IP addressnya.

contoh :

```
highway:~# ifconfig eth0
eth0      Link encap:Ethernet HWaddr 00:13:D4:CC:4E:2A
          inet addr:192.168.0.167 Bcast:192.168.0.255 Mask:255.255.255.0
          inet6 addr: fe80::213:d4ff:fecc:4e2a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:5630764 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2730184 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2746138798 (2.5 GiB) TX bytes:404795862 (386.0 MiB)
          Base address:0xdc00 Memory:febc0000febe0000
```

IP address kita adalah 192.168.0.167

• Tips untuk asisten :

Apabila PC belum memiliki IP, asisten diharapkan memberikan IP dengan perintah :

```
# ifconfig eth0 192.168.0.*
```

Dimana * digantikan dengan angka 1 – 254, dengan syarat tidak ada IP yang sama antar PC

5. Buat direktori dengan nama kelas – group, contoh : 2D4TA1

```
# cd
```

```
# mkdir 2D4TA1
```

Tanda # tidak perlu ikut diketik !!!

6. Masuk ke direktori tersebut ...

```
# cd 2D4TA1
```

7. Untuk memulai pengetikan program di linux, peserta menggunakan program “VIM”. Ketikkan source program client.c atau server.c dengan perintah :

```
# vim client.c
```

atau

```
# vim server.c
```

• Tips untuk asisten:

Peserta praktikum dibagi menjadi beberapa kelompok dan ditunjuk supaya ada yang memprogram *client* dan ada yang memprogram *server*

8. Tekan tombol “Ins” / “Insert” pada keyboard untuk memulai pengetikan hingga terlihat indikasi “INSERT” di layar bagian bawah kiri. Ketik sesuai dengan source pada lampiran.
9. Setelah semua source di ketik, simpan source tersebut dengan menekan tombol “ESC”, hingga indikator “INSERT” hilang, dilanjutkan dengan “:wq” (tanpa tanda petik).
10. Lakukan kompilasi program dengan cara :
 Untuk program server.c
 # gcc -o server server.c
 Untuk program client.c
 # gcc -o client client.c
 Apabila terjadi error, lakukan pengecekan dengan membuka file source seperti pada langkah ke7.
11. Jalankan program dengan perintah, sebagai berikut :
 Untuk server :
 # ./server
 Untuk client :
 # ./client 192.168.0.25 “percobaan pesan”
 Dimana 192.168.0.25 adalah IP dari komputer yang melakukan pemrograman *server*. Pesan yang dikirim adalah ***percobaan pesan***. Pada komputer yang menjalankan program server akan tampil data text tersebut.
 * **Tips** : Untuk mematikan program lakukan dengan menekan “Ctrl + C”
12. Lakukan pengiriman text tersebut dengan kondisi sebagai berikut, kemudian amati pada komputer tersebut dan apabila muncul error catat di laporan sementara!
 1. Program server dijalankan di komputer A, pada komputer B kirim pesan dengan program client ke komputer A.
 2. Matikan program server pada komputer A, pada komputer B kirim pesan dengan program client ke komputer A.

5. TUGAS

1. Lampirkan RFC yang berhubungan dengan protokol TCP (RFC 793)

6. REFERENSI

- RFC 793
- man socket, bind, send, recv, ip, accept, listen, connect

LEMBAR ANALISA

Praktikum Komunikasi Data – 5

Tanggal praktikum :

Nama :

NRP :

Kelas :

IP Server :

IP Client :

<i>No</i>	<i>Server</i>	<i>Client</i>	<i>Pesan Error</i>
1.	Dijalankan	Dijalankan	
2.	Dimatikan	Dijalankan	

LAMPIRAN

```
/*
** client.c – a stream socket client for KOMDAT
** by Sritrusta Sukaridhoto, ST
*/

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>

#define PORT 3490 // the port client will be connecting to
#define MAXDATASIZE 100 // max number of bytes we can get at once

int main(int argc, char *argv[])
{
    int sockfd;
    struct hostent *he;
    struct sockaddr_in their_addr; // connector's address information

    if (argc != 3) {
        fprintf(stderr, "Penggunaan: %s server pesan\n", argv[0]);
        exit(1);
    }

    if ((he=gethostbyname(argv[1])) == NULL) { // get the host info
        perror("gethostbyname");
        exit(1);
    }

    if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == 1) {
        perror("socket");
        exit(1);
    }

    their_addr.sin_family = AF_INET; // host byte order
    their_addr.sin_port = htons(PORT); // short, network byte order
    their_addr.sin_addr = *((struct in_addr *)he->h_addr);
    memset(&(their_addr.sin_zero), '\0', 8); // zero the rest of the struct

    if (connect(sockfd, (struct sockaddr *)&their_addr, sizeof(struct sockaddr)) == 1) {
        perror("connect");
        exit(1);
    }

    if ((send(sockfd, argv[2], strlen(argv[2]), 0)) == 1) {
        perror("send");
        exit(0);
    }

    printf("mengirimkan %s ke %s\n", argv[2], argv[1]);
}
```



```

    close(sockfd);

    return 0;
}

/*
** server.c – a stream socket server for KOMDAT
** by Sritrusta Sukaridhoto, ST
*/

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/wait.h>
#include <signal.h>

#define MYPORT 3490                // the port users will be connecting to

#define BACKLOG 10                // how many pending connections queue will hold

#define MAXDATA 100000

void sigchld_handler(int s)
{
    while(wait(NULL) > 0);
}

int main(void)
{
    int sockfd, new_fd;            // listen on sock_fd, new connection on new_fd
    struct sockaddr_in my_addr;    // my address information
    struct sockaddr_in their_addr; // connector's address information
    int sin_size, numbytes;
    char buf[MAXDATA];
    struct sigaction sa;
    int yes=1;

    if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == 1) {
        perror("socket");
        exit(1);
    }

    if (setsockopt(sockfd, SOL_SOCKET, SO_REUSEADDR, &yes, sizeof(int)) == 1) {
        perror("setsockopt");
        exit(1);
    }

    my_addr.sin_family = AF_INET;    // host byte order
    my_addr.sin_port = htons(MYPORT); // short, network byte order
    my_addr.sin_addr.s_addr = INADDR_ANY; // automatically fill with my IP

```

```

memset(&(my_addr.sin_zero), '\0', 8); // zero the rest of the struct

if (bind(sockfd, (struct sockaddr *)&my_addr, sizeof(struct sockaddr)) == 1) {
    perror("bind");
    exit(1);
}

if (listen(sockfd, BACKLOG) == 1) {
    perror("listen");
    exit(1);
}

sa.sa_handler = sigchld_handler; // reap all dead processes
sigemptyset(&sa.sa_mask);
sa.sa_flags = SA_RESTART;

if (sigaction(SIGCHLD, &sa, NULL) == 1) {
    perror("sigaction");
    exit(1);
}

printf("SERVER: siap menerima koneksi\n");
while(1) { // main accept() loop
    sin_size = sizeof(struct sockaddr_in);
    if ((new_fd = accept(sockfd, (struct sockaddr *)&their_addr, &sin_size)) == 1) {
        perror("accept");
        continue;
    }
    printf("SERVER: menerima koneksi dari %s\n", inet_ntoa(their_addr.sin_addr));
    if (!fork()) { // this is the child process
        close(sockfd); // child doesn't need the listener
        numbytes=recv(new_fd, buf, MAXDATA-1, 0);
        if (numbytes < 0) {
            perror("recv");
            exit(1);
        }
        printf("Data: %s\n", buf);

        buf[numbytes]= '\0';

        close(new_fd);
        exit(0);
    }
    close(new_fd); // parent doesn't need this
}

return 0;
}

```

PERCOBAAN VI

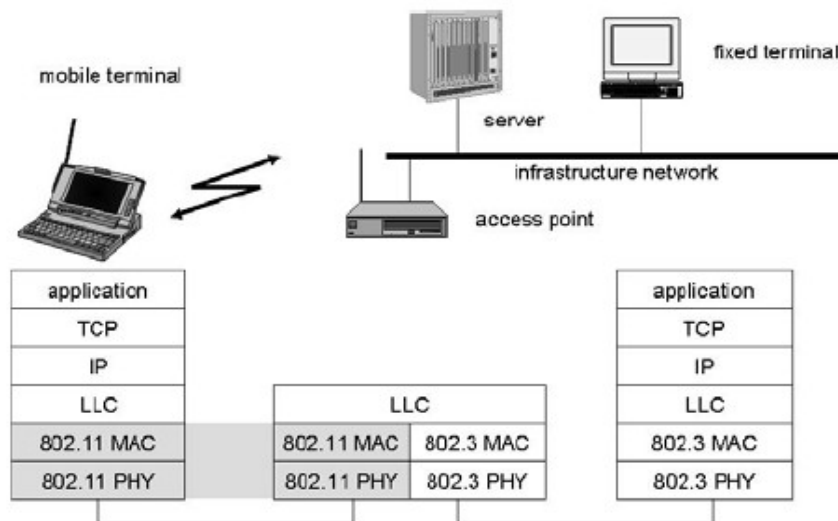
Wireless – LAN(Indoor)

I. TUJUAN

- Mahasiswa dapat mengetahui cara kerja WLAN
- Mahasiswa dapat melakukan konfigurasi WLAN
- Mahasiswa dapat menggunakan aplikasi WLAN

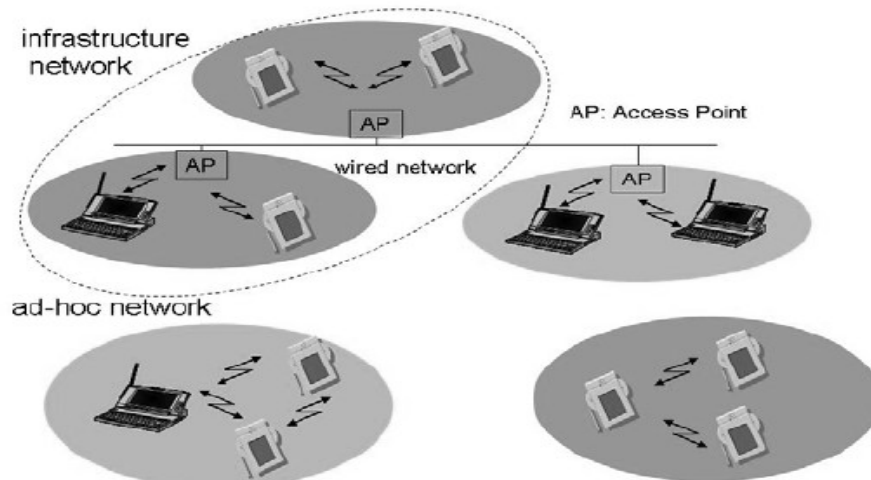
II. DASAR TEORI

Wireless LAN merupakan salah satu cara komunikasi data yang tidak menggunakan penghubung kawat melainkan melewati udara. Standarisasi yang digunakan adalah IEEE 802.11. Standar yang sudah berada di pasar IEEE 802.11a/b/g



Gb 1: OSI Layer dan IEEE802.11

Tipe arsitektur dari Wireless LAN : Infrastrucure vs AdHoc network



Gb 2: Arsitektur WirelessLAN

Ciri dari jaringan infrastruktur pada wifi, terdapat adanya access point (AP)
Logo perangkat yang mendukung WirelessLAN



Gb 3: Logo WIFI

Extended Service Set Identifier (ESSID), nama yang digunakan untuk pengelompokan jaringan wireless dengan model infrastruktur.

Perangkat wireless dibagi menjadi 2 kategori, yaitu access point dan perangkat client.

Access Point adalah perangkat wireless yang berguna untuk menyambungkan jaringan kabel(wired) dan jaringan nirkabel (wireless).



Gb 4: Access Point

Perangkat client adalah perangkat yang digunakan di sisi penerima, misal wireless PCMCIA, wireless PCI, atau wireless USB.

Notebook/Laptop keluaran terbaru sudah terintegrasi dengan perangkat wireless internal.



Untuk menggunakan perangkat WIFI di linux diperlukan paket aplikasi Wireless-tools, Paket aplikasi wireless tools terdiri dari beberapa aplikasi antara lain :

- iwconfig
- iwevent
- iwgetid
- iwlist

- iwpriv
- iwspy

Perintah-perintah tersebut digunakan untuk memodifikasi layer 2 dari OSI.

Untuk memodifikasi layer 3 atau yang biasa disebut IP, digunakan perintah *ifconfig* untuk melakukan perubahan.

Aplikasi Linux yang dapat digunakan untuk memonitoring keadaan jaringan wireless adalah *wavemon*.

III. PERALATAN

1. Access Point
2. Perangkat Client (PCMCIA, PCI, USB)
3. Laptop
4. Software : wireless tools, wavemon, ifconfig, ping
5. PDA (optional)

IV. LANGKAH PRAKTIKUM

1. Physical Layer

1. Pasangkan perangkat wireless client pada komputer. Apabila menggunakan laptop, aktifkan dengan menggeser switch wifi pada posisi “on”.
2. Cek dengan perintah “iwconfig”, apakah perangkat sudah terpasang dengan benar

```
creative:~#iwconfig
lo          nowirelessextensions.
eth0        nowirelessextensions
eth1        IEEE802.11DSESSID:" "Nickname:"PrismI"
            Mode:      Managed      Access      Point:00:00:00:00:00:00
            BitRate:11Mb/sTxPower=15dBm      Sensitivity:1/3
            Retryminlimit:8RTSthr:off      Fragmentthr:off
            Encryptionkey:off PowerManagement:off
sit0        nowirelessextensions.
```

3. Aktifkan perangkat tersebut dengan perintah “*ifconfig eth1 up*”. (Apabila perangkat wireless terdeteksi di-*eth1*, bila lain sesuaikan dengan hasil dari perintah *iwconfig*)

2. Data-link Layer

- 1 Lakukan konfigurasi wireless upaya masuk ke jaringan dengan ESSID “*eepishotspot*” dengan perintah :

```
# iwconfig eth1 essid eepishotspot
```

2. Lakukan konfigurasi wireless supaya jaringan tersebut menggunakan mode :
Mode Infrastructure

```
# iwconfig eth1 mode managed
```

Mode Adhoc

```
# iwconfig eth1 mode adhoc
```

3. Catat pada laporan sementara hasil perubahan dengan perintah “*iwconfig*”.

3. Network Layer

1. Lakukan konfigurasi wireless supaya perangkat tersebut memiliki IP dengan perintah:

ifconfig eth1 192.168.0. **

Tips : ganti * dengan angka 1-254, dan jangan ada yang menggunakan angka yang sama

2. Catat pada laporan sementara hasil perubahan dengan perintah “*ifconfig*”.

4. Transport Layer

1. Lakukan pengecekan jaringan dengan protocol ICMP, dengan perintah “ping” dengan tujuan IP address dari komputer yang lain.

ping 192.168.0.50

***Catatan:** IP 192.168.0.50 adalah IP dari komputer lain.

2. Catat pada laporan sementara hasil dari perintah “*ping*” tersebut.
 3. Lakukan monitoring dengan perintah “*wavemon*”.
5. Lakukan percobaan di atas untuk mode infrastruktur dan adhoc, catat hasilnya pada laporan sementara

V. TUGAS

1. Jelaskan tentang standarisasi wireless IEEE802.11 a/b/g
2. Cari spesifikasi tentang perangkat yang digunakan pada praktikum

VI. REFERENSI

1. IEEE802.11 a/b/g
2. <http://www.dlink.com>
3. man ifconfig, iwconfig, wavemon, ping

LEMBAR ANALISA

Praktikum Komunikasi Data-6

Tanggal praktikum :
Nama :
NRP :
Kelas :

Access Point Spesifikasi

– Merk/Type :
– MAC :
– Serial :

Wireless Client spesifikasi (Isikan sebanyak client yang dipergunakan)

– Merk / Type :
– MAC :
– Serial :

Analisa

1. Mode Infrastructure

1. Physical Layer

iwconfig

#ifconfig

2. DataLink Layer

#iwconfig

3. Network Layer

#ifconfig

4. Transport Layer

#ping

#wavemon

2. Mode Ad-Hoc

1. Physical Layer

iwconfig

#ifconfig

3. DataLink Layer

#iwconfig

5. Network Layer

#ifconfig

6. Transport Layer

#ping

#wavemon

PERCOBAAN VII

Wireless – LAN (Outdoor)

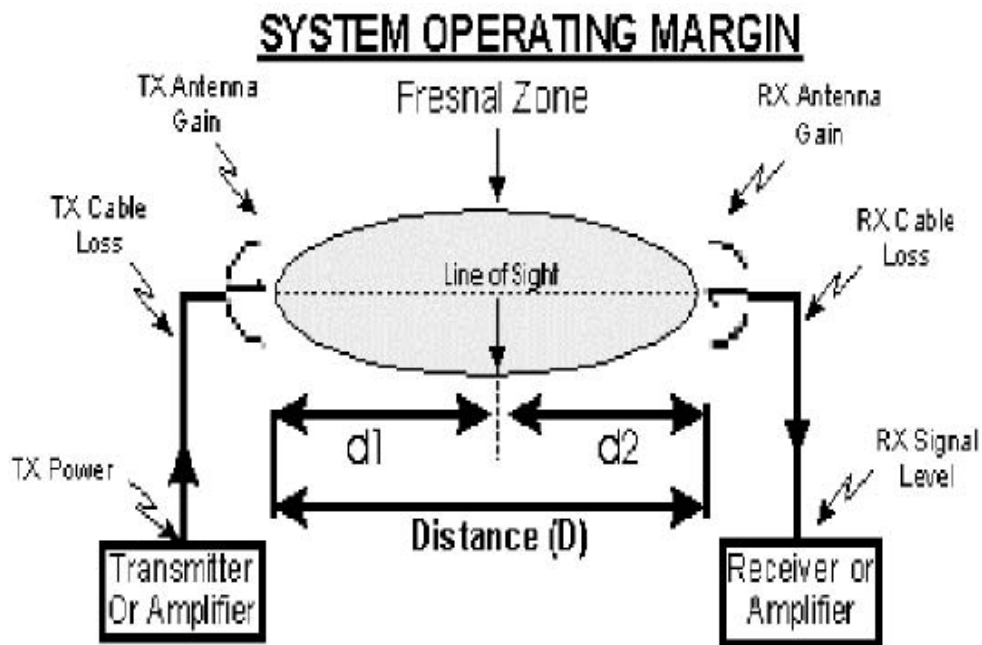
I. TUJUAN

1. Mahasiswa dapat mengetahui cara kerja WLAN
2. Mahasiswa dapat melakukan konfigurasi WLAN Outdoor
3. Mahasiswa dapat menggunakan aplikasi WLAN
4. Mahasiswa dapat mengenal tentang aplikasi HOTSPOT

II. DASAR TEORI

Faktor-faktor yang perlu diperhatikan dalam membangun Wireless Outdoor adalah seperti berikut :

System Operating Margin



Gb 1: Aturan SOM

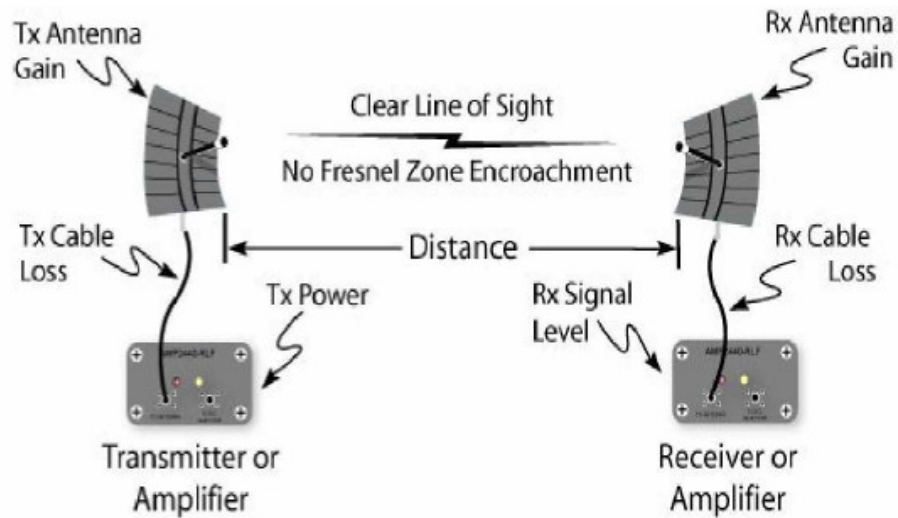
Dimana untuk konversinya :

- $\text{dBm} = (10\log_{10}(\text{Power Watts})) + 30$
- $\text{Watts} = 10^{((\text{dBm} - 30)/10)}$
- $\text{MilliWatts} = 10^{(\text{dBm}/10)}$

Dengan perumusan SOM:

- $SOM = RX \text{ Signal Level} - RX \text{ Sensitivity}$
- $RX \text{ Signal Level} = TX \text{ Power} - TX \text{ Cable Loss} + TX \text{ Antenna Gain} - FSL + RX \text{ Antenna Gain} - RX \text{ Cable Loss}$

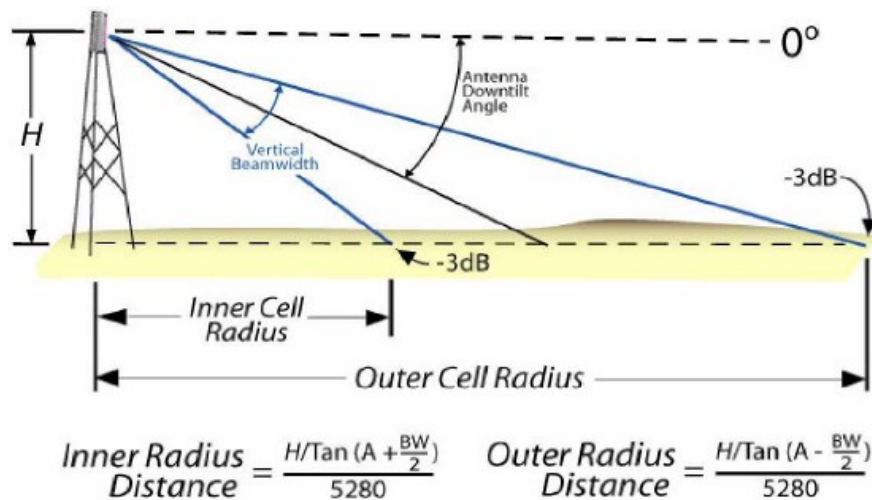
Dimana untuk perumusan Free Space Loss (FSL) :



$$Free \ Space \ Loss = 20\log_{10}(MHz) + 20\log_{10}(Distance \ in \ Miles) + 36.6$$

Gb 2: FSL

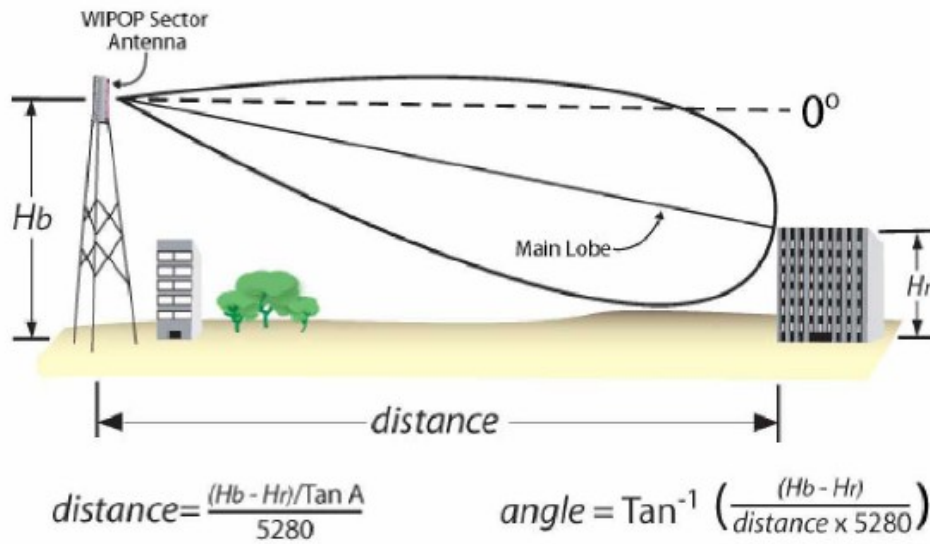
Downtilt Coverage Radius, daerah jangkuan yang bias tercover dari BTS yang kita bangun dengan memperhatikan parameter dari kemiringan antenna, propagasi dari antenna dan ketinggian tiang dari antenna tersebut.



$$Inner \ Radius \ Distance = \frac{H/Tan(A + \frac{BW}{2})}{5280} \quad Outer \ Radius \ Distance = \frac{H/Tan(A - \frac{BW}{2})}{5280}$$

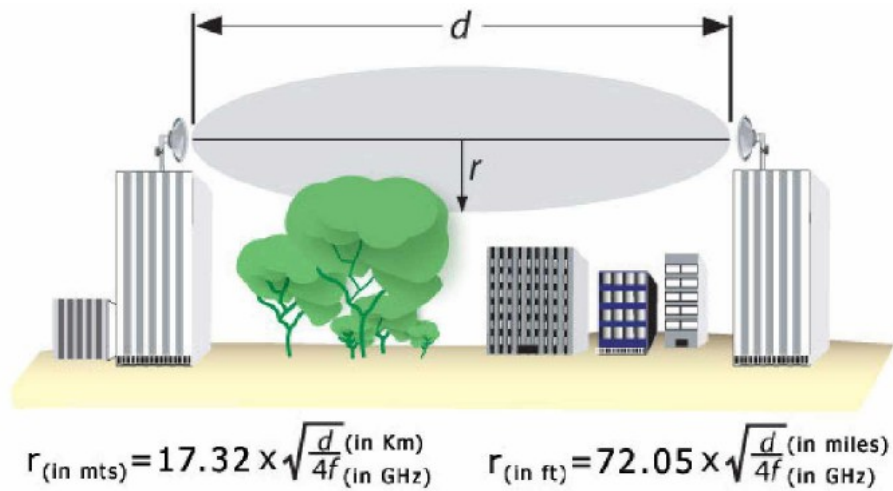
Gb 3: Downtilt Coverage Radius

Downtilt Antenna, kemiringan antenna yang dapat mempengaruhi jarak dan target coverage



Gb 4: Downtilt Antenna

Fresnel Zone, Daerah yang visualisasi dari hasil penyebaran line of sight dimana signal telah keluar dari antenna.



Gb 5: Fresnel Zone

Perangkat Wireless Outdoor antara lain :



Gb 6: Contoh Perangkat Wireless

Antenna



Gb 7: OmniAntenna



Gb 8: DirectGrid Antenna



Gb 9: PanelAntenna

LMR Cable



Gb 10: LMR cable

Pig-tail cable



Gb 11: Pig-tail Cable

Amplifier



Gb 12: Amplifier

Lightning Protector



Gb 13: Lightning Protector

PowerOverEthernet (PoE)



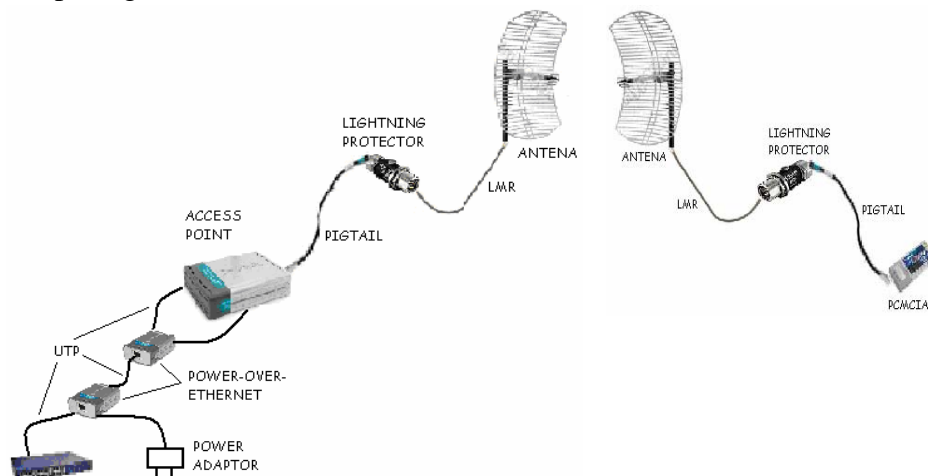
Gb 14: PoE

III. PERALATAN

1. Access Point
2. Perangkat Client (PCMCIA, PCI, USB)
3. Laptop
4. Software : wireless tools, wavemon, ifconfig, ping
5. PDA
6. Antenna
7. Lightning Protector
8. PoE
9. Amplifier
10. LMR Cable
11. Pigtail cable

IV. LANGKAH PRAKTIKUM

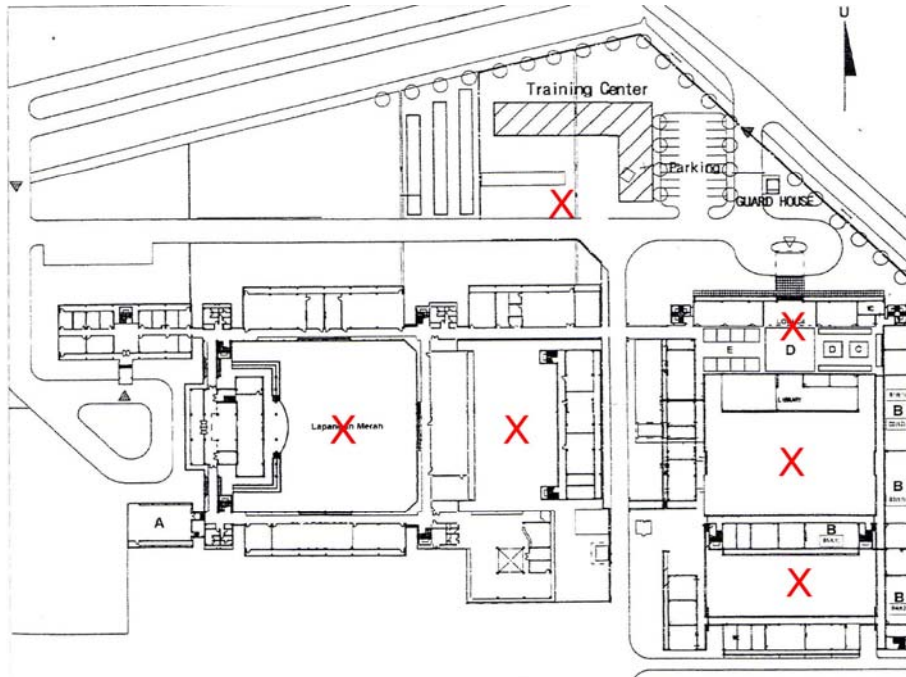
1. Point to Point Wireless
 1. Pasangkan perangkat wireless client pada computer sesuai dengan diagram pada gb 15



2. Perangkat sebelah kiri berada di gedung D3 dan perangkat sebelah kanan berada di gedung D4.
3. Cek dengan aplikasi “wavemon”, dan catat hasilnya pada laporan sementara.

2. Hotspot Wireless

1. Gunakan aplikasi “iwlisteth1 scanning” pada laptop dan aplikasi “wifigraph” pada PDA.
2. Lakukan scanning di daerah X sesuai dengan denah gb. 16.



3. Catat hasil dari aplikasiaplikasi tersebut pada laporan sementara.

V. TUGAS

1. Cari spesifikasi / data sheet tentang perangkat yang digunakan pada praktikum

VI. REFERENSI

1. IEEE802.11 a/b/g
2. <http://www.dlink.com>
3. <http://www.hyperlinktech.com>
4. man ifconfig, iwconfig, wavemon, ping

LEMBAR ANALISA

Praktikum Komunikasi Data – 7

Tanggal praktikum :

Nama :

NRP :

Kelas :

1. Pointtopoint

1. Perangkat

-
-
-
-

2. Wavemon

2. Hotspot

1. Wifigraph

2. # iwlist eth1 scanning