

**MODUL PRAKTIKUM  
TEKNIK TELEKOMUNIKASI I**



**TIM DOSEN TEKNIK TELEKOMUNIKASI**

**PROGRAM STUDI TEKNIK ELEKTRO  
DEPARTEMEN PENDIDIKAN TEKNIK ELEKTRO  
FAKULTAS PENDIDIKAN TEKNOLOGI DAN KEJURUAN  
UNIVERSITAS PENDIDIKAN INDONESIA  
BANDUNG  
2019**

## DAFTAR ISI

DAFTAR ISI .....	2
PERCOBAAN 1. RANGKAIAN PENGUAT TRANSISTOR .....	3
PERCOBAAN 2. FILTER ANALOG FREKUENSI TINGGI.....	8
PERCOBAAN 3. AMPLITUDO MODULATION .....	17
PERCOBAAN 4. FREQUENCY MODULATION.....	20
PERCOBAAN 5. SINYAL .....	24
PERCOBAAN 6. SISTEM.....	36
PERCOBAAN 7. PLANE WAVE – NORMAL INCIDENCE (AIR-DIELECTRIC-AIR)....	52
PERCOBAAN 8. QUARTER WAVE TRANSFORMER .....	59
PERCOBAAN 9. WAVEGUIDES – RECTANGULAR WAVEGUIDE .....	68
PERCOBAAN 10. INSTALASI JARINGAN LAN.....	73
PERCOBAAN 11. INSTALASI JARINGAN WLAN DENGAN CISCO PACKET TRACER .....	77
PERCOBAAN 12. ARDUINO .....	88

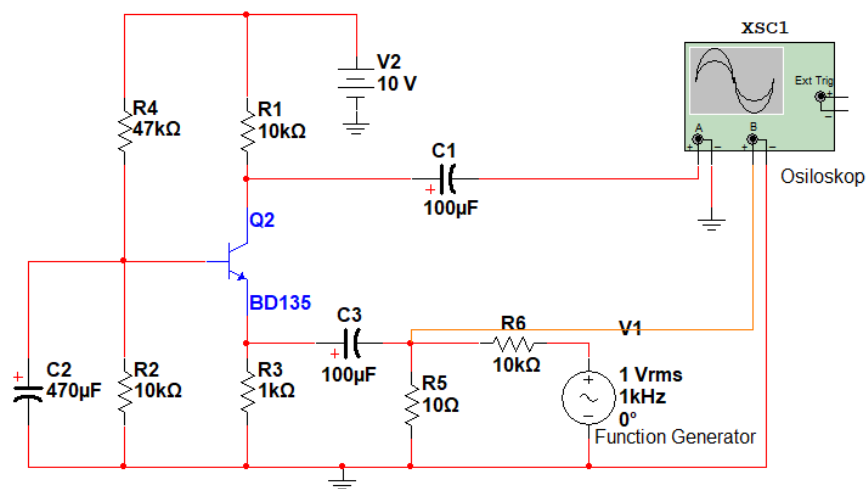
# PERCOBAAN 1. RANGKAIAN PENGUAT TRANSISTOR

## RANGKAIAN COMMON BASE

### A. Tujuan Percobaan

1. Mengukur penguatan tegangan.
2. Mengukur penguatan arus.
3. Mengukur penguatan daya
4. Mengukur beda fasa

### B. Rangkaian



### C. Alat dan Bahan

- |                             |        |
|-----------------------------|--------|
| 1. Transistor BD 130/BD 135 | 1 buah |
| 2. Resistor:                |        |
| a. 1k ohm                   | 2 buah |
| b. 10k ohm                  | 3 buah |
| c. 47k ohm                  | 1 buah |
| d. 10 ohm                   | 1 buah |
| 3. Kapasitor:               |        |
| a. 470 uF                   | 1 buah |
| b. 100 uF                   | 2 buah |
| 4. Power Supply             | 1 buah |
| 5. Function Generator       | 1 buah |
| 6. Osiloskop                | 1 buah |
| 7. Multimeter               | 1 buah |

#### D. Langkah Percobaan

1. Rangkai sirkuit seperti pada gambar di atas.
2. Hubungkan dengan osiloskop.  
Channel a = input  
Channel b = output
3. Pasang Function Generator dan set frekuensi pada 1kHz.
4. Ukur tegangan dan beda fasa input dan output menggunakan osiloskop.
5. Ukur arus pada input dan output menggunakan multimeter dan hitung dayanya.

#### E. Tugas

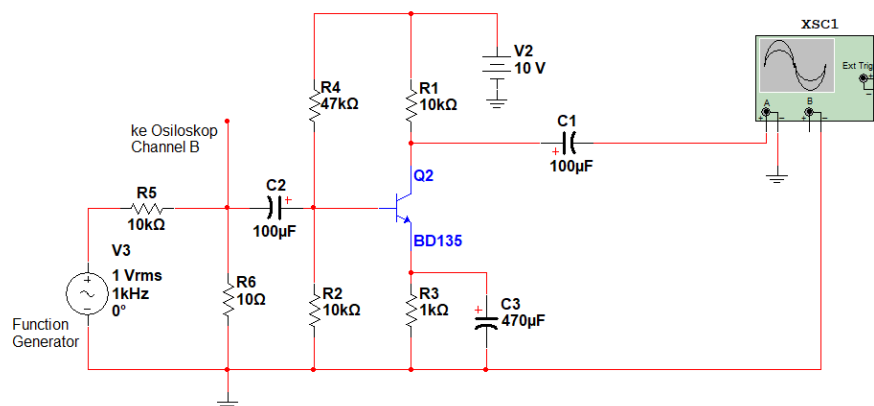
1. Berapa penguatan tegangannya?
2. Berapa penguatan arusnya?
3. Berapa penguatan dayanya?
4. Berapa beda fasanya?

### RANGKAIAN COMMON EMITTER

#### A. Tujuan Percobaan

1. Mengukur penguatan tegangan.
2. Mengukur penguatan arus.
3. Mengukur penguatan daya.
4. Mengukur beda fasa antara input dan output.

#### B. Rangkaian



#### C. Alat dan Bahan

1. Transistor BD 130/BD 135                      1 buah

2. Resistor:
  - a. 1k ohm 1 buah
  - b. 10k ohm 3 buah
  - c. 10 ohm 1 buah
  - d. 47k ohm 1 buah
3. Kapasitor:
  - a. 470 uF 1 buah
  - b. 100 uF 2 buah
4. Power Supply 1 buah
5. Function Generator 1 buah
6. Osiloskop 1 buah
7. Multimeter 1 buah

#### **D. Langkah Percobaan**

1. Rangkai sirkuit seperti pada gambar di atas.
2. Hubungkan dengan osiloskop.  
Channel a = input  
Channel b = output
3. Pasang Function Generator dan set frekuensi pada 1kHz.
4. Ukur tegangan dan beda fasa input dan output menggunakan osiloskop.
5. Ukur arus pada input dan output menggunakan multimeter dan hitung dayanya.

#### **E. Tugas**

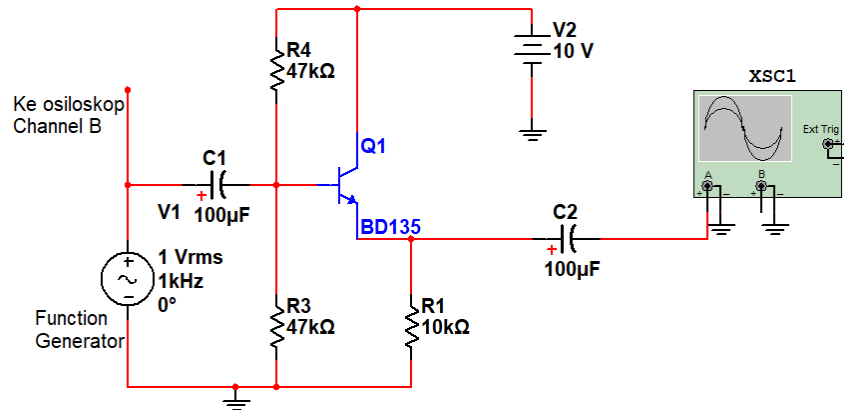
1. Berapa penguatan tegangannya?
2. Berapa penguatan arusnya?
3. Berapa penguatan dayanya?
4. Berapa beda fasanya?

### **RANGKAIAN COMMON COLLECTOR**

#### **A. Tujuan Percobaan**

1. Mengukur penguatan tegangan.
2. Mengukur penguatan arus.
3. Mengukur penguatan daya.
4. Mengukur beda fasa antara input dan output.

## B. Rangkaian



## C. Alat dan Bahan

- |                       |        |
|-----------------------|--------|
| 1. Transistor BD 130  | 1 buah |
| 2. Resistor:          |        |
| a. 10k                | 1 buah |
| b. 47k                | 2 buah |
| 3. Kapasitor 100uF    | 2 buah |
| 4. Power Supply       | 1 buah |
| 5. Function Generator | 1 buah |
| 6. Osiloskop          | 1 buah |
| 7. Multimeter         | 1 buah |

## D. Langkah Percobaan

1. Rangkai sirkuit seperti pada gambar di atas.
2. Hubungkan dengan osiloskop.  
Channel a = input  
Channel b = output
3. Pasang Function Generator dan set frekuensi pada 1kHz.
4. Ukur tegangan dan beda fasa input dan output menggunakan osiloskop.
5. Ukur arus pada input dan output menggunakan multimeter dan hitung dayanya.

## E. Tugas

1. Berapa penguatan tegangannya?
2. Berapa penguatan arusnya?
3. Berapa penguatan dayanya?

4. Berapa beda fasanya?
5. Bandingkan hasil percobaan common base, common emitter, dan common collector dalam satu tabel.

	<b>V (Tegangan)</b>			<b>I (Arus)</b>			<b>P (Daya)</b>			<b>Beda Fasa</b>
	<b>in</b>	<b>out</b>	<b>gain</b>	<b>In</b>	<b>out</b>	<b>gain</b>	<b>in</b>	<b>out</b>	<b>gain</b>	
<b>CB (Common Base)</b>										
<b>CE (Common Emitter)</b>										
<b>CC (Common Collector)</b>										

## PERCOBAAN 2. FILTER ANALOG FREKUENSI TINGGI

### A. Tujuan Percobaan

1. Mahasiswa mampu memahami prinsip kerja filter analog untuk aplikasi system komunikasi.
2. Mahasiswa mampu menentukan jenis filter analog yang sesuai untuk tujuan tertentu pada system komunikasi.
3. Mahasiswa mampu membuat grafik tanggapan frekuensi dari sebuah filter analog.

### B. Alat dan Bahan

1. 1 set Osiloskop lengkap dengan konektornya.
2. Function Generator
3. Kapasitor
  - a. 500 pF 2 buah
  - b. 10 nF 1 buah
  - c. 1 uF 1 buah
4. Resistor
  - a. 20  $\Omega$  1 buah
  - b. 10 K $\Omega$  1 buah
  - c. 1 K $\Omega$  1 buah
5. Induktor 140  $\mu$ H 2 buah
6. Multimeter 1 buah
7. Jumper
8. Project Board 1 buah

### C. Teori Dasar

Dikatakan filter aktif karena selain menggunakan beberapa resistor dan kapasitor juga menggunakan beberapa komponen aktif seperti OpAmp, dengan penguatan yang bisa diatur sesuai dengan yang kita inginkan. Besarnya nilai tanggapan biasa dinyatakan dalam volt ataupun dalam dB dengan bentuk respon yang berbeda pada setiap jenis filter. Besar nilai respon dapat diperoleh dari perhitungan fungsi alih:

$$H_s = \frac{|V_{out}|}{|V_{in}|}$$

Dengan:  $H_s$  = Fungsi alih

$V_{out}$  = Tegang keluran



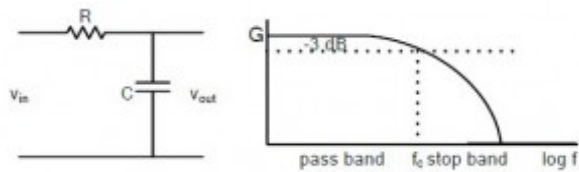
$V_{in}$  = Tegangan masukan

Setiap filter mempunyai frekuensi cutoff yaitu frekuensi di 0,707 atau -3dB.

**Ada 4 jenis filter berdasarkan frekuensi kerjanya, yaitu:**

### 1. Low Pass Filter

Low Pass Filter (LPF) atau Filter Lolos Bawah adalah filter yang hanya melewatkan sinyal dengan frekuensi yang lebih rendah dari frekuensi cut-off ( $f_c$ ) dan akan melemahkan sinyal dengan frekuensi yang lebih tinggi dari frekuensi cut-off ( $f_c$ ). Pada filter LPF yang ideal sinyal dengan frekuensi diatas frekuensi cut-off ( $f_c$ ) tidak akan dilewatkan sama sekali (tegangan output = 0 volt). Rangkaian low pass filter RC merupakan jenis filter pasif, dengan respon frekuensi yang ditentukan oleh konfigurasi R dan C yang digunakan. Rangkaian dasar LPF dan grafik respon frekuensi LPF dapat dilihat pada gambar 1.



Gambar 1. Rangkaian Dasar dan Respon Frekuensi LPF

Frekuensi cut-off ( $f_c$ ) dari filter pasif lolos bawah (Low Pass Filter, LPF) dengan RC dapat dituliskan dalam persamaan matematik sebagai berikut.

$$f_c = \frac{1}{2\pi RC}$$

Rangkaian filter pasif LPF RC diatas terlihat seperti pembagi tegangan menggunakan R. Dimana pada filter LPF RC ini teganga output diambil pada titik pertemuan RC. Tegangan output ( $V_{out}$ ) filter pasif LPF seperti terlihat pada rangkaian diatas dapat diekspresikan dalam persamaan matematis sebagai berikut.

$$V_{out} = \frac{1/j\omega C}{1/(j\omega C + R)} \cdot V_{in}$$

Besarnya penguatan tegangan ( $G$ ) pada filter pasif yang ideal maksimum adalah  $1 = 0\text{dB}$  yang hanya terjadi pada frekuensi sinyal input dibawah frekuensi cut-off ( $f_c$ ). Penguatabn tegangan ( $G$ ) filter LPF RC pasif dapat dituliskan dalam persamaan matematis sebagai berikut.

$$G = \left| \frac{V_{out}}{V_{in}} \right|$$

Dan penguatan tegangan ( $G$ ) LPF RC dapat dituliskan dalam satuan dB sebagai berikut.

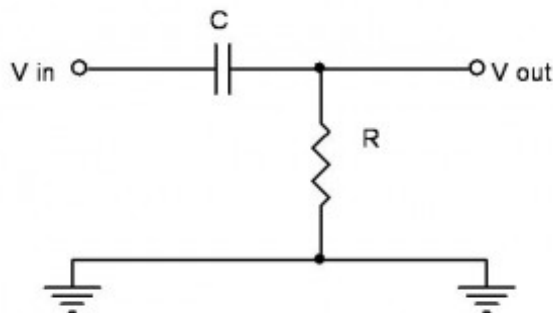
$$G = 20 \log \left( \frac{V_{out}}{V_{in}} \right) = 20 \log \left( \frac{1}{1 + \omega^2 R^2 C^2} \right)$$

Pada filter lolos bawah (low pass filter, LPF) terdapat beberapa karakteristik mendasar sebagai berikut.

- Pada saat frekuensi sinyal input lebih rendah dari frekuensi cut-off ( $f_c$ ) ( $f_{in} \ll f_c$ ) maka penguatan tegangan / Gain ( $G$ ) = 1 atau  $G=0\text{dB}$ .
- Pada saat frekuensi sinyal input sama dengan frekuensi cut-off ( $f_c$ ) ( $f_{in} = f_c$ ) maka  $\omega = 1/RC$  sehingga penguatan tegangan / Gain ( $G$ ) menjadi  $-3 \text{ dB}$  atau terjadi pelemahan tegangan sebesar  $3 \text{ dB}$ .
- Pada saat frekuensi sinyal input lebih tinggi dari frekuensi cut-off ( $f_c$ ) ( $f_{in} \gg f_c$ ) maka besarnya penguatan tegangan ( $G$ ) =  $1/\omega RC$  atau  $G = -20 \log \omega RC$ .
- Sehingga dapat ditarik kesimpulan bahwa Filter Lolos Rendah (Low Pass Filter, LPF) hanya meloloskan sinyal dengan frekuensi yang lebih rendah dari frekuensi cut-off ( $f_c$ ) saja.

## 2. High Pass Filter

Filter high-pass atau sering juga disebut dengan filter lolos atas adalah suatu rangkaian yang akan melewatkan suatu isyarat yang berada diatas frekuensi cut-off ( $\omega_c$ ) sampai frekuensi cut-off ( $\omega_c$ ) rangkaian tersebut dan akan menahan isyarat yang berfrekuensi dibawah frekuensi cut-off ( $\omega_c$ ) rangkaian tersebut. Filter high-passs dasar disusun dengan rangkaian RC seperti pada gambar 2.



Gambar 2. Contoh Rangkaian High Pass Filter

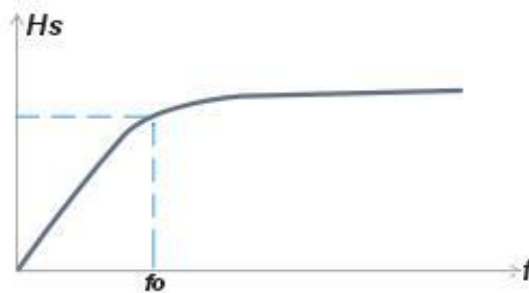
Prinsip kerja dari filter high pass atau filter lolos atas adalah dengan memanfaatkan karakteristik dasar komponen C dan R, dimana C akan mudah melewatkan sinyal AC sesuai dengan nilai reaktansi kapasitifnya dan komponen R yang lebih mudah melewatkan sinyal dengan frekuensi yang rendah. Prinsip kerja rangkaian filter lolos atas atau high pass filter (HPF) dengan RC dapat diuraikan sebagai berikut, apabila rangkaian filter high pass ini diberikan sinyal input dengan

frekuensi diatas frekuensi cut-off ( $\omega_c$ ) maka sinyal tersebut akan di lewatkan ke output rangkaian melalui komponen C. Kemudian pada saat sinyal input yang diberikan ke rangkaian filter lolos atas atau high pass filter memiliki frekuensi di bawah frekuensi cut-off ( $\omega_c$ ) maka sinyal input tersebut akan dilemahkan dengan cara dibuang ke ground melalui komponen R.

Frekuensi resonansi dari filter high-pass mengikuti nilai time constant ( $\tau$ ) dari rangkaian RC tersebut. Sehingga frekuensi cut-off dari filter tersebut adalah:

$$f_c = \frac{1}{2\pi RC}$$

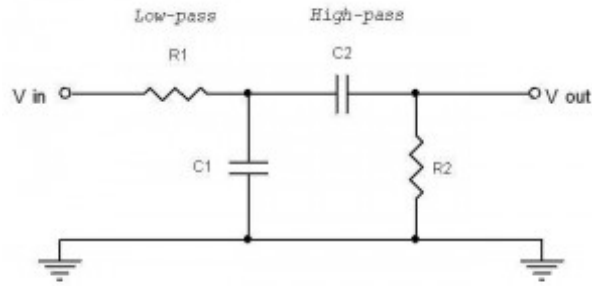
Grafik karakteristik dari high pass filter (HPF) atau filter lolos atas dengan komponen RC dapat digambarkan dengan perbandingan antara tegangan output filter terhadap frekuensi yang diberikan kepada rangkaian filter high pass (HPF) tersebut. Untuk lebih jelasnya grafik karakteristik filter high pass (HPF) ditunjukkan pada gambar 3.



Gambar 3. Respon Frekuensi High Pass Filter

### 3. Band Pass Filter

Filter band-pass adalah sebuah rangkaian yang dirancang hanya untuk melewatkan isyarat dalam suatu pita frekuensi tertentu dan untuk menahan isyarat diluar jalur pita frekuensi tersebut. Jenis filter ini memiliki tegangan keluaran maksimum pada satu frekuensi tertentu yang disebut dengan frekuensi resonansi ( $\omega_r$ ) Jika frekuensinya berubah dari frekuensi resonansi maka tegangan keluarannya turun, ada satu frekuensi diatas frekuensi resonansi ( $\omega_r$ ) dan satu dibawah ( $\omega_r$ ) dimana gainnya tetap 0,707 Ar. Frekuensi ini diberi tanda ( $\omega_h$ ) frekuensi cutoff atas dan ( $\omega_l$ ) frekuensi cutoff bawah. Pita frekuensi antara ( $\omega_h$ ) dan ( $\omega_l$ ) adalah band width (B). Contoh Rangkaian Band Pass Filter dapat dilihat pada gambar 4.



Gambar 4. Rangkaian Band Pass Filter

Nilai frekuensi cut-off pertama ditentukan oleh filter high-pass sebagai berikut:

$$f_{c1} = \frac{1}{2\pi R_2 C_2}$$

dan frekuensi cut-off kedua ditentukan oleh filter low-pass sebagai berikut:

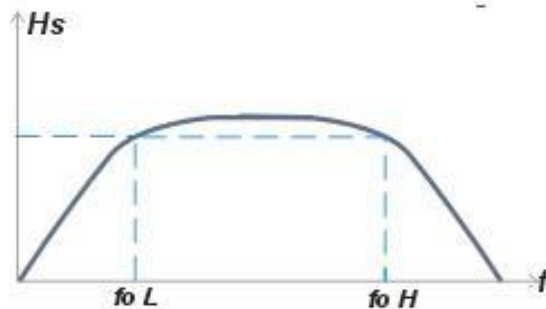
$$f_{c2} = \frac{1}{2\pi R_1 C_1}$$

Jika salah satu  $f_c$  lebih besar dari pada yang lain maka  $f_c$  tersebut merupakan frekuensi cut-off atas ( $f_{ch}$ ) dan untuk yang lainnya menjadi frekuensi cut-off bawah ( $f_{cl}$ )

sehingga besarnya bandwidth adalah:

$$B = f_{ch} - f_{cl}$$

Grafik respon frekuensi Band Pass Filter dapat dilihat pada gambar 5.

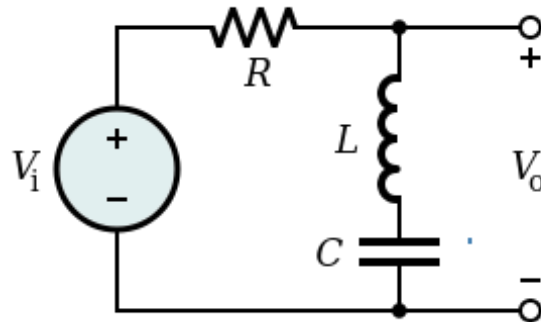


Gambar 5. Respon Frekuensi Band Pass Filter

#### 4. Band Stop Filter

Band stop filter (BPF), band elimination filter, band reject filter dan sering juga disebut dengan notch filter atau filter tolak jalur memiliki pengertian yang sama sebagai filter yang memiliki karakteristik akan menahan sinyal dengan frekuensi sesuai frekuensi cut-off rangkaian dan akan melewatkan sinyal dengan frekuensi di luar frekuensi cut-off rangkaian filter tersebut baik dibawah atau diatas frekuensi cut-

off rangkaian filter. Band stop filter merupakan kebalikan dari band pass filter. Seperti pada filter band-pass, filter band-elimination atau band stop filter (BPF) RC juga disusun dari dua buah filter low-pass dan filter high-pass yang disusun secara paralel seperti terlihat pada gambar berikut. Rangkaian band stop filter (BPF) ini merupakan contoh sederhana dari filter pasif band stop seperti tampak pada gambar 6.



Gambar 6. Rangkaian Band Stop Filter

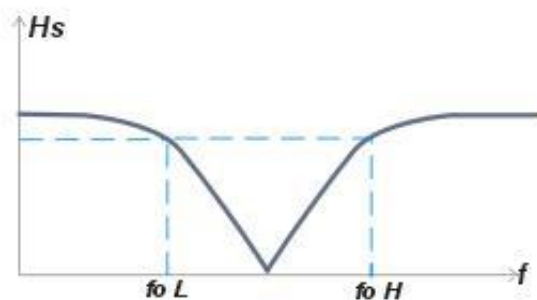
Dari rangkaian itu dapat dicari frekuensi cut off atas:

$$f_{ch} = \frac{R}{2\pi L}$$

Dan frekuensi cut off bawah:

$$f_{cL} = \frac{1}{2\pi RC}$$

Karakteristik dari filter band stop atau filter band elimination ini dapat dilihat pada gambar 7.

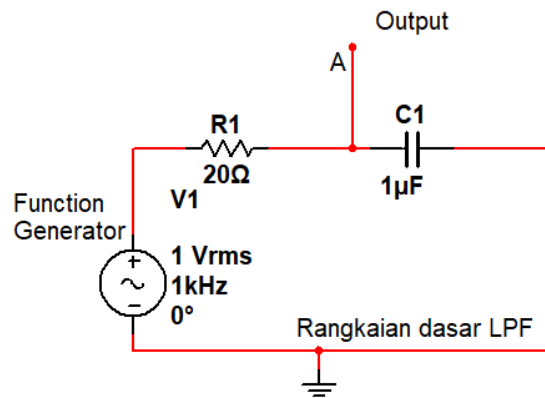


Gambar 7. Respon Frekuensi Band Stop Filter

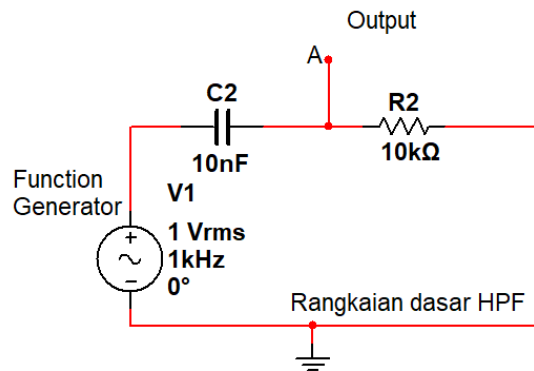
Dari grafik karakteristik dari band stop filter diatas terlihat bahwa tegangan dari sinyal input pada frekuensi cut-off rangkaian filter akan dilemahkan dari level aslinya dan sinyal dengan frekuensi di luar frekuensi cut-off baik diatas atau dibawah frekuensi cut-off akan dilewakan ke output rangkaian filter band sop (BPF) RC tersebut.

#### D. Rangkain

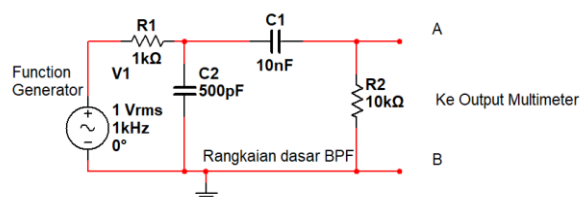
## 1. Low Pass Filter



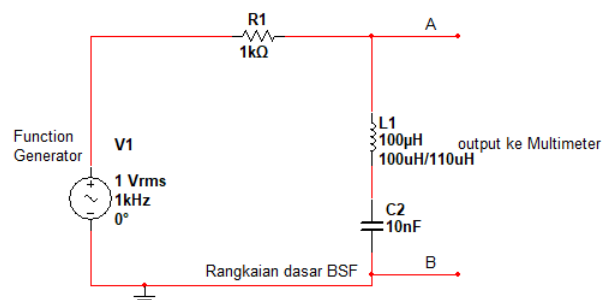
## 2. High Pass Filter



## 3. Band Pass Filter



## 4. Band Stop Filter



### E. Langkah Percobaan

1. Rangkailah rangkaiannya sesuai dengan rangkaian yang digambar
2. Pasang probe osiloskop pada sumber (sebagai **input**) dan pada **output**
3. Ubah-ubah nilai frekuensi pada Function Generator lalu catat tegangannya
4. Cari frekuensi cut-offnya baik menggunakan rumus, dan dengan pengukuran
5. Bandingkan antara perhitungan dengan pengukuran

### F. Table Hasil Praktikum

#### 1. Low Pass Filter

Frekuensi (Hz)	1K	100 K	200 K	300 K	350 K	500 K	1 M
Input							
Output							

Cut-off (pengukuran)?

Cut-off (perhitungan)?

#### 2. High Pass Filter

Frekuensi (Hz)	1K	100 K	200 K	300 K	350 K	500 K	1 M
Input							
Output							

Cut-off (pengukuran)?

Cut-off (perhitungan)?

#### 3. Band Pass Filter

Frekuensi (Hz)	1K	100 K	200 K	300 K	350 K	500 K	1 M
Input							
Output							

Cut-off (pengukuran)?

Cut-off (perhitungan)?

#### 4. Band Stop Filter

Frekuensi	1K	100 K	200 K	300 K	350 K	500 K	1 M
-----------	----	-------	-------	-------	-------	-------	-----

<b>(Hz)</b>							
<b>Input</b>							
<b>Output</b>							

**Cut-off (pengukuran)?**

**Cut-off (perhitungan)?**

### **G. Tugas**

1. Buat grafik tegangan terhadap frekuensi dari data yang telah di dapat?



## PERCOBAAN 3. AMPLITUDO MODULATION

### A. Tujuan Percobaan

1. Mahasiswa dapat memahami bagaimana cara kerja Modulasi Amplitudo.
2. Mahasiswa dapat memahami cara menggunakan software SistemVue.
3. Mahasiswa dapat menggunakan software SistemVue untuk mensimulasikan Modulasi Amplitudo.

### B. Alat dan Bahan

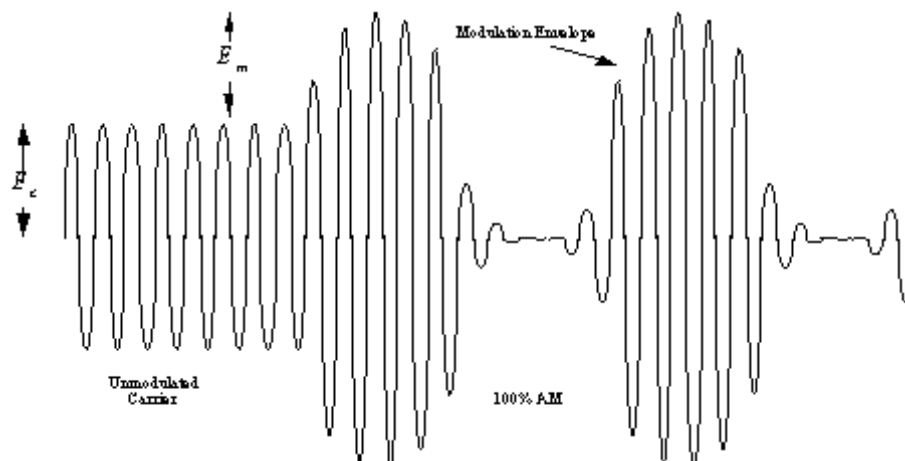
1. PC 1 buah
2. Software SystemVue 1 buah

### C. Teori Dasar

Modulasi amplitudo digunakan dalam radio komersial dan siaran televisi. Meskipun secara tradisional digunakan dalam sistem analog, proses ini dapat disesuaikan untuk membawa informasi digital.

Persamaan AM:  $e_{am} = (1 + \cos\omega_m)\cos\omega_c$

Sebuah sinyal pembawa sinewave adalah bentuk  $e_c = E_c \sin\omega_c t$  dan modulasi sinyal sinewave adalah dalam bentuk  $e_m = E_m \sin\omega_m t$ . Perhatikan bahwa amplitudo pembawa frekuensi tinggi mengambil bentuk yang lebih rendah sinyal modulasi frekuensi membentuk apa yang disebut amplop modulasi.



Indeks modulasi didefinisikan sebagai rasio amplitudo sinyal modulasi amplitudo sinyal pembawa.  $m = \frac{E_m}{E_c}$  dimana  $0 \leq m \leq 1$ . Keseluruhan sinyal dapat dijelaskan oleh:

$$\begin{aligned} e_{am} &= (E_c + E_m \sin \omega_m t) \sin \omega_c t \\ &= (E_c + mE_c \sin \omega_m t) \sin \omega_c t \end{aligned}$$

Akibatnya, memperluas seketika hasil ekspresi di AM:

$$\begin{aligned}
 e_{am} &= E_c \sin \omega_c t + mE_c \sin \omega_m t \sin \omega_c t \\
 &= \underbrace{E_c \sin \omega_c t}_{\text{Carrier}} + \underbrace{\frac{mE_c}{2} \sin(\omega_c - \omega_m)}_{\text{LSB}} - \underbrace{\frac{mE_c}{2} \sin(\omega_c + \omega_m)}_{\text{USB}}
 \end{aligned}$$

Dari sini kita amati bahwa sidebands atas dan bawah dibuat ketika menggunakan

modulasi amplitudo. Amplitudo sideband:  $\frac{mE_c}{2}$ , dan total spektrum diduduki adalah dua kali bandwidth sinyal modulasi atau  $2f_m$ . Seringkali, amplitudo pembawa dinormalisasi dan ekspresi tertulis:

$$e_{am} = (1 + m \sin \omega_m t) \sin \omega_c t$$

Sinyal AM sering dicirikan dalam hal kekuasaan, karena kekuasaan, yang digunakan untuk menggerakkan antena. Daya total dalam  $1 \Omega$  resistor diberikan oleh:

$$\begin{aligned}
 P_T &= E_c^2 + \left(\frac{mE_c}{2}\right)^2 + \left(\frac{mE_c}{2}\right)^2 \\
 &= P_c + \frac{m^2}{4} P_c + \frac{m^2}{4} P_c \\
 &= P_c \left(1 + \frac{m^2}{2}\right)
 \end{aligned}$$

Dari sini kita amati bahwa dengan indeks modulasi 0, daya yang ditransmisikan sama dengan daya pembawa. Namun, ketika indeks modulasi 1, daya yang ditransmisikan total meningkat 1,5 kali daya pembawa. Pada 100% modulasi, hanya 1/3 dari total daya dalam sidebands atau hanya 1/2 dari daya pembawa dalam sidebands.

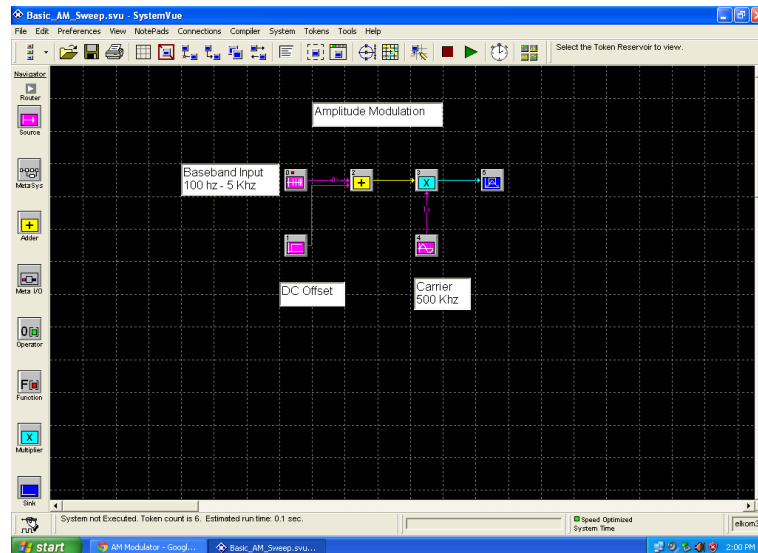
Dalam hal tegangan dan arus:

$$E_T = E_c \sqrt{1 + \frac{m^2}{2}} \quad I_T = I_c \sqrt{1 + \frac{m^2}{2}}$$

Jika pembawa dimodulasi oleh sinyal yang kompleks, modulasi efektif dapat ditentukan oleh kombinasi indeks modulasi masing-masing komponen.

$$m_{eff} = \sqrt{m_1^2 + m_2^2 + m_3^2 + \dots} \text{ (must not exceed 1)}$$

#### D. Rangkaian



### E. Langkah Percobaan

1. Rangkaian pada software seperti pada rangkaian di atas (dengan  $V$  baseband = 1 Volt; dan  $V$  Carrier = 1.5Volt)
2. Lalu ubah parameter seperti berikut:
  - a. Baseband voltage = 2 Volt; dan Carrier = 4 Volt
  - b. Baseband voltage = 1 Volt; dan Carrier = 1 Volt
  - c. Baseband voltage = 3 Volt; dan Carrier = 2 Volt
3. Amati perubahan sinyal dari tiap tiap parameter yang diberikan
4. Apa yang berubah dari sinyal AM dengan adanya perubahan parameter tersebut?
5. Bandingkan Hasil dari tiap parameter pada masing masing rangkaian

### F. Tugas

1. Berapa nilai indeks modulasi dari tiap parameter?  
Coba dengan tegangan (baseband = 1 Volt; dan Carrier = 1.5Volt) yang sama ubah ubah nilai frekuensi pada baseband dan carrier, apa yang berubah dari sinyal AM?

## PERCOBAAN 4. FREQUENCY MODULATION

### A. Tujuan Percobaan

1. Mahasiswa dapat memahami bagaimana cara kerja Modulasi Frekuensi.
2. Mahasiswa dapat memahami cara menggunakan software SistemVue.
3. Mahasiswa dapat menggunakan software SistemVue untuk mensimulasikan Modulasi Frekuensi.

### B. Alat dan Bahan

1. PC
2. Software SystemVue

### C. Teori Dasar

#### 1. Pengertian FM

Modulasi frekuensi didefinisikan sebagai deviasi frekuensi sesaat sinyal pembawa (dari frekuensi tak termodulasinya) sesuai dengan amplitudo sesaat sinyal pemodulasi. Sinyal pembawa dapat berupa gelombang sinus, sedangkan sinyal pemodulasi (informasi) dapat berupa gelombang apa saja (sinusoidal, kotak, segitiga, atau sinyal lain misalnya sinyal audio). Gambar 1 mengilustrasikan modulasi frekuensi sinyal pembawa sinusoidal dengan menggunakan sinyal pemodulasi yang juga berbentuk sinyal sinusoidal. Secara matematis, sinyal termodulasi FM dapat dinyatakan dengan:

$$e_{FM} = V_c \sin(\omega_c t + m_f \sin \omega_m t)$$

dengan:

$e_{FM}$  : sinyal termodulasi FM

$e_m$  : sinyal pemodulasi

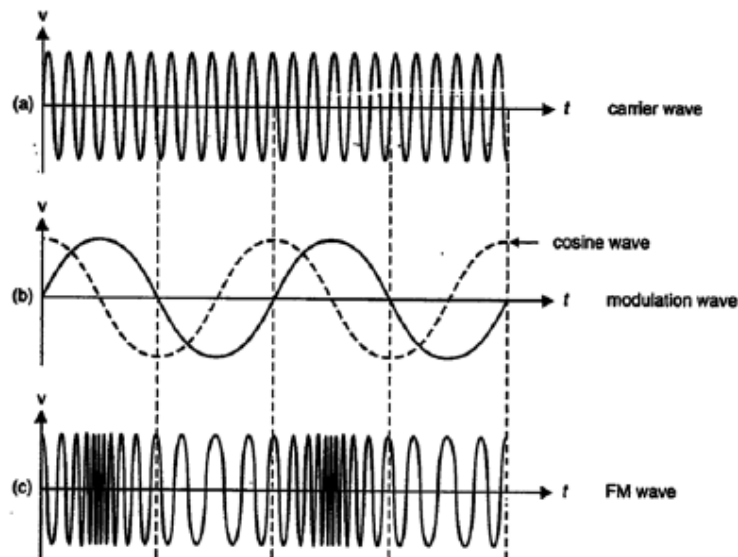
$e_c$  : sinyal pembawa

$V_c$  : amplitudo maksimum sinyal pembawa

$m_f$  : indeks modulasi FM

$\omega_c$  : frekuensi sudut sinyal pembawa (radian/detik)

$\omega_m$  : frekuensi sudut sinyal pemodulasi (radian/detik)



Gambar 1. (a) Sinyal pembawa (b) Sinyal pemodulasi  
(c) Sinyal termodulasi FM

## 2. Indeks Modulasi FM

Seperti telah dibahas, pada modulasi frekuensi maka frekuensi sinyal pembawa diubah-ubah sehingga besarnya sebanding dengan dengan besarnya amplitudo sinyal pemodulasi. Semakin besar amplitudo sinyal pemodulasi, maka semakin besar pula frekuensi sinyal termodulasi FM. Besar selisih antara frekuensi sinyal termodulasi FM pada suatu saat dengan frekuensi sinyal pembawa disebut deviasi frekuensi. Deviasi frekuensi maksimum didefinisikan sebagai selisih antara frekuensi sinyal termodulasi tertinggi dengan terendahnya.

Indeks modulasi FM ( $m_f$ ) merupakan perbandingan antara deviasi frekuensi maksimum dengan frekuensi sinyal pemodulasi:

$$m_f = \delta / f_m$$

dengan:

$\delta$  : deviasi frekuensi maksimum

$f_m$  : frekuensi maksimum sinyal pemodulasi

$m_f$  : indeks modulasi FM

Besarnya indeks modulasi FM dapat dipilih sebesar mungkin sejauh tersedia *bandwidth* (lebar bidang) untuk keperluan transmisinya. Biasanya besarnya indeks modulasi ini akan dimaksimalkan dengan cara mengatur besarnya deviasi frekuensi maksimal yang diijinkan.

Sinyal sinusoidal umum termodulasi FM didefinisikan dengan:

$$e_{fm} = \sin \left( \omega_c t - \frac{\Delta f}{f_m} \cos \omega_m t \right)$$

Indeks modulasi FM didefinisikan sebagai rasio deviasi carrier untuk modulasi frekuensi:

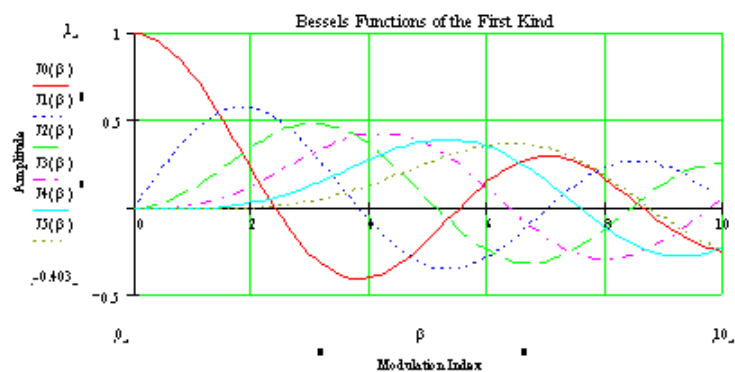
$$m_{fm} = \frac{\Delta f}{f_m}$$

Hasilnya, persamaan FM umumnya ditulis sebagai:

$$e_{fm} = \sin \left( \omega_c t - m_{fm} \cos \omega_m t \right)$$

FM menghasilkan jumlah frekuensi sisi tak terbatas yang masing-masing merupakan kelipatan bilangan bulat dari frekuensi modulasi sinyal. Fungsi Bessel digunakan untuk menghitung amplitudo mereka.

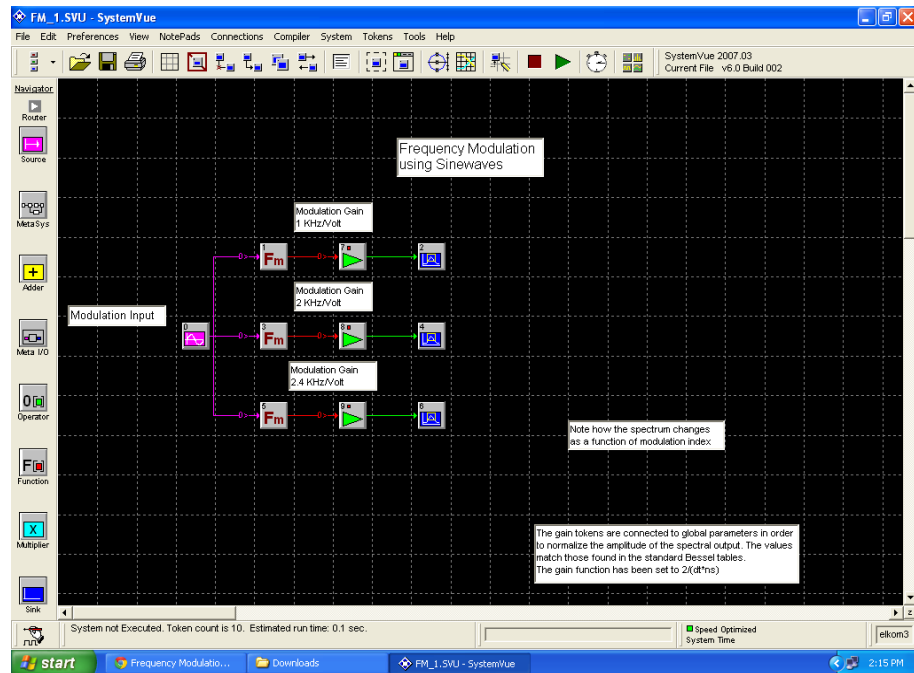
Sebuah plot amplitudo dari carrier dan lima pertama frekuensi sisi sebagai fungsi dari indeks modulasi menyerupai:



Gambar 2. Indeks Modulasi

#### D. Langkah Percobaan

1. Rangkaian seperti pada gambar di bawah ini (sinyal informasi sebesar 4 volt dan 500 Hz sedangkan sinyal carrier sebesar 2 volt dengan frekuensi sebesar 20 KHz)



2. Bandingkan apa perbedaan dari ketiga parameter tersebut
3. Analisa tentang hasil dari ketiga sinyal tersebut

## E. Tugas

1. Coba ubah ubah amplitude dari sinyal informasi (Baseband signal), apa yang terjadi?  
Buatlah satu rangkaian FM (Baseband 2Volt dan 1KHz; Carrier 1 Volt 10KHz) dengan menggunakan software SystemVue!

## PERCOBAAN 5. SINYAL

### A. Tujuan Percobaan

1. Mahasiswa dapat mengaplikasikan sistem menggunakan *Software* MATLAB.
2. Mahasiswa dapat mengaplikasikan fungsi Sinyal Waktu Kontinyu menggunakan *Software* MATLAB.
3. Mahasiswa dapat mengaplikasikan fungsi Sinyal Waktu Diskrit menggunakan *Software* MATLAB.

### B. Alat dan Bahan

1. Program MATLAB
2. Modul Praktikum Telekomunikasi

### C. Teori Dasar

#### 1. Definisi Sinyal

Sinyal didefinisikan sebagai kuantitas alami yang bervariasi sesuai dengan satu atau lebih variabel yang independen seperti waktu dan ruang. Waktu biasanya merupakan variabel yang independen, tetapi variabel seperti frekuensi dapat juga dipertimbangkan. Contoh dari sinyal adalah suara, gambar, arus listrik, tegangan, pesan yang ditransmisikan, dan lain-lain. Dari fungsi matematika, sinyal digambarkan oleh fungsi dari satu atau variabel bebas. Menurut jumlah variabel independen, sinyal dicirikan sebagai satu dimensi (1-D), dua dimensi (2-D), atau sinyal multidimensi.

#### 2. Kategori Tipe Variabel

Terdapat tiga kategori utama dimana sebuah sinyal dapat diklasifikasikan sesuai dengan jenis dari variabel independen dan dependen.

##### a. Sinyal Waktu Kontinyu

Sinyal dikatakan sinyal waktu kontinyu (analog) jika variabel independen (waktu) didefinisikan dalam interval terus menerus. Untuk sinyal 1-D, domain dari sinyal adalah interval yang kontinyu dari sumbu nyata. Selain itu, nilai dependen yang biasanya menunjukkan amplitudo sinyal juga merupakan variabel kontinyu. Sinyal analog dinyatakan oleh fungsi  $x(t)$ , dimana  $t$  adalah nilai real. Bagaimanapun, sinyal waktu kontinyu atau fungsi pendekatan menggunakan fungsi waktu diskrit yang sesuai dengan waktu yang sangat singkat.

##### b. Sinyal Waktu Diskrit



Sebuah sinyal dikatakan sinyal waktu diskrit jika  $n$  variabelnya bebas (waktu) yang dirumuskan pada interval diskrit (misalnya, himpunan bilangan integer), ketika nilai bergantung yang dirumuskan pada nilai kontinyu.

c. Sinyal Digital

Sinyal digital adalah sinyal yang kedua variabelnya bebas dan tidak bebas (bergantung) yang mengambil nilai diskrit.

3. Sinyal Waktu Kontinyu Dasar

a. Sinyal Sinusoida

Kategori dasar yang ditunjukkan adalah sinyal sinusoida. Jenis sinyal dalam bentuk  $x(t) = A \cos(\Omega t + \theta)$ , dimana  $\Omega$  adalah frekuensi sudut, diberikan dalam rad/s,  $A$  adalah amplitudo sinyal amplitudo, dan  $\theta$  adalah fasa (dalam radian). Sinyal sinusoida adalah sinyal periodic dengan periode dasar  $T$  yang diberikan oleh  $T = 2\pi/\Omega$  s. Jadi frekuensi didefinisikan sebagai  $f = 1/T$  atau  $f = \Omega/2\pi$ .

b. Sinyal Eksponensial

Sinyal eksponensial adalah sinyal dalam bentuk  $x(t) = Ae^{bt}$ . Jika  $b > 0$ , fungsi  $x(t)$  meningkat ketika  $b < 0$ ,  $x(t)$  adalah fungsi penurunan. Pada  $t = 0$  sinyal mengambil nilai  $x(0) = A$  sebagai  $e^{bt} = 1$

c. Fungsi  $u(t)$

Fungsi  $u(t)$  ditentukan oleh:

$$u(t) = \begin{cases} 1, & t > 0 \\ \frac{1}{2}, & t = 0 \\ 0, & t < 0 \end{cases}$$

Namun,  $u(t)$  adalah sinyal waktu kontinyu; sehingga nilai  $t = 0$  dapat dihilangkan dan  $u(t)$  dapat didefinisikan sebagai

$$u(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

Instruksi MATLAB menghasilkan fungsi unit step  $u(t)$  dengan instruksi `heaviside(t)`.

Berdasarkan program MATLAB, fungsi unit step ditentukan sebagai

$$u(t) = \begin{cases} 1, & t > 0 \\ 0, & t < 0 \end{cases}$$

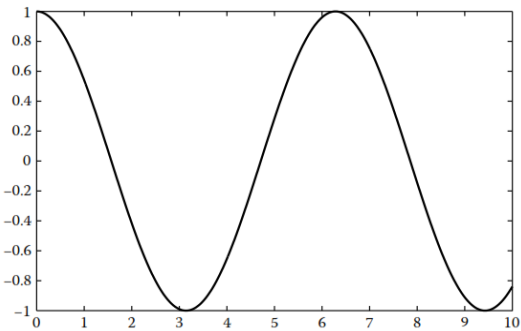
d. Fungsi Pulsa Kotak

Fungsi pulsa kotak  $pT(t)$  adalah pulsa kotak dengan unit amplitudo dan durasi waktu  $T$ . Hal ini didefinisikan dalam fungsi satuan  $u(t)$  sebagai berikut

$$pT(t) = u\left(t + \frac{T}{2}\right) - u\left(t - \frac{T}{2}\right) = \begin{cases} 1, & -T/2 \leq t \leq T/2 \\ 0, & \text{elsewhere} \end{cases}$$

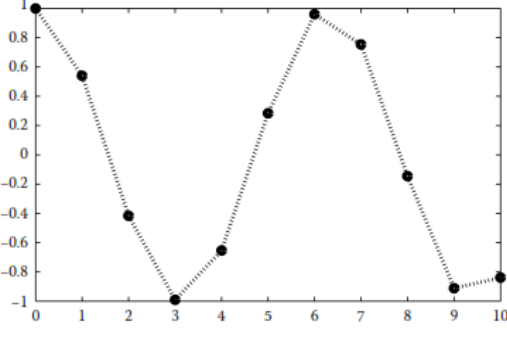
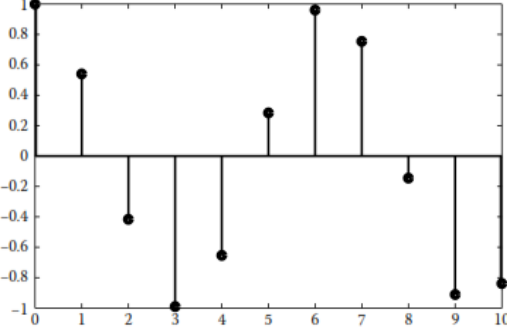
#### D. Langkah Percobaan

- Langkah untuk menggambar fungsi sinyal analog  $y(t) = \cos(t)$ ,  $0 \leq t \leq 10$

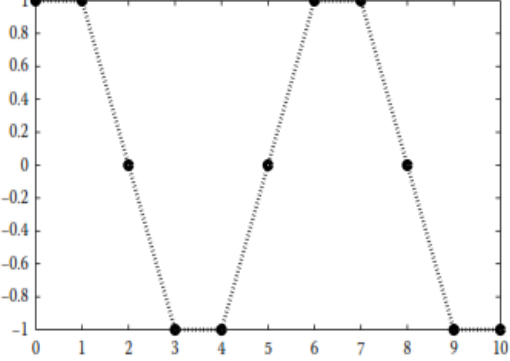
Instruksi	Hasil	Keterangan
$t=0:0.01:10$		Waktu (variabel bebas) didefinisikan dengan menggunakan langkah yang singkat (waktu = 0.01) dalam domain kontinyu $0 \leq t \leq 10$
$y=\cos(t);$		Variabel yang ber gantung $y(t)$ di-definisikan pada nilai kontinyu $-1 \leq y(t) \leq 1$ .
plot (t,y)		Sinyal analog di- gambarkan meng- gunakan instruksi plot.

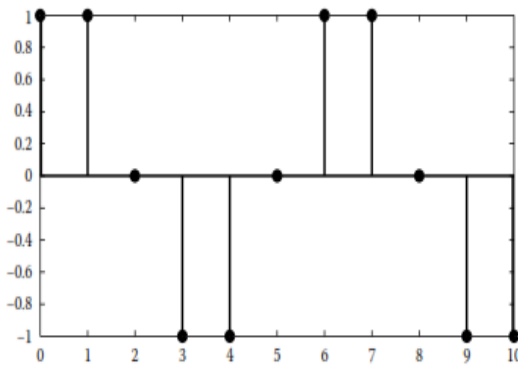
- Langkah untuk menggambar fungsi Sinyal Waktu Diskrit  $y(n) = \cos[n]$ . Perhatikan bahwa ketika mengacu pada waktu diskrit variabel  $n$  biasanya digunakan untuk menyatakan waktu.

Instruksi	Hasil	Keterangan
$n=0 : 10;$		
$y= \cos (n);$		

Plot $(n,y,'o')$		Grafik sinyal waktu diskrit $y[n]$ menggunakan instruksi plot dengan sintak yang tepat. Poin yang ditentukan oleh lingkaran adalah nilai-nilai $y[n]$ .
Stem $(n,y)$		Menggunakan instruksi stem lebih tepat ketika berhubungan dengan sinyal waktu diskrit.

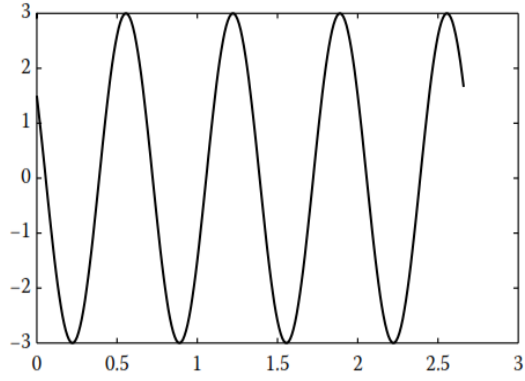
3. Langkah untuk menggambar fungsi Sinyal Digital  $y[n] = \cos[n]$  diplot sekali lagi, tetapi kita menggunakan instruksi *round* untuk membatasi nilai-nilai  $y[n]$ . Artinya,  $y[n]$  dapat -1, 0, 1.

Instruksi	Hasil	Keterangan
$n=0:10;$		Waktu diskrit $n$ (variabel bebas) definisikan dengan langkah 1.
$y=\cos(n);$		Didefinisikan sebagai variabel yang bergantung.
$y=\text{round}(y);$		Variabel $y[n]$ dibulatkan ke integer yang terdekat; yaitu diambil dari nilai-nilai diskrit
$\text{plot}(n,y,'o')$		Grafik sinyal waktu diskrit menggunakan instruksi <i>plot</i> dengan sintak yang tepat. Poin yang ditentukan oleh lingkaran adalah nilai-nilai $y[n]$ .

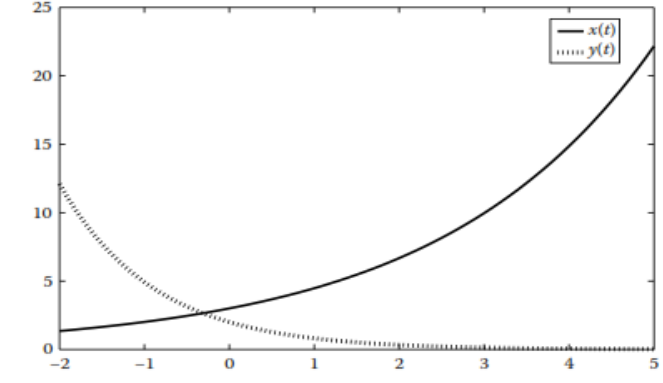
stem(n,y)		Menggunakan instruksi <i>stem</i> yang lebih tepat untuk sinyal digital.
-----------	---	--

4. Langkah untuk menggambarkan Sinyal Sinusoida  $x(t) = 3 \cos(3\pi t + \pi/3)$  di empat periode

Pertama, periode  $T$  di hitung dengan  $T = 2\pi/\Omega = 2\pi/3\pi = 2/3$ . Oleh karena itu, penerapan MATLAB sebagai berikut:

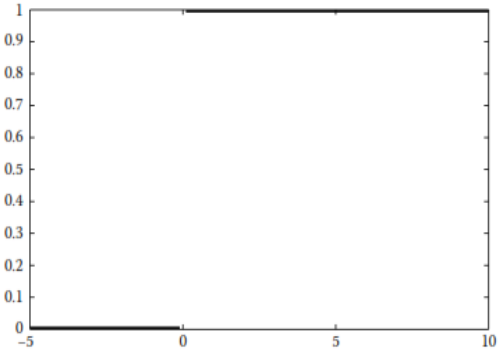
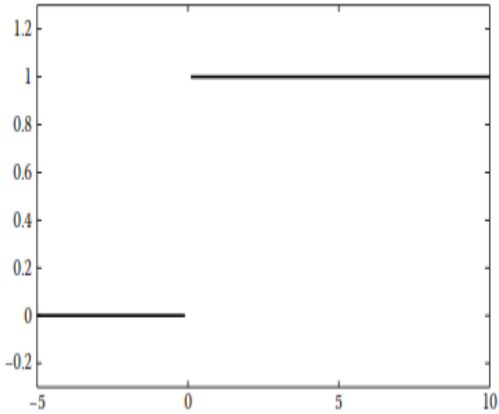
Instruksi	Hasil	Keterangan
A=3;		Amplitudo sinyal adalah 3
omega = 3*pi;		Sudut frekuensi adalah $3\pi$
thita = pi/3;		Phasa = $\pi/3$
T=2*pi/omega;		Perioda = $2/3$
t=0:0.01:4*T;		Waktu yang dirumuskan dari 0 – 4T.
x=A*cos(omega*t+thita);		Mendefinisikan sinyal
plot(t,x)		

5. Gambarkan fungsi sinyal eksponensial  $x(t) = 3e^{0.4t}$  dan  $y(t) = 2e^{-0.9t}$  pada interval waktu  $-2 \leq t \leq 5$ .

Instruksi	Hasil/Keterangan
<pre>t = -2:.1:5; x = 3*exp(0.4*t); y = 2*exp(-0.9*t); plot(t,x,t,y, ' : '); legend('x (t)', 'y (t)')</pre>	

6. Langkah untuk menggambarkan fungsi  $u(t)$ .

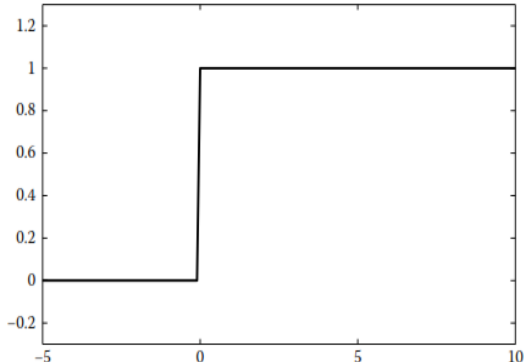
- Metode pertama → menggunakan instruksi Heaviside.

Instruksi	Hasil	Keterangan
<code>t=-5:0.1:10;</code>	<code>t=-5,...,-0.1, 0, 0.1,...,10</code>	Definisi dari interval waktu $-5 \leq t \leq 10$
<code>u=heaviside(t);</code>	<code>u=0,...,0, NaN, 1,...,1</code>	Definisi dari $u(t)$
<code>plot(t,u)</code>		Grafik dari $u(t)$
<code>ylim([-0.3 1.3])</code>		Perubahan Penampilan grafik yang lebih baik dan terbaca.

- Metode kedua → menggunakan teknik pendefinisian dan fungsi plotting bagian demi bagian.

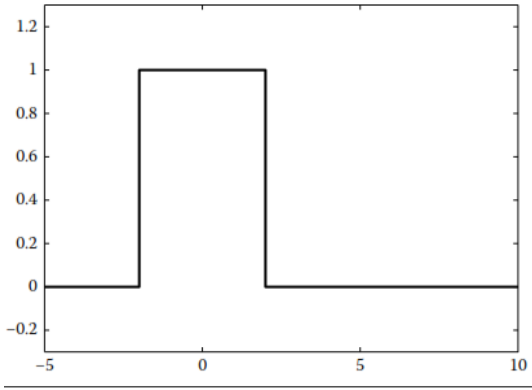
Instruksi	Hasil	Keterangan
<code>t1=-5:1:0;</code>	<code>t1 = -5, -4.9, ..., 0</code>	Definisi pada waktu interval pertama $-5 \leq t \leq 0$ .
<code>t2=0:1:10;</code>	<code>t2 = 0, 0.1, ..., 10</code>	Definisi pada waktu interval $0 \leq t \leq 10$ .
<code>u1=zeros(size(t1));</code>	<code>u1 = 0, 0, ..., 0</code>	Implementasi bagian dari $u(t)$ yang sesuai dengan waktu $t1$ .
<code>u2=ones(size(t2));</code>	<code>u2 = 1, 1, ..., 1</code>	Implementasi bagian dari $u(t)$ yang sesuai dengan waktu $t2$ .
<code>t=[t1 t2];</code>	<code>t = -5, ..., -0.1, 0, ..., 10</code>	Gabungan dari dua vektor waktu.
<code>u=[u1 u2];</code>	<code>u = 0, ..., 0, 1, ..., 1</code>	Gabungan dari dua vektor waktu.
<code>plot(t,u);</code>		Grafik dari fungsi $u(t)$ pada waktu interval $0 \leq t \leq 10$ .
<code>ylim([-0.3 1.3])</code>		Perubahan Penampilan grafik yang lebih baik dan terbaca.

- Metode ketiga → implementasi dengan nilai 0 dan 1.

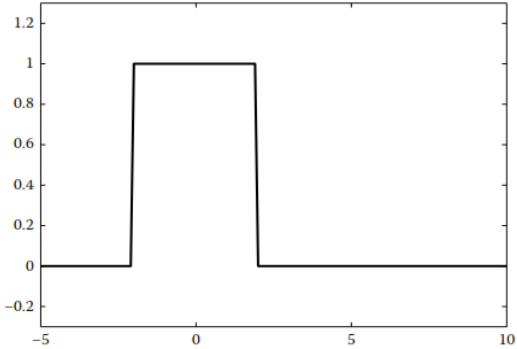
Instruksi	Hasil	Keterangan
<code>t = -5:1:10;</code>	<code>t = -5,...,-0.1, 0,...,10</code>	Definisi waktu
<code>u=[zeros(1,50) ones(1,101)];</code>	<code>U = 0,...,0, 1,...,1</code>	Vektor <code>t</code> terdiri dari 151 elemen. Dengan demikian yang pertama elemen 50 pada <code>t</code> cocok dengan nol "0". Sedangkan berikutnya elemen 101 (termasuk <code>t=0</code> ) pada <code>t</code> cocok dengan "1". Kedua vektor yang berhubungan.
<code>plot(t,u) ylim([-0.3 1.3])</code>		Grafik dari fungsi $u(t)$ pada interval waktu $0 \leq t \leq 10$ .

7. Langkah untuk menggambarkan sinyal pulsa kotak,  $T = 4$ , maka sinyal  $p_4(t)$  memiliki durasi  $T = 4$  dengan pusat pada  $t = 0$ .

Instruksi	Hasil	Keterangan
<code>t1=-5:1:-2;</code>		Definisi interval waktu pertama $-5 \leq t \leq -2$ .
<code>t2=-2:1:2;</code>		Definisi interval waktu kedua $-2 \leq t \leq 2$ .
<code>t3=2:1:10;</code>		Definisi interval waktu ketiga $2 \leq t \leq 10$ .

<code>p1=zeros(size(t1));</code>		Fungsi $pT(t)$ adalah nol adalah untuk $-5 \leq t \leq -2$ .
<code>p2=ones(size(t2));</code>		Fungsi $pT(t)$ adalah satu untuk $-2 \leq t \leq 2$ .
<code>p3=zeros(size(t3));</code>		Fungsi $pT(t)$ adalah nol untuk $2 \leq t \leq 10$ .
<code>t=[t1 t2 t3];</code>		Gabungan waktu
<code>p=[p1 p2 p3];</code>		Gabungan fungsi
<code>plot(t,p);</code> <code>ylim([-0.3 1.3]);</code>		Grafik sinyal dan dimodifikasi agar tampilan lebih baik dan terbaca.

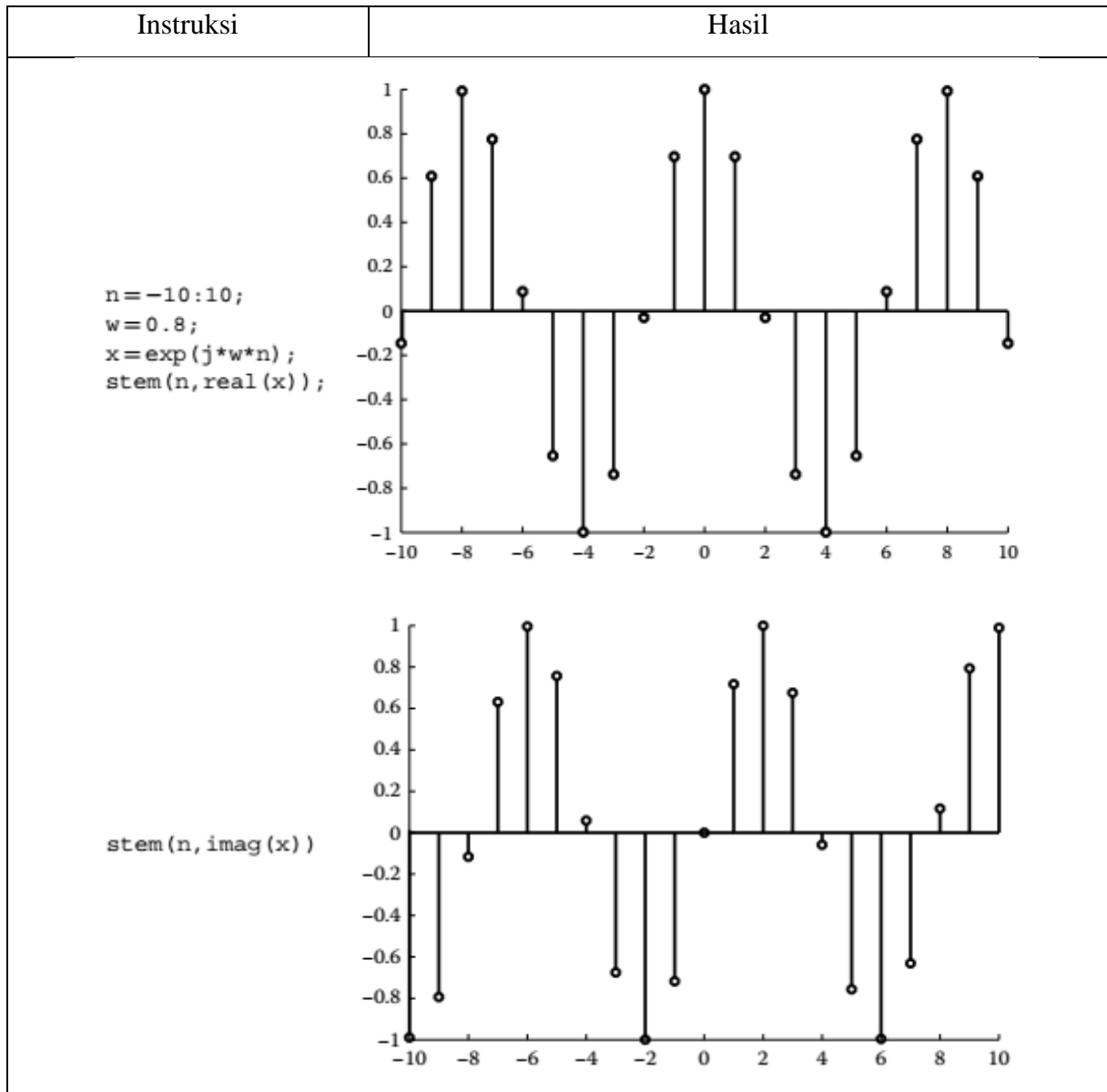
- Gambarkan pulsa kotak dengan menggunakan instruksi “rectpuls”. Dengan menggunakan sintak “rectpuls” akan didapatkan pulsa kotak dengan durasi  $T = 1$ .

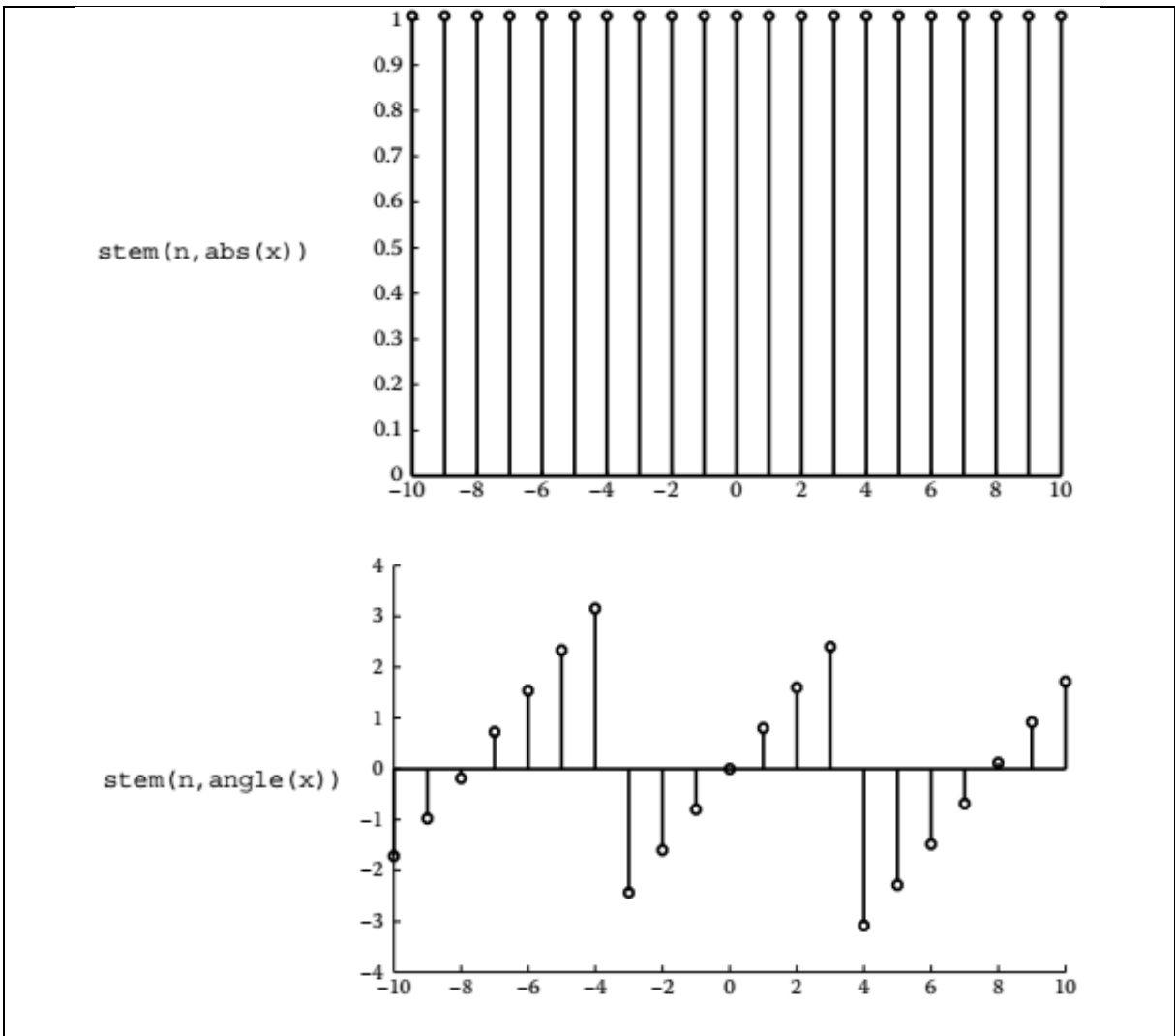
Instruksi	Hasil	Keterangan
<code>t=-5:1:10;</code>	<code>t = -5,...,-0.1, 0,...,10</code>	Definisi waktu
<code>s=rectpuls(t,4);</code>		Pulsa kotak dengan lebar $T = 4$
<code>plot(t,s)</code> <code>ylim([-0.3 1.3])</code>		Grafik sinyal dan modifikasi dari poros y dan tampilan yang lebih baik.



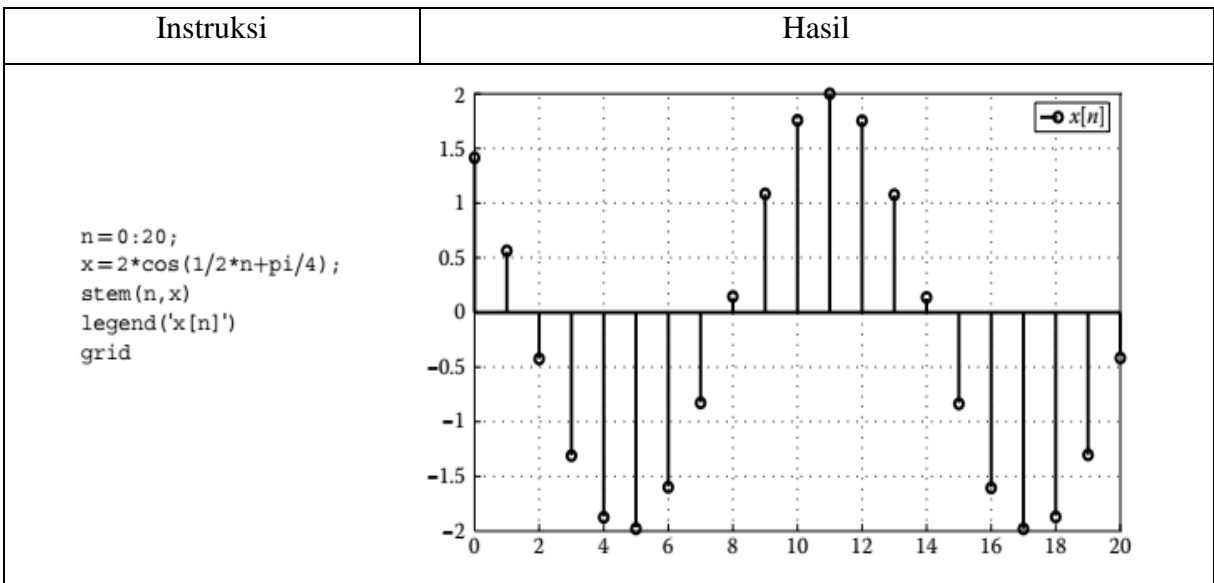
## 8. Sinyal Waktu Diskrit

- Complex Exponential Sequence

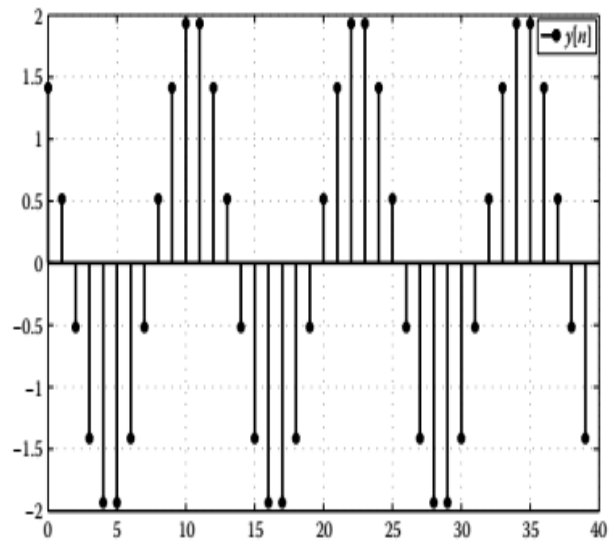




- Sinusoidal Sequence



```
y=2*cos(pi/6*n+pi/4);  
stem(n,y)  
legend('y[n]')  
grid
```



## PERCOBAAN 6. SISTEM

### A. Tujuan Percobaan

1. Mahasiswa dapat mengetahui dan memahami konsep sistem dalam *Matrix Laboratory* (MATLAB).
2. Mahasiswa dapat memrogram *software* MATLAB.
3. Mahasiswa dapat membuat sebuah sistem sinyal dalam *software* MATLAB.

### B. Alat dan Bahan

1. Program MATLAB
2. Modul Praktikum Telekomunikasi

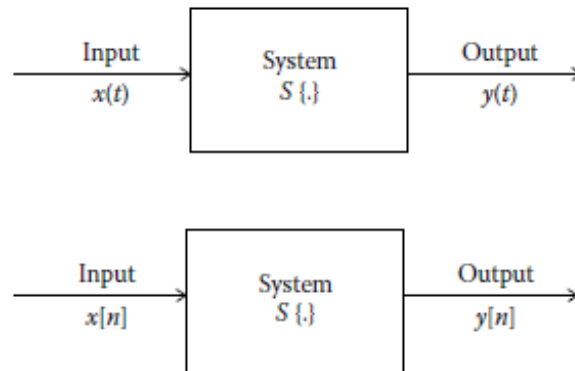
### C. Teori Dasar

Sebuah sistem merupakan konsep dengan beragam fungsi yang digunakan dalam berbagai kehidupan sehari-hari. Dalam pandangan teknik elektro, sistem merupakan kesatuan manipulasi satu atau lebih sinyal untuk menunjukkan sebuah operasi dan mengembalikan hasil operasi sebagai satu atau lebih sinyal. Pengertian umum lainnya, bahwa sistem waktu kontinu atau waktu diskrit merupakan kesatuan yang merubah sinyal masukan  $x(t)$  atau  $x[n]$  menjadi keluaran  $y(t)$  atau  $y[n]$  yang menunjukkan keluaran operasi tertentu. Transformasi  $x(t)$  atau  $x[n]$  ditunjukkan dengan  $y(t)=S[x(t)]$  atau  $y[n]=S[x[n]]$ , dimana S merupakan sistem. Dalam sebuah sinyal sistem, sebuah sistem dianggap sebagai *black box*, yang berarti sistem memeriksa hubungan dengan masukan, keluaran, dan segala sesuatunya tanpa perlu mengetahui masukannya bekerja.

Sebuah contoh sederhana adalah gitar elektrik. Dengan gitar yang dipetik merupakan masukannya, maka gitar elektrik, penguat, dan boks *speaker* merupakan sebuah sistem. Maka keluaran sinyal berbeda dengan sinyal masukannya (suaranya lebih keras, misal memiliki amplitudo yang lebih besar dan biasanya durasi yang lebih lama). Dengan pendekatan *black box*, kita akan memperhatikan sinyal masukan dan keluarannya, dibandingkan dengan memperhatikan implementasi dalam sistem tersebut (contohnya rangkaian *amplifier*).

1. Klasifikasi Sistem
  - a. Sistem Berdasarkan Jumlah Masukan dan Keluaran
    - 1) Sistem *Single-Input Single-Output* (SISO)

Gambar 2.1 menunjukkan sistem SISO, dimana pada sistem ini menunjukkan hubungan I/O dalam bentuk sebuah masukan  $x(t)$  dan sebuah keluaran  $y(t)$  yang merupakan respon sistem terhadap sinyal masukan.



Gambar 2.1 Blok Diagram Menggambarkan Sistem Waktu Kontinyu (atas) dan Waktu Diskrit (bawah)

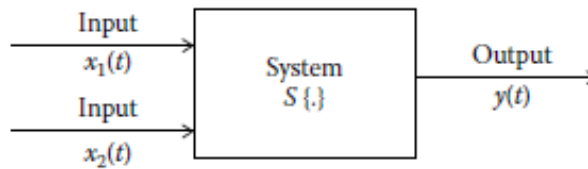
Jika keluaran  $y(t) = x(t-2)$ , dan masukan  $x(t) = t \cos(2\pi t)$ ,  $0 \leq t \leq 3$ .

Instruksi	Hasil	Keterangan
<pre>t=0:.01:3; x=t.*cos(2*pi*t); plot(t,x); title('Input signal x(t)')</pre>		Grafik masukan $x(t)$ .
<pre>plot (t+2,x); title('Output signal y(t)')</pre>		Grafik keluaran $y(t)$ .

## 2) Sistem *Multiple-Input Single-Output* (MISO)

Sistem ini terdiri dari beberapa masukan sistem dan sebuah keluaran.

Gambar 3.2 merupakan blok diagram dari sistem MISO.



Gambar 2.2 Blok Diagram Sistem MISO

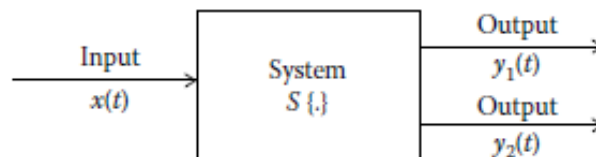
Jika keluaran  $y(t) = x_1(t) + x_2(t) \cdot x_3(t)$ , dan masukan  $x_1(t) = u(t) - u(t-3)$ ,  $x_2(t) = t \sin(t)$ ,  $0 \leq t \leq 4$ .  $x_3(t) = t \cos(t)$ ,  $0 \leq t \leq 4$ .

Instruksi	Hasil	Keterangan
<pre>t=0:0.01:4; x1=heaviside(t)- heaviside(t-3); x2=t.*sin(t); x3=t.*cos(t); plot(t,x1,'-', t,x2,t,x3,':'); legend ('x1(t)','x2(t)','x3(t)')</pre>		Grafik masukan sinyal $x_1(t)$ , $x_2(t)$ dan $x_3(t)$
<pre>y=x1+x2.*x3; plot(t,y); legend('Output y(t)')</pre>		Grafik respon sistem $y(t)$ .

### 3) Sistem *Single-Input Multiple-Output* (SIMO)

Sistem yang terdiri dari satu masukan dan beberapa keluaran sistem.

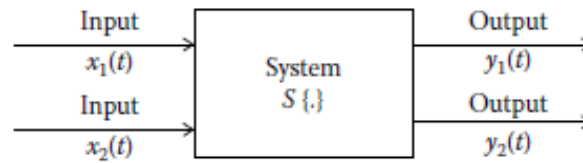
Gambar 2.3 merupakan blok diagram sistem SIMO.



Gambar 2.3 Blok Diagram Sistem SIMO

### 4) Sistem *Multiple-Input Multiple-Output* (MIMO)

Beberapa sinyal menjadi masukan, begitupun keluaran dari sistem ini adalah beberapa keluaran. Gambar 2.4 menunjukkan blok diagram dari sistem MIMO ini.



Gambar 2.4 Blok Diagram Sistem MIMO

Berikut percobaan MIMO.

Jika keluaran  $y_1(t) = x_1(t) + x_2(t)$ , dan keluaran  $y_2(t) = x_1(t) - x_2(t)$

Masukan  $x_1(t) = u(t)$ , dan  $x_2(t) = 0,5 \cdot u(t-1)$

Instruksi	Hasil	Keterangan
<pre>t=0:0.01:4; x1=heaviside(t); x2=0.5*heaviside(t-1); plot(t,x1,t,x2,':'); legend ('x1(t)','x2(t)')</pre>		Grafik masukan sinyal.
<pre>y1=x1+x2; y2=x1-x2; plot(t,y1,t,y2,':'); ylim([-0.1 2]); legend ('y1(t)','y2(t)')</pre>		Grafik respon sistem.

b. Sinyal Waktu Kontinu dan Waktu Diskrit

Klasifikasi sistem kedua adalah berhubungan dengan keluaran dari sistem yang disebut dengan sistem normal. Secara spesifik, sistem waktu kontinu adalah masukan sinyal waktu kontinu, dan keluaran sinyal waktu kontinu. Dengan kata lain, sistem waktu diskrit merupakan sebuah sistem yang masukannya merupakan sinyal waktu diskrit, dan keluarannya merupakan sinyal waktu diskrit. Selanjutnya, disebut sistem *hybrid* jika masukan sinyal waktu kontinu dan keluarannya menjadi sinyal waktu diskrit, begitupun sebaliknya. Pengubah sinyal

analog menjadi sinyal digital atau (A/D) *converter* merupakan contoh sistem *hybrid*.

c. Sinyal Deterministik dan Stokastik

Klasifikasi ketiga dari sistem adalah antara sistem deterministik dan sistem stokastik. Dimana, sistem deterministik jika tidak adanya pengacakan yang dilibatkan pada sinyal masukan hingga menjadi sinyal keluaran. Atau dengan kata lain, keluaran dari sistem tersebut dapat diprediksi. Berbeda dengan sistem stokastik dimana keluaran dari sistem berbeda dengan masukan yang diberikan, dengan kata lain tidak dapat diprediksi keluaran dari sistem tersebut.

## 2. Sifat Sistem

Ada beberapa sifat sistem yang terdapat dalam sebuah sistem, baik waktu kontinyu dan waktu diskrit, adalah sebagai berikut.

a. Sistem Kausal dan Non-Kausal

Sebuah sistem dikatakan kausal jika keluaran sistem  $y(t_0)$  pada waktu  $t=t_0$  tidak tergantung pada masukan sistem  $x(t)$  untuk  $t > t_0$ . Dengan kata lain, untuk setiap sinyal masukan  $x(t)$ , bersamaan dengan  $y(t)$  tergantung hanya pada nilai sekarang dan nilai sebelum dari  $x(t)$ . Begitupun dengan sistem jika dimasukan sinyal waktu diskrit, sistem disebut kausal, jika keluaran  $y[n_0]$  pada waktu  $n=n_0$  tergantung hanya pada nilai masukan  $x[n]$  untuk  $n \leq n_0$ . Semua sistem normal adalah kausal. Tetapi pada teknik, banyak sekali sistem non-kausal, contohnya adalah pemrosesan data *offline*.

b. Sistem Statis (*Memoryless*) dan Dinamis (*with Memory*)

Sebuah sistem statis atau *memoryless*, jika untuk setiap masukan sinyal  $x(t)$  atau  $x[n]$  diikuti dengan keluaran  $y(t)$  atau  $y[n]$  tergantung hanya pada nilai masukan pada waktu yang sama. Tetapi, sebuah sistem tidak statis disebut dengan sistem dinamis (memakai memori).

c. Sistem Linear dan Non-linear

Dengan  $y(t)$  merupakan respon sistem dari sistem S terhadap sinyal masukan  $x(t)$ , sehingga  $y(t)=S[x(t)]$ . Sistem S disebut linear jika setiap sinyal masukan  $x_1(t)$  dan  $x_2(t)$  serta setiap nilai skalar  $a_1$  dan  $a_2$  mengikuti persamaan berikut.

$$S[a_1x_1(t) + a_2x_2(t)] = a_1S[x_1(t)] + a_2S[x_2(t)] \quad \text{persamaan 2.1}$$

Dengan kata lain, respon sistem linear terhadap sinyal masukan merupakan sebuah kombinasi linear dari dua sinyal yang sebanding dengan



kombinasi linear dari respon sistem pada setiap sinyal masukan. Sifat linearitas merupakan kombinasi dari dua sifat, yaitu sifat penjumlahan dan homogenitas. Berikut merupakan sifat penjumlahan.

$$S[x_1(t) + x_2(t)] = S[x_1(t)] + S[x_2(t)]$$

Dan berikut merupakan sifat homogenitas dengan setiap nilai skalar  $a$  untuk setiap masukan  $x(t)$ .

$$S[ax(t)] = aS[x(t)]$$

Pada sinyal waktu diskrit sifat linear dimiliki dalam persamaan.

$$[a_1x_1[n] + a_2x_2[n]] = a_1S[x_1[n]] + a_2S[x_2[n]] \quad \text{persamaan 2.2}$$

**d. Sistem Invarian Waktu dan Varian Waktu**

Sistem invarian waktu, jika waktu geser pada sinyal masukan adalah sama dengan waktu geser pada sinyal keluaran. Dengan kata lain, jika Dengan kata lain, jika  $y(t)$  adalah respon sistem invarian waktu terhadap sinyal masukan  $x(t)$ , maka respon terhadap sinyal masukan  $x(t-t_0)$  adalah  $y(t-t_0)$ . Maka, persamaan secara matematis adalah.

$$y(t - t_0) = S[x(t - t_0)] \quad \text{persamaan 2.3}$$

Dan pada waktu sinyal diskrit adalah sebagai berikut.

$$y[n - n_0] = S[x(n - n_0)] \quad \text{persamaan 3.4}$$

Dan untuk sistem varian waktu adalah adanya perbedaan waktu geser pada hasil keluaran jika dibandingkan dengan sinyal masukan.

**e. Sistem Dapat Dibalikkan (*Invertible*) dan Tidak Dapat Dibalikkan (*Non-invertible*)**

Sistem dapat dibalikkan jika sinyal masukan  $x(t)$  yang diaplikasikan pada sistem dapat diperoleh kembali dari respon sistem  $y(t)$ . Dengan kata lain, sebuah sistem dapat dibalikkan jika hubungan i/o  $y(t)=S[x(t)]$  adalah satu persatu, yaitu, jika nilai masukan berbeda sama dengan nilai keluaran yang berbeda.

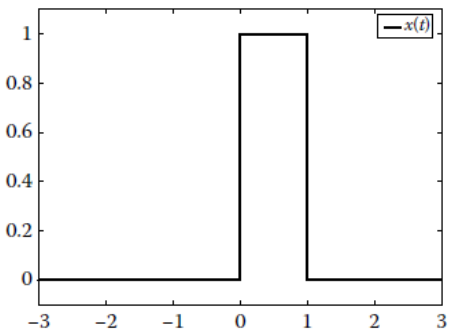
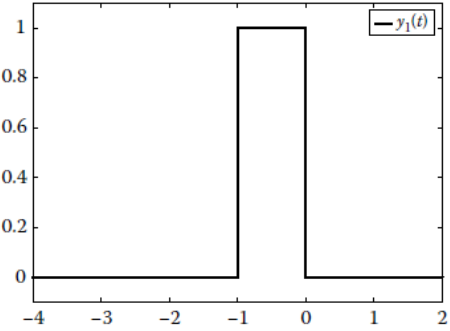
**f. Sistem Stabil dan Tidak Stabil**

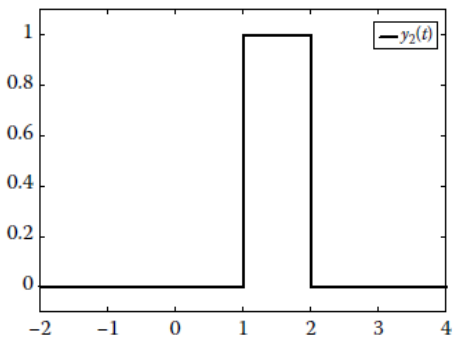
Sistem stabil atau stabil *bounded-input bounded-output* (BIBO), jika respon sistem untuk setiap sinyal masukan terbatas (*bounded-input*) menghasilkan keluaran terbatas juga (*bounded-output*). Secara matematis, dengan positif nomor  $M < \infty$ , seperti  $|x(t)| \leq M$ . Sistem stabil jika  $\forall t \in \mathbb{R}$  sebuah nomor positif  $N < \infty$ , sehingga  $|x(t)| \leq N$ .

## D. Langkah Percobaan

### 1. Membuat Sifat Sistem Kausal/Non-kausal

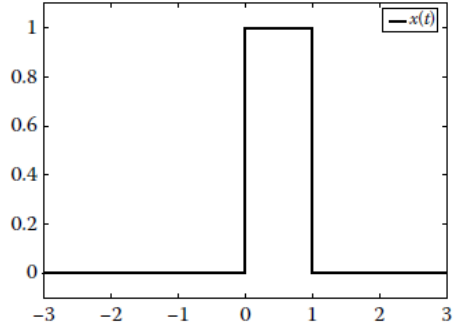
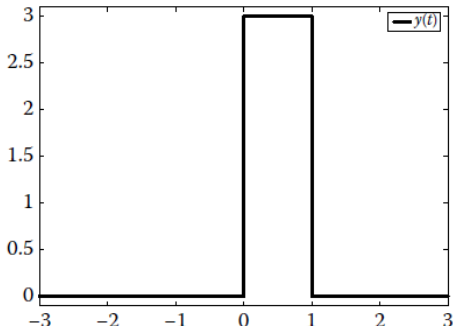
Jika sistem pertama ( $S_1$ ) memiliki hubungan i/o  $y(t)=x(t+1)$ , dan sistem kedua ( $S_2$ ) memiliki hubungan i/o  $y(t)=x(t-1)$ . Gunakan masukan  $x(t)=u(t)-u(t-1)$ . Tentukan sistem kausal atau non-kausal.

Instruksi	Hasil	Keterangan
<pre>t1 = -3 : .1 : 0; x1 = zeros(size(t1)); t2 = 0 : .1 : 1; x2 = ones(size(t2)); t3 = 1 : .1 : 3; x3 = zeros(size(t3)); t = [t1 t2 t3]; x = [x1 x2 x3]; plot (t,x); ylim ([-0.1 1.1]); legend ('x(t)')</pre>		<p>Grafik pada interval <math>-3 \leq t \leq 3</math> pada masukan sinyal <math>x(t)=u(t)-u(t-1)</math>-</p> $u(t-1) = \begin{cases} 1, & 0 \leq t \leq 1 \\ 0, & \text{elsewhere} \end{cases}$
<pre>plot (t-1,x) ylim ([-0.1 1.1]); legend ('y_1(t)')</pre>		<p>Keluaran <math>S_1</math> adalah <math>y(t)=x(t+1)</math>. Masukan <math>x(t)</math> bernilai nol untuk <math>t &lt; 0</math>, tetapi keluaran <math>y(t)</math> tidak bernilai nol untuk <math>t &lt; 0</math>. Dengan kata lain, <math>y(t)</math> tergantung pada nilai yang akan datang dari <math>x(t)</math>. Sistem tersebut tidak kausal.</p>

<pre>plot (t+1,x) ylim ([-0.1 1.1]); legend ('y_2(t)')</pre>		<p>Keluaran <math>S_2</math> adalah <math>y(t)=x(t-1)</math>. Keluaran <math>y(t)</math> bernilai nol untuk <math>t &lt; 1</math>. Dengan kata lain, <math>y(t)</math> tergantung hanya pada nilai masa lampau dari <math>x(t)</math>. Sistem tersebut kausal.</p>
--	--	--

## 2. Membuat Sifat Sistem Statis/Dinamis

- a. Gunakan masukan  $x(t)=u(t)-u(t-1)$ . Jika sistem pertama ( $S_1$ ) memiliki hubungan i/o  $y(t)=3x(t)$ , dan sistem kedua ( $S_2$ ) memiliki hubungan i/o  $y(t)=x(t)+(t-1)$ ..  
Tentukan sistem statis atau dinamis.

Instruksi	Hasil	Keterangan
<pre>t1 = -3 : .1 : 0; x1 = zeros(size(t1)); t2 = 0 : .1 : 1; x2 = ones(size(t2)); t3 = 1 : .1 : 3; x3 = zeros(size(t3)); t = [t1 t2 t3]; x = [x1 x2 x3]; plot (t,x); ylim ([-0.1 1.1]); legend ('x(t)')</pre>		<p>Grafik pada interval <math>-3 \leq t \leq 3</math> pada masukan sinyal <math>x(t)=u(t)-u(t-1)</math></p> $u(t-1) = \begin{cases} 1, & 0 \leq t \leq 1 \\ 0, & \text{elsewhere} \end{cases}$
<pre>plot (t,3*x); ylim ([-0.1 3.1]); legend ('y(t)')</pre>		<p>Keluaran sistem dengan hubungan i/o adalah <math>y(t)=3x(t)</math> tergantung hanya pada nilai masukan dengan waktu yang sama. Dengan kata lain, sistem tersebut</p>

		merupakan sistem statis ( <i>memoryless</i> ).
--	--	--

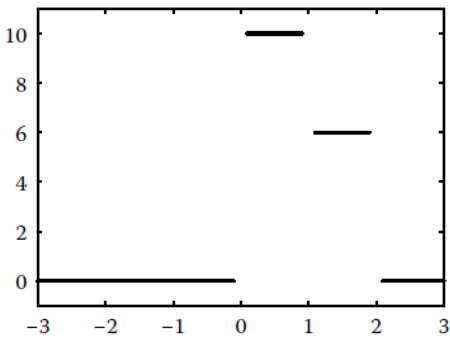
Untuk sistem kedua yang memiliki hubungan i/o  $y(t)=x(t)+x(t-1)$ , ingat kembali bahwa  $x(t)=u(t)-u(t-1)=1, 0 \leq t \leq 1$ ; dengan demikian  $x(t-1)=u(t-1)-u(t-2)=1, 1 \leq t \leq 2$  dan  $y(t)=u(t)-u(t-2)=1, 0 \leq t \leq 2$ . Nilai dari  $y(t)$  tergantung pada nilai masa lampau  $x(t)$ , jadi sistem tersebut adalah dinamis.

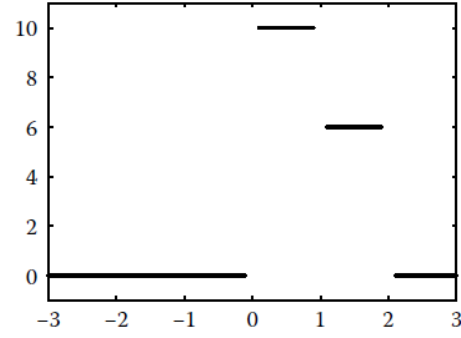
- b. Tentukan jika waktu diskrit dengan hubungan i/o  $y[n]=x^2[n]$  dan  $y[n]=x[n/2]$  adalah sistem statis atau dinamis. Gunakan masukan  $x[n]=[0 \ 1 \ 2 \ 3 \ 4], -1 \leq n \leq 3$ .

Instruksi	Hasil	Keterangan
<pre>n = -3 : 3; x = [0 1 2 3 4]; stem (n,x); axis([-1.1 3.1 -1.1 4.1]); legend ('x[n]')</pre>		Grafik sinyal waktu diskrit $x[n]=[0 \ 1 \ 2 \ 3 \ 4], -1 \leq n \leq 3$ .
<pre>y = x.^2; stem (n,y); axis([-1.1 3.1 -1.1 16.1]); legend ('y_1[n]')</pre>		Grafik dengan $y[n]=x^2[n]$ , dimana nilai $y[n]$ tergantung hanya pada nilai $x[n]$ untuk $n$ -yang sama. Dengan kata lain, sistem merupakan sistem statis.
<pre>a = 1/2; y = upsample(x,1/a); stem (-2:7,y); axis([-2.2 7.2 -1.1 4.1]); legend ('y_2[n]')</pre>		Hubungan i/o $y[n]=x[n/2]$ dimana sebagai contoh nilai $y[n]$ untuk $n=6$ tergantung pada nilai $x[n]$ untuk $n=3$ . Sistem tersebut merupakan sistem dengan memori (dinamis).

### 3. Membuat Sifat Sistem Linear/non-Linear

- a. Dengan  $x_1(t)=u(t)-u(t-1)$  dan  $x_2(t)=u(t)-u(t-2)$  menjadi masukan sistem yang memiliki hubungan i/o  $y(t)=2x(t)$  Tentukan sistem bersifat linear atau non-linear.

Instruksi	Hasil	Keterangan
<pre>t = -3 : .1 : 3; x1 = heaviside (t) - heaviside (t-1); x2 = heaviside (t) - heaviside (t-2);</pre>		Deskripsi sinyal masukan $x_1(t)$ dan $x_2(t)$
<pre>%Komputerisasi untuk bagian kiri dari persamaan 3.1  a1=2; a2=3; z=a1*x1+a2*x2;</pre>		Untuk pernyataan $a_1x_1(t)+a_2x_2(t)$
<pre>y = 2*z; plot (t,y); ylim ([-1 11]);</pre>		Sisi kiri dari persamaan 2.1 atau $S[a_1x_1(t)+a_2x_2(t)]$ dikomputerisasi dan hasilnya ditampilkan.
<pre>%Komputerisasi untuk bagian kanan dari persamaan 3.2  z1=2*x1; z2=2*x2; z=a1*x1+a2*x2;</pre>		Pernyataan dari $S[x_1(t)]$ $S[x_2(t)]$

<pre> y = a1*z1+a2*z2; plot (t,y); ylim ([-1 11]); </pre>		<p>Sisi kanan dari persamaan 2.1 atau <math>a_1S[x_1(t)] + a_2S[x_2(t)]</math> dikomputerisasi dan hasilnya ditampilkan.</p>
---	--	--

Dari dua grafik diperoleh hasil yang sama, sehingga hubungan  $y(t)=2x(t)$  adalah linear.

- b. Dengan  $x_1[n]=0.8[n]$ ,  $0 \leq n \leq 5$  dan  $x_2[n]=\cos[n]$ ,  $0 \leq n \leq 5$  menjadi masukan sistem yang memiliki hubungan i/o  $y[n]=2^{x[n]}$ . Tentukan sistem tersebut apakah linear atau non-linear.

Instruksi	Hasil	Keterangan
<pre> n = 0 : 5; x1 = 0.8.^n; x2 = cos(n); a1 = 2; a2 = 3; z = a1*x1+a2*x2; y1 = 2.^z </pre>	<pre> y1 = 32.0000 9.3237 1.0221       0.2595 0.4532 2.8409 </pre>	<p>Komputerisasi sisi kiri dari persamaan 2.2</p>
<pre> z1 = 2.^x1; z2 = 2.^x2; y2 = a1*z1+a2*z2 </pre>	<pre> y2 = 10.0000 7.8450 5.3649       4.3625 4.5637 6.1618 </pre>	<p>Komputerisasi sisi kanan dari persamaan 2.2. Hasil dari kedua sisi berbeda, sehingga dapat disimpulkan bahwa <math>y[n]=2^{x[n]}</math> adalah tidak linear.</p>

#### 4. Membuat Sifat Sistem Invarian Waktu/Varian Waktu

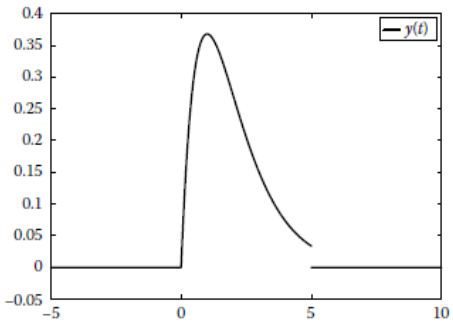
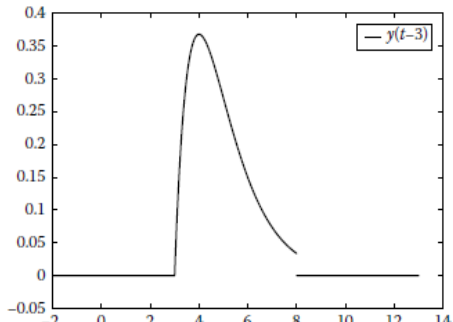
- a. Jika keluaran sinyal  $y(t)=te^{-t}x(t)$ . Tentukan jika sistem adalah invarian waktu dengan menggunakan masukan sinyal  $x(t)=u(t)-u(t-5)$ .

Sisi kiri persamaan 2.3

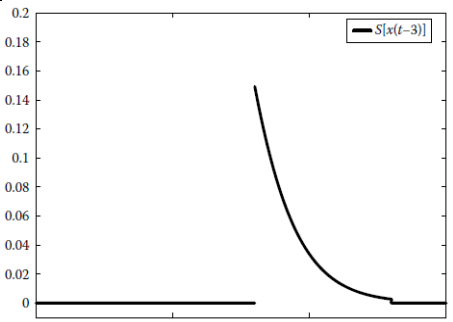
- 1) Plot terlebih dahulu masukan sinyal  $x(t)=u(t)-u(t-5)$ .
- 2) Plot respon sistem  $y(t)$  dengan menggeser 3 unit ke kanan untuk merepresentasikan sinyal  $y_1(t)=y(t-3)$ .

Sisi kanan persamaan 2.3

- 1) Plot terlebih dahulu masukan sinyal  $x(t)$  yang digeser sebanyak 3 unit ke kanan untuk merepresentasikan  $x(t-3)$
- 2) Plot respon sistem  $y_2(t) = S[x(t-3)]$ .

Instruksi	Hasil	Keterangan
<pre>t = -5 : .001 : 10; p = heaviside(t)- heaviside(t-5); y = t .* exp(-t) .* p; plot (t,y) ylim ([-.05 .4]); legend ('y(t)');</pre>		<p>Respon <math>y(t)</math> pada sistem terhadap sinyal masukan <math>x(t)=u(t)-u(t-5)</math> adalah <math>y(t)=te^{-t}[u(t)-u(t-5)]=te^{-t}</math>, <math>0 \leq t \leq 5</math>.</p>
<pre>plot (t+3,y) ylim ([-.05 .4]); legend ('y(t-3)');</pre>		<p>Keluaran sinyal <math>y(t)</math> digeser 3 unit ke kanan untuk mendapatkan sinyal <math>y_1(t)=y(t-3)</math>.</p>

Sinyal masukan  $x_2(t)=x(t-3)$  didapatkan dengan  $x_2(t)=u(t-3)-u(t-8)$ . Dan respon sistem  $y_2(t)=S[x_2(t)]=S[x(t-3)]$  dikomputerisasi menjadi  $y_2(t)=te^{-t}[u(t-3)-u(t-8)]$ .

Instruksi	Hasil	Keterangan
<pre>t = -5 : .001 : 10; p = heaviside(t-3)- heaviside(t-8); y2 = t .* exp(-t) .* p; plot (t,y2) ylim ([-.01 .2]); legend ('S[x(t-3)]');</pre>		

Dari dua grafik yang diperoleh, kedua grafik tidak sama. Sehingga, hubungan i/o  $y(t)=te^{-t}x(t)$  adalah varian waktu, karena keluaran sistem tergantung pada keluaran waktu  $t$  diluar dari  $x(t)$ .

- b. Tentukan apabila  $y[n]=x^2[n]$  adalah invarian waktu. Gunakan sinyal masukan  $x[n]=0.8^n(u[n]-u[n-5])$ .

Sama dengan cara untuk waktu sinyal kontinyu, maka sinyal waktu diskrit adalah.

- 1) Plot terlebih dahulu  $y[n]$  digeser sebanyak 2 unit ke kanan, sehingga didapatkan sinyal  $y[n-2]$ .
- 2) Sinyal masukan  $x[n]$  digeser 2 unit ke kanan untuk mendapatkan sinyal  $x[n-2]$ .
- 3) Sistem respon  $S[x[n-2]]$  terhadap sinyal masukan  $x[n-2]$  dikomputerisasi dan jika sama dengan  $y[n-2]$ , maka sistem tersebut invarian waktu.

Instruksi	Hasil	Keterangan
<pre>n = 0 : 5; x = 0.8.^n; y = x.^2; stem (n,y); xlim ([-.1 5.1]); legend ('y[n]')</pre>		Respon sistem $y[n]=S[x[n]]$
<pre>stem (n+2,y) legend ('y[n-2]') xlim ([1.9 7.1])</pre>		Sinyal keluaran digeser 2 unit. $y_1[n]=y[n-2]$
<pre>n = 2 : 7; y2 = (0.8.^(n-2)).^2; stem (n,y2); xlim ([1.9 7.1]); legend ('S[x[n-2]]')</pre>		Respon sistem $y_2[n]=S[x[n-2]]$ terhadap sinyal masukan $x[n-2]$ adalah $y_2[n]=0.8^{n-2}(u[n-2]-u[n-7])$

Dalam hal ini  $y[n-2]=S[x[n-2]]$ ; sistem waktu diskrit menjelaskan hubungan  $y[n]=x^2[n]$  adalah invarian.



### 5. Sistem *Invertible* dan non-*Invertible*

Jika sistem  $S_1$  dan  $S_2$  diuraikan dengan hubungan  $y_1[n]=3x[n]$ , dan  $y_2[n]=x^2[n]$ , keduanya adalah *invertible*. Dengan masukan  $x[n]=2n$ ,  $-2 \leq n \leq 2$ .

Instruksi	Hasil	Keterangan
$n = -2 : 2;$ $x = 2*n$	$x = -4 \ -2 \ 0 \ 2 \ 4$	Komputerisasi sisi kiri dari persamaan 3.2
$y_1 = 3*x$	$y_1 = -12 \ -6 \ 0 \ 6 \ 12$	Keluaran sinyal $y_1[n]$ dari $S_1$ .
$y_2 = x.^2$	$y_2 = 16 \ 4 \ 0 \ 4 \ 16$	Keluaran sinyal $y_2[n]$ dari $S_1$ .

Dapat disimpulkan bahwa, untuk  $y_1(t)=3x(t)$  adalah *invertible*, karena sistem pertama adalah satu persatu sebagai nilai masukan yang berbeda pada nilai keluaran yang berbeda (keduanya sama). Sedangkan, untuk  $y_2(t)=x^2(t)$  tidak satu persatu yaitu nilai masukan yang berbeda dengan nilai keluaran. Sehingga, hubungan i/o  $y_2(t)=x^2(t)$  adalah tidak *invertible*.

#### Mengembalikan Sistem *Invertible*

Peroleh kembali sistem *inverse* dengan  $y_1[n]=3x[n]$  dan  $y_2[n]=x^2[n]$ .

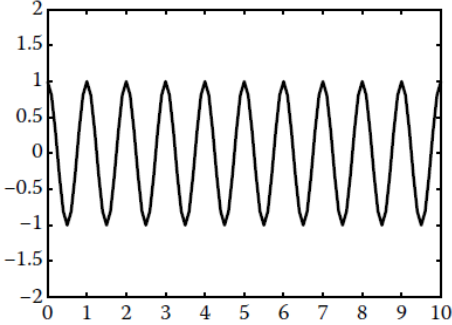
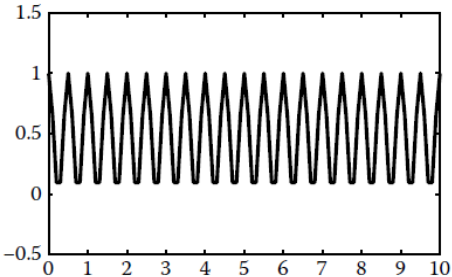
Hubungan i/o pada sistem pertama adalah  $z_1[n]=(1/3)y_1[n]$ , sedangkan pada sistem kedua (meskipun sistem ini bukan *invertible*) kita coba dengan hubungan i/o  $z_2[n]=\sqrt{y_1[n]}$ .

Instruksi	Hasil	Keterangan
$y_1 = [-12 \ -6 \ 0 \ 6 \ 12];$		Keluaran sinyal $y_1[n]=3x[n]$ digunakan sebagai masukan sistem <i>inverse</i> .
$z_1 = (1/3)*y_1;$	$z_1 = -4 \ -2 \ 0 \ 2 \ 4$	Respon $z_1[n]$ dengan hubungan i/o $z_1[n]=(1/3)y_1[n]$ dengan sinyal masukan adalah sama dengan sinyal asli $x[n]$ . maka, sistem tersebut adalah <i>invertible</i> .
$y_2 = [16 \ 4 \ 0 \ 4 \ 16];$		Keluaran sinyal $y_2[n]=x^2[n]$ digunakan sebagai masukan sistem <i>inverse</i> .

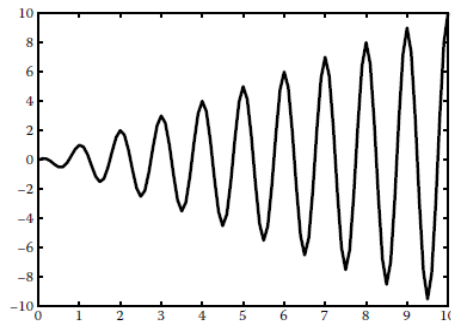
$Z2 = \text{sqrt}(y2);$	$z2 = 4 \ 2 \ 0 \ 2 \ 4$	Respon $z_2[n]$ dengan hubungan i/o $z_2[n] = \sqrt{y_1[n]}$ dengan sinyal masukan adalah tidak sama dengan sinyal asli $x[n]$ . maka, sistem tersebut adalah bukan <i>invertible</i> .
-------------------------	--------------------------	---

### 6. Membuat Sifat Sistem Stabil/Tidak Stabil

Dengan sinyal masukan  $x(t) = \cos(2\pi t)$ , digunakan pada hubungan i/o  $y_1(t) = x^2(t)$  dan  $y_2(t) = tx(t)$ .

Instruksi	Hasil	Keterangan
<pre>t = 0 : .1 : 10; x = cos(2*pi*t); plot(t,x); ylim ([-2 2]);</pre>		Grafik $x(t)$ . sinyal masukan terbatas pada $-1 \leq x(t) \leq 1$ , sehingga, $x(t)$ terbatas pada $M=1$ atau $ x(t)  \leq M$ .
<pre>y1 = x.^2; plot(t,y1); ylim ([-0.5 1.5]);</pre>		Keluaran sinyal $y_1(t)$ terbatas pada $0 \leq y(t) \leq 1$ , sehingga $y_1(t)$ dibatasi dengan $N=1$ , atau $ y(t)  \leq N$ . Sehingga sistem yang memiliki hubungan i/o $y_1(t) = x^2(t)$ adalah stabil BIBO.

```
y2 = t.*x;  
plot (t,y2);
```



Keluaran sinyal  $y_2(t)$  tidak dibatasi yaitu amplituda semakin membesar sebanding dengan bertambahnya waktu. Sehingga sistem yang memiliki hubungan i/o  $y_2(t)=tx(t)$  adalah tidak stabil BIBO.

### E. Tugas

1. Tentukan sistem yang memiliki hubungan i/o  $y(t)=\cos(x(t))$  apakah statis atau dinamis

## PERCOBAAN 7. PLANE WAVE – NORMAL INCIDENCE (AIR-DIELECTRIC-AIR)

### A. Tujuan Percobaan

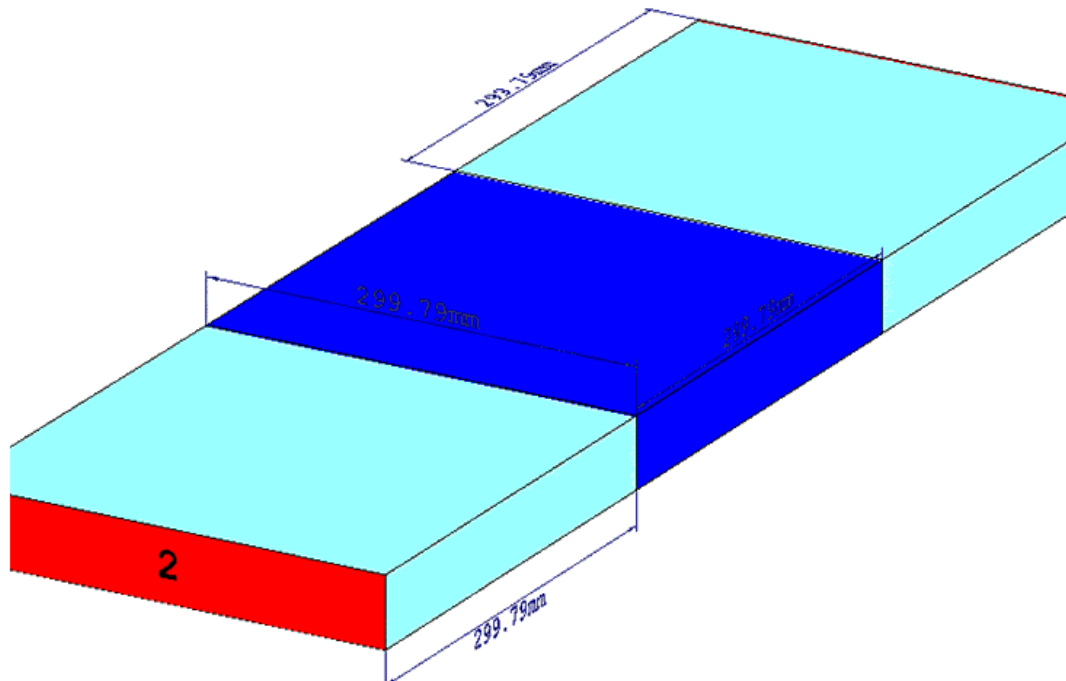
1. Mahasiswa dapat mengaplikasikan Model Elektrostatis Antara Dua Beban Bola pada Software CST EM STUDIO.

### B. Alat dan Bahan

1. Komputer
2. Software CST EM STUDIO (*Computer Simulation Technology*)

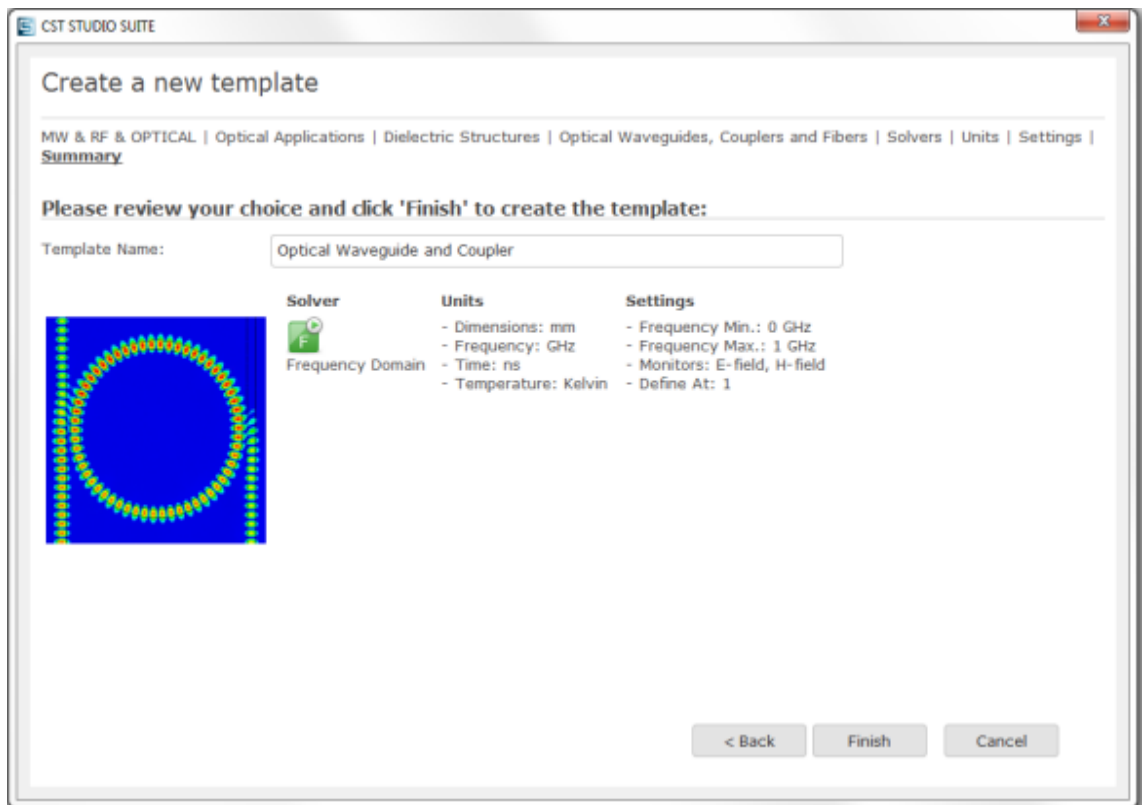
### C. Teori Dasar

Ini adalah model yang sangat sederhana di mana wave plane (yang dihasilkan menggunakan Waveguide port) dengan polarisasi linear melalui volume udara dibagi oleh slab dielektrik, seperti ditunjukkan pada Gambar di bawah ini. Permittivitas dielektrik pada parameter untuk memungkinkan koefisien refleksi harus dihitung untuk berbagai skenario yang berbeda. Model ini disimulasikan dengan solver domain frekuensi pada frekuensi 1 GHz.

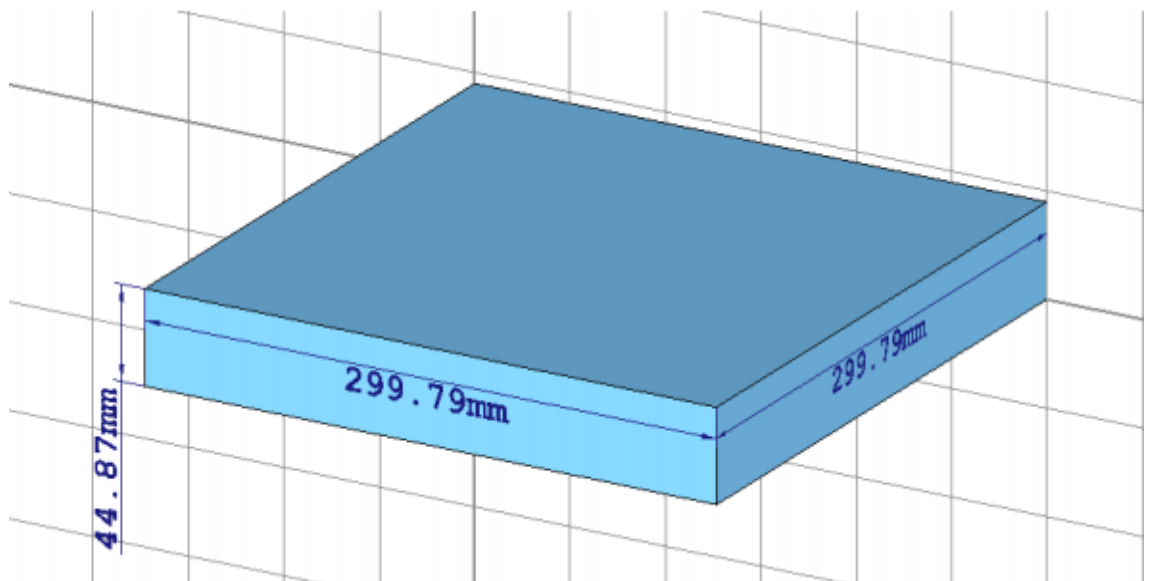


### D. Langkah Percobaan

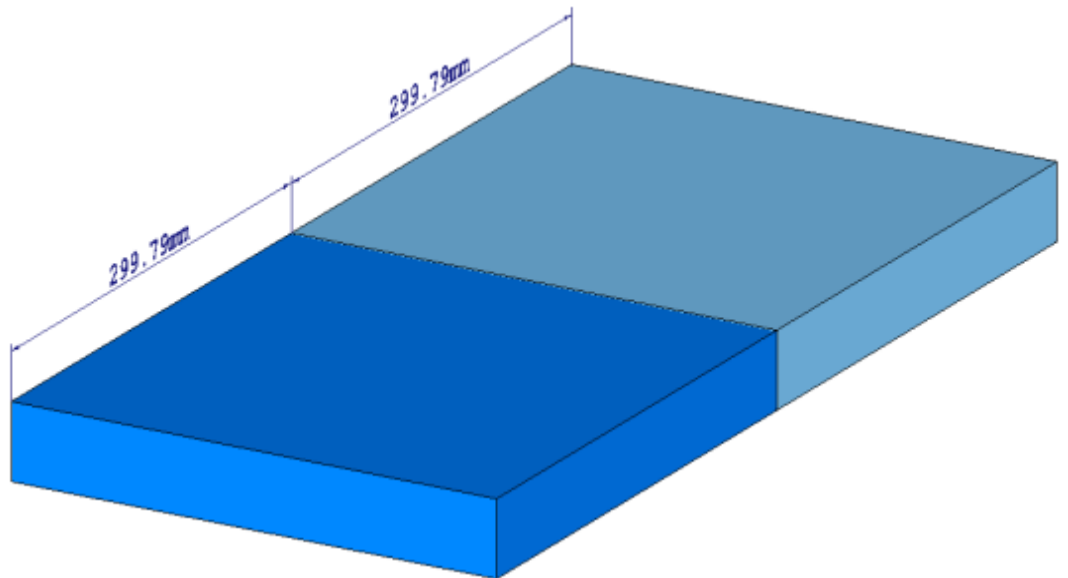
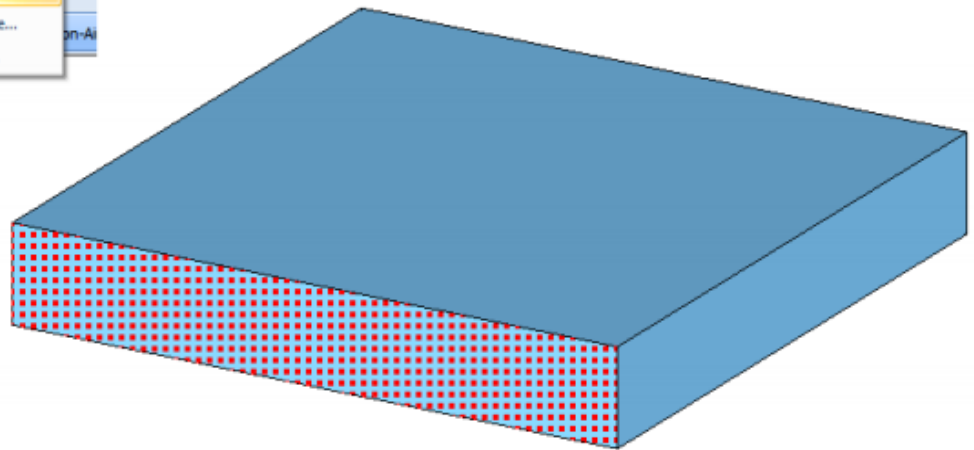
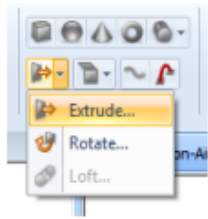
1. Mulai Proyect wizard dengan new project.



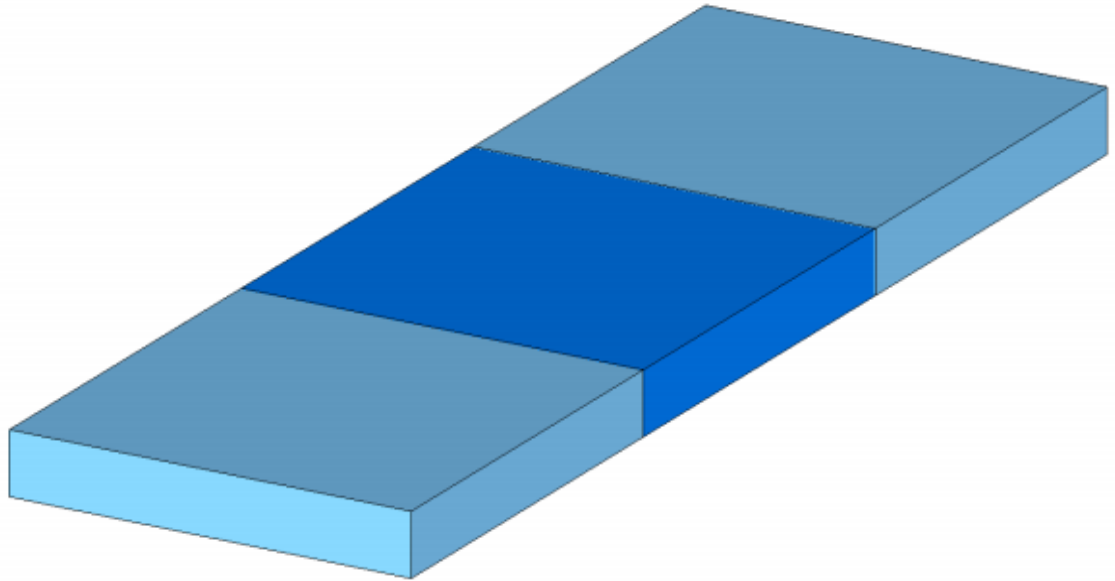
2. Pertama-tama untuk kawasan pertama. Bahan Background adalah normal. Mendefinisikan blok dengan dimensi yang ditunjukkan pada gambar di bawah ini.



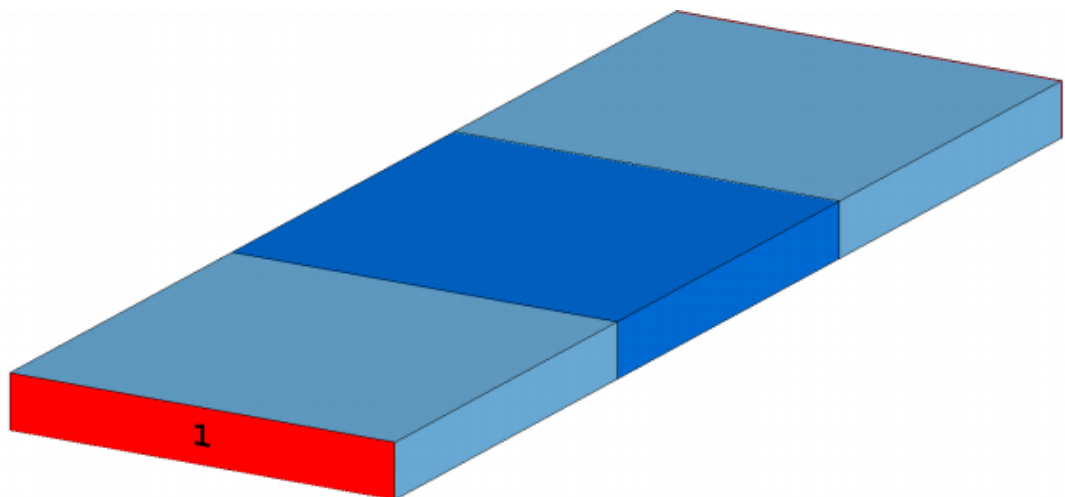
3. Kemudian klik pada pick operation dan pilih extrude. Buat kawasan dielektrik dengan panjang 299,79 mm. Buat mterial baru dengan parameter permittivity. Permittivity meruakan bahan dielektrik yang seharusnya di parameter “eps”. Dengan nilai awal 4.



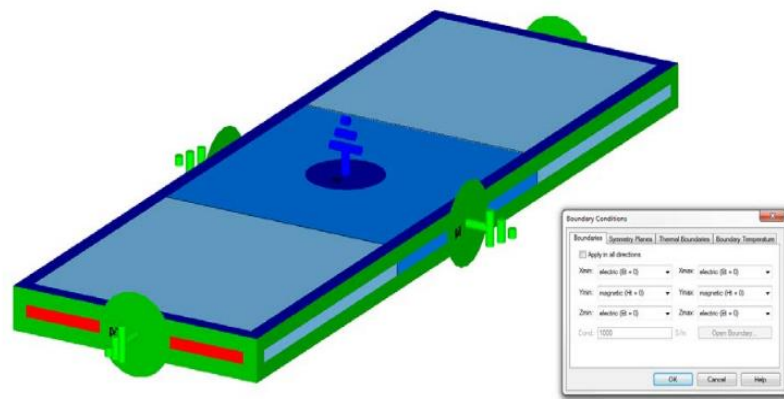
4. Lakukan lagi operasi extrude untuk membuat kawasan bidang yang kedua. Seperti gambar di bawah ini.



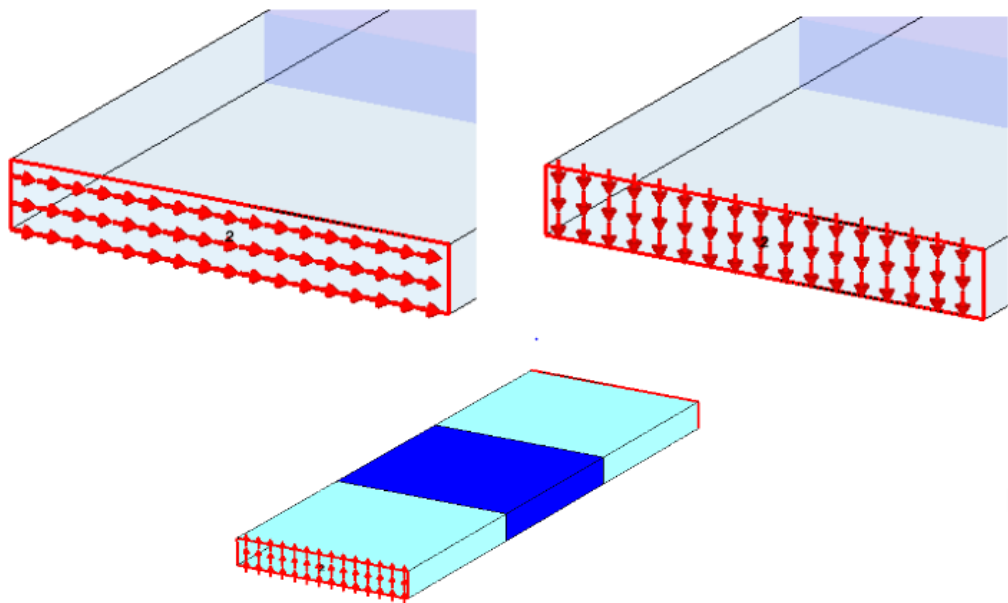
5. Untuk membuat *port waveguide* pada model. Pilih end face pada model dan pilih “Waveguide port” pada simulasi ribbon.



6. Langkah berikutnya adalah meniru Plane Wave dengan port Waveguide. Yaitu dengan cara menghasilkan TEM mode (kedua bidang E dan H yang memiliki transversal komponen ke arah propagasi). Hal ini dapat dicapai dengan menggunakan kondisi batas yang tepat. Dengan menegakkan mode TEM sehingga vektor medan magnet hanya menunjuk arah y, dan medan listriknya dititik x.

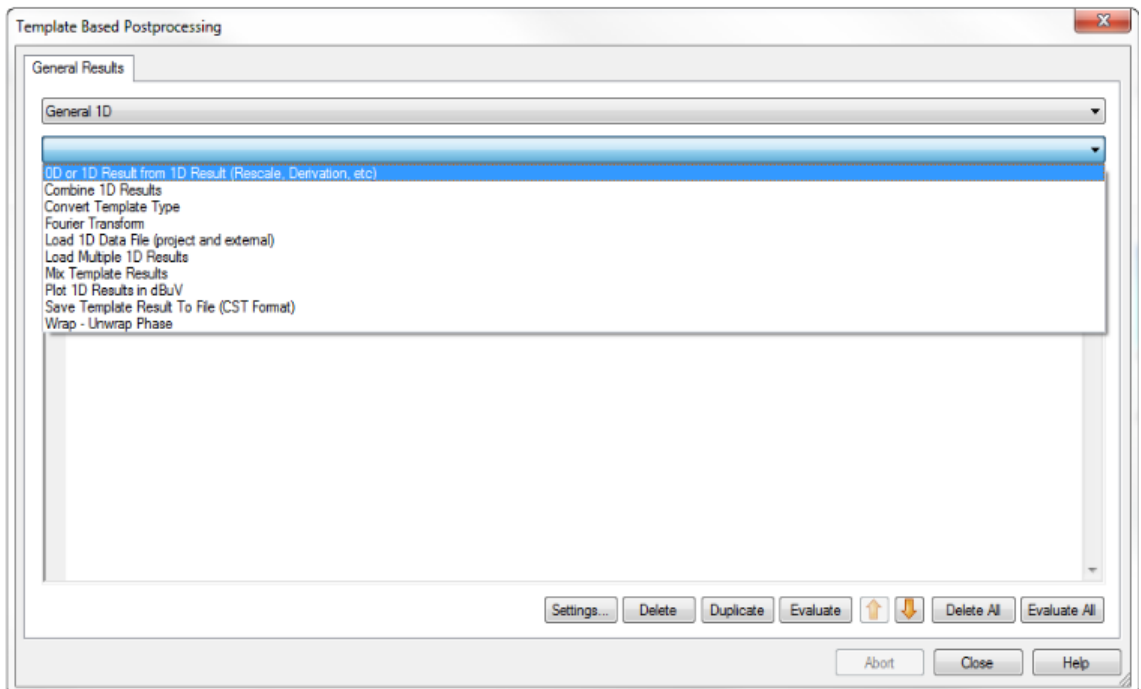


7. Dan kemudian mode TEM, port pada simulasi ditunjukkan E-Field (kiri) dan H-Field (kanan)

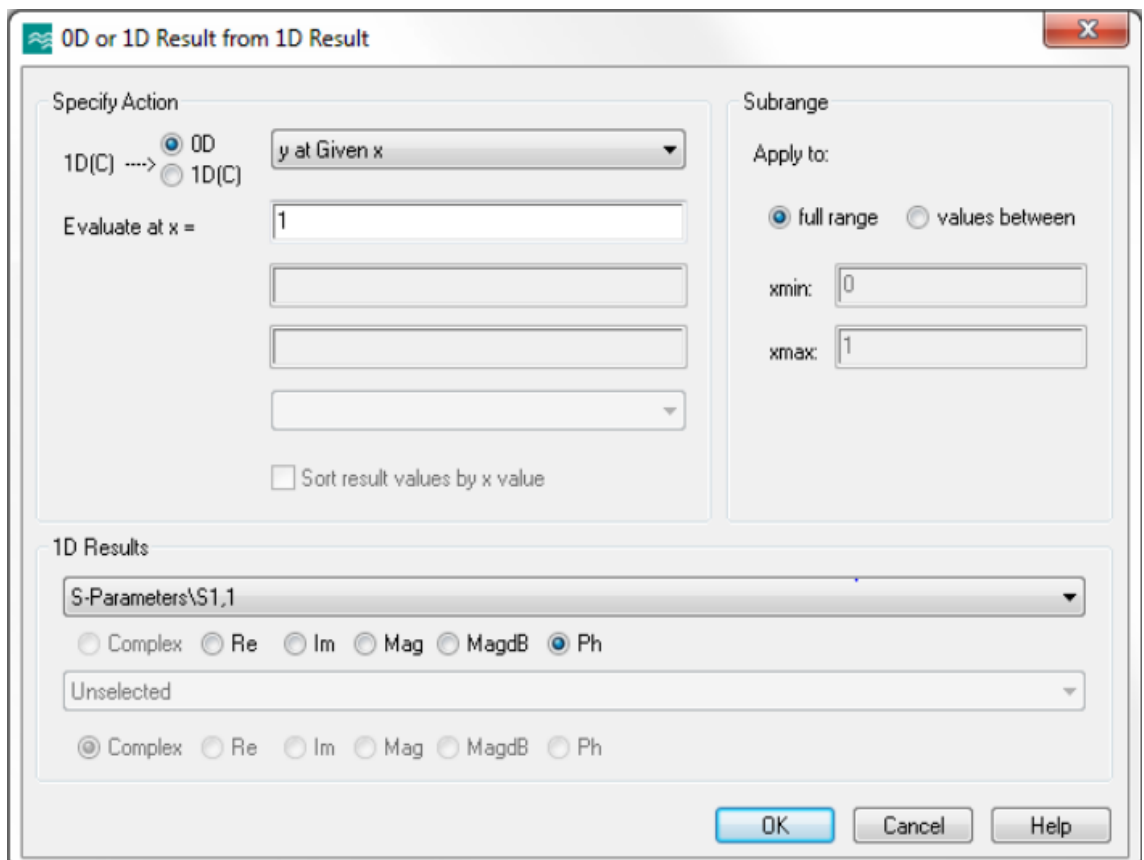


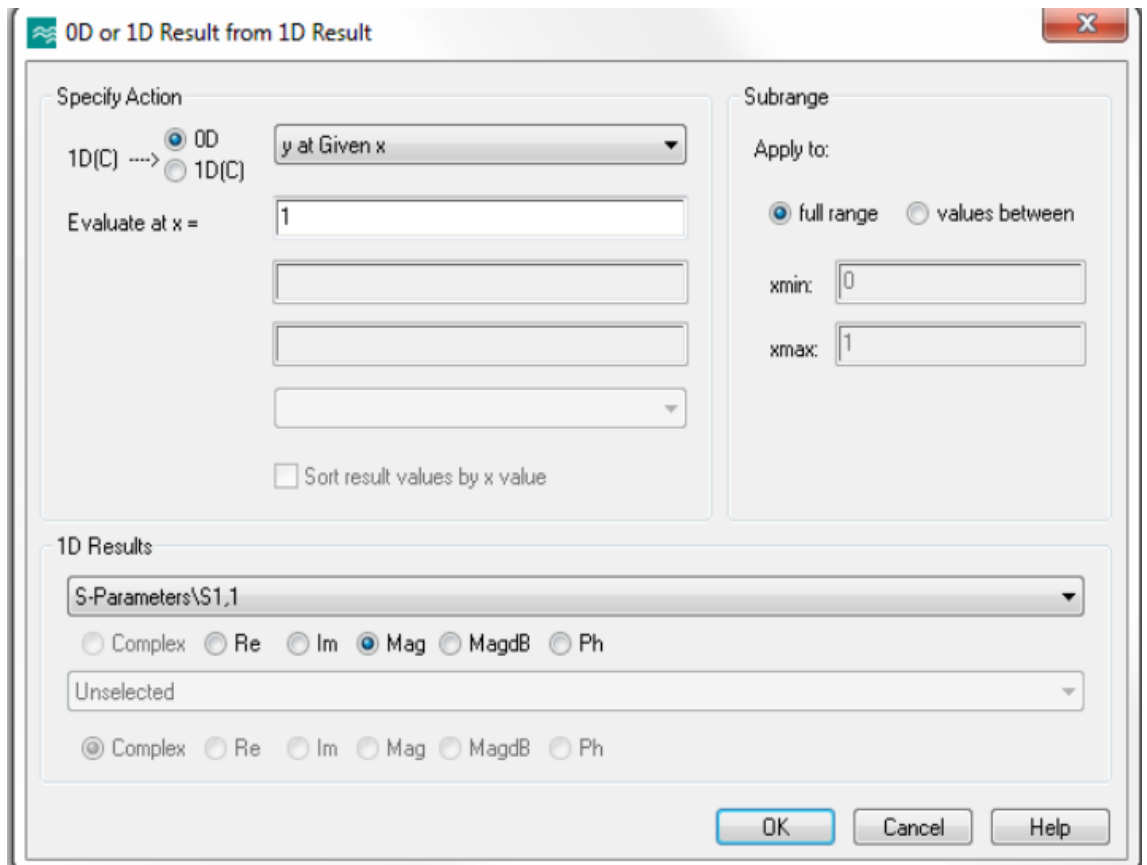
8. Pengaturan Post Processing Templates. Disini ditunjukkan bagaimana mengatur “Post Processing Template” untuk mendapatkan amplitudodan fasa koefisien refleksi. Pada “Pos Processing” pilih “Template Based Post Processing” dan kemudian cari “0D atau 1D Result dari 1D Result” pada kategori “General 1D” di bawah.





9. Tentukan hasil 0D, “Y at given x” evaluasi pada  $x = 1$  (dimana “x” adalah frekuensi). Gunakan “S-Parameter\S1,1”. Hasil untuk menemukan fasa, klik “Ph”, kemudian tentukan magnitudenya, dengan cara klik “Mag”.





10. Model ini disimulasikan dengan domain frekuensi solver. Untuk percobaan ini, kita tidak perlu melakukan broadband sweep. Sebaliknya, kita mensimulasikan hanya pada satu titik 1 GHz. Sebuah parameter sweep digunakan untuk bervariasi permetivitas (“eps”) dari dielektrik slab. Untuk melihat hasil grafik, parameter sweep di jalankan dari 1 sampai 10, dengan interval 0,25.

## PERCOBAAN 8. QUARTER WAVE TRANSFORMER

### A. Tujuan Percobaan

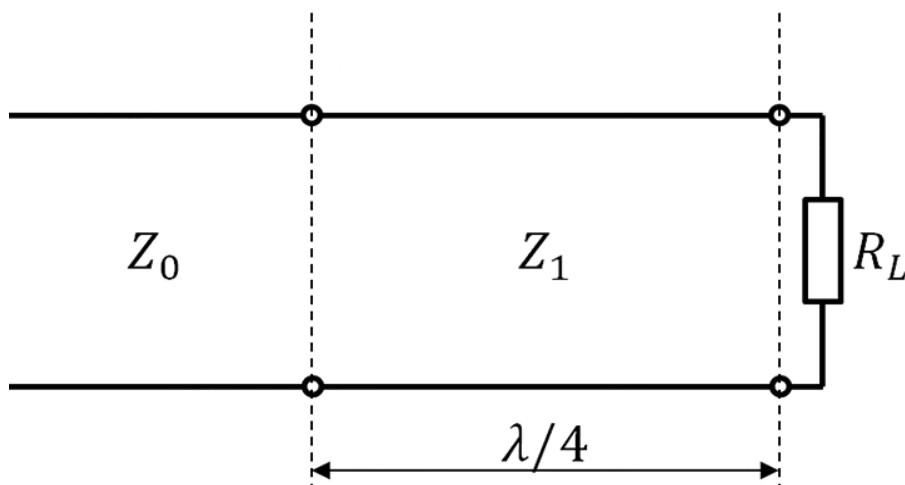
1. Mahasiswa dapat membuat dan memahami model quarter wave transformer pada program CST.

### B. Alat dan Bahan

1. Komputer
2. Software CST EM STUDIO (*Computer Simulation Technology*)

### C. Teori Dasar

Transformator seperempat gelombang adalah transformator impedansi sederhana yang umumnya digunakan dalam pencocokan impedansi untuk meminimalkan energi yang dipantulkan ketika saluran transmisi terhubung ke beban. Transformator seperempat gelombang menggunakan saluran transmisi dengan impedansi karakteristik yang berbeda dan dengan panjang seperempat dari panjang gelombang dipandu untuk mencocokkan garis dengan beban. Skema dasar ditunjukkan pada Gambar 1.



Gambar 1. Skema sirkuit transformator seperempat gelombang

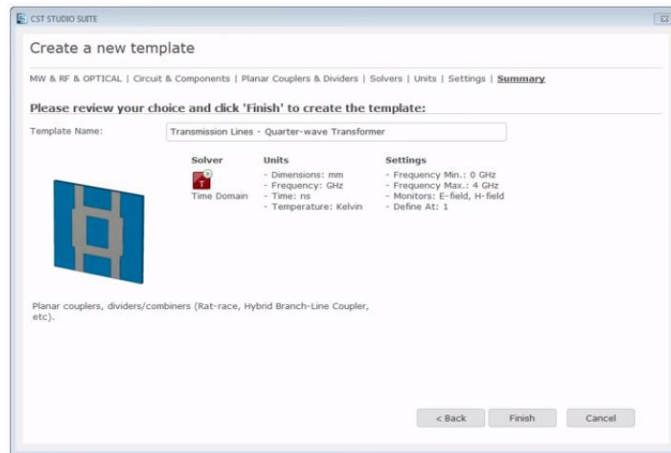
Dari literatur [1] kami menulis persamaan yang menentukan nilai impedansi  $Z_1$

$$Z_1 = \sqrt{Z_0 R_L}$$

Dari impedansi ini, lebar mikrostrip dapat dihitung dengan persamaan yang lebih kompleks [1]. Panjang gelombang terpandu  $\lambda$  lebih pendek dari panjang gelombang ruang bebas karena adanya substrat dielektrik. Persamaan untuk panjang gelombang dipandu juga diberikan dalam.

## D. Langkah Percobaan

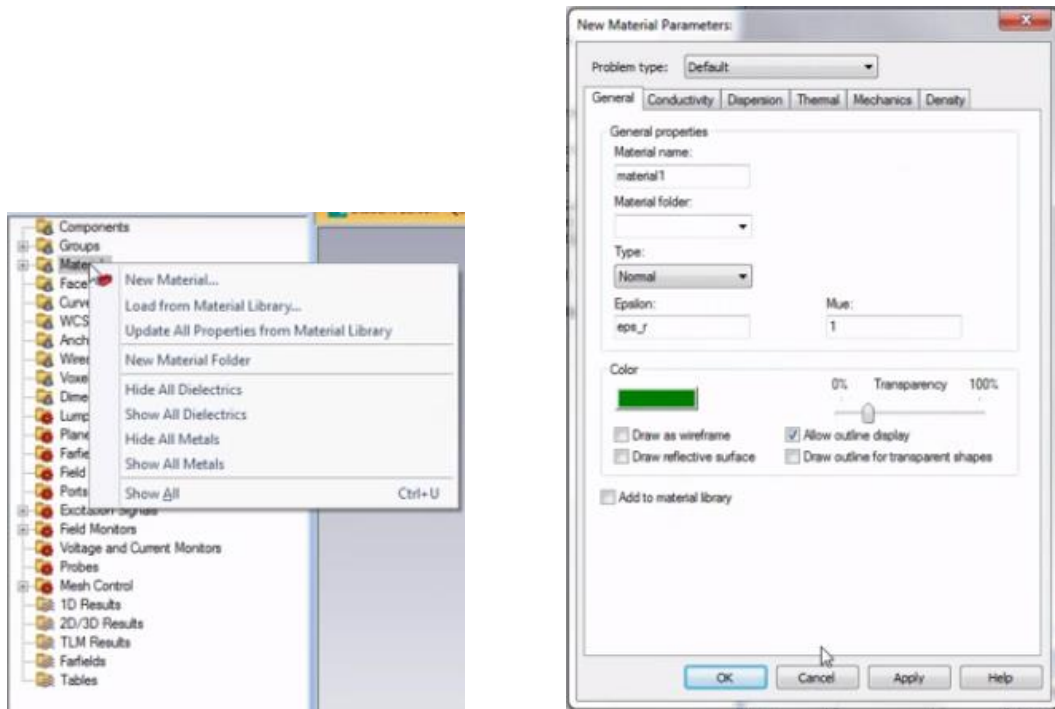
1. Buat template seperti pada gambar di bawah ini:



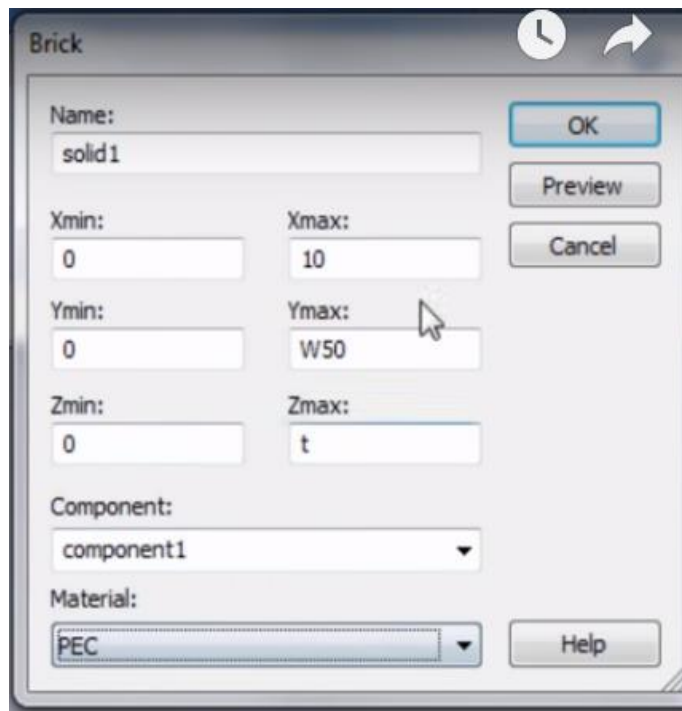
2. Input parameter dengan seperti gambar di bawah ini, klik tombol yang di tandai merah untuk memasukan parameter

Name	Expression	Value	Description	Type
eps_r	= 4.3	4.3	Substrate permittivity	Undefined
h	= 1.2	1.2	Substrate thickness	Undefined
l70	= 41.84	41.84	Transformer length	Undefined
RL	= 100	100	Load impedance	Undefined
t	= 0.035	0.035	Metallization thickness	Undefined
W50	= 2.35	2.35	50 Ohms line width	Undefined
W70	= 1.23	1.23	70.71 Ohms line width	Undefined
Z0	= 50	50	System characteristic impedance	Undefined

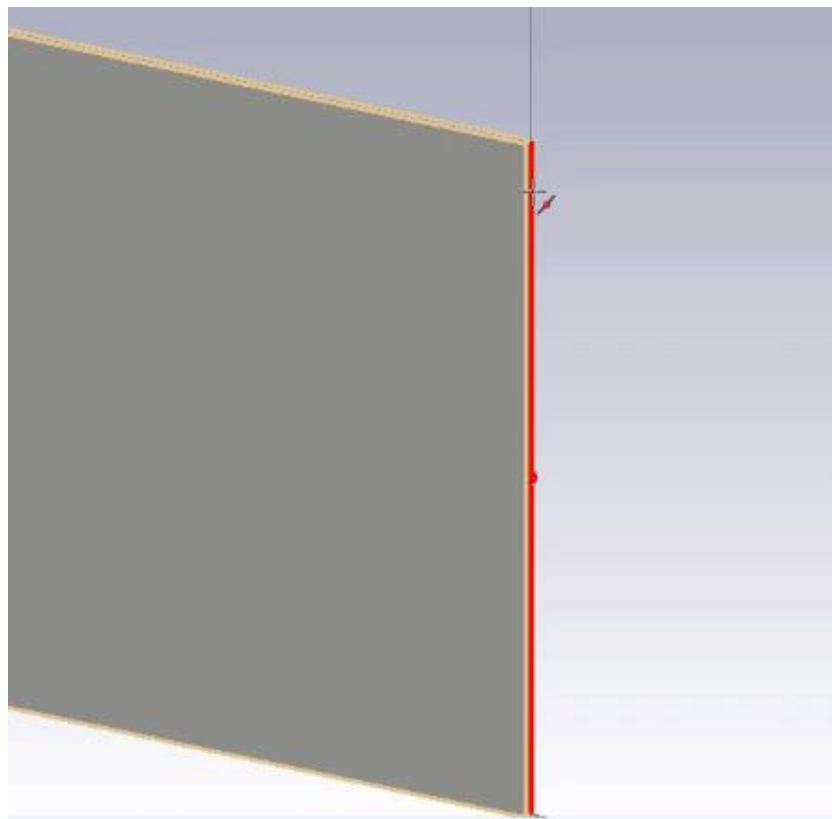
3. Klik material, dan buat material baru seperti gambar di bawah ini dan klik ok



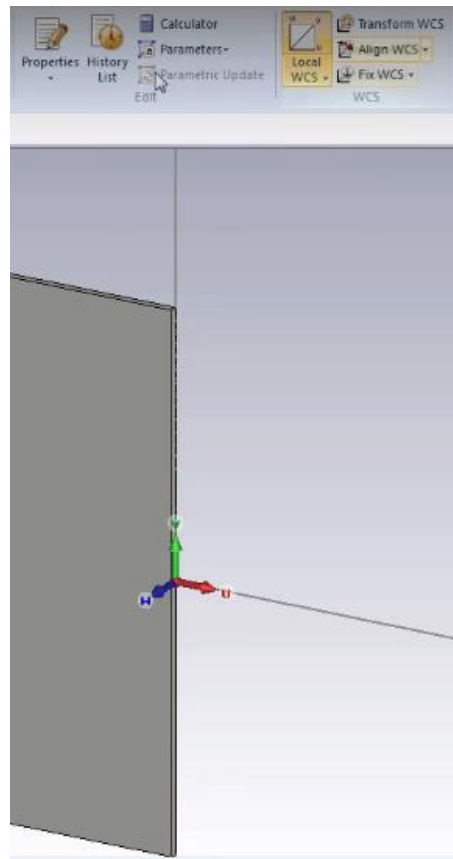
4. Buat model Brick dengan memasukan parameter seperti pada gambar di bawah ini lalu klik ok



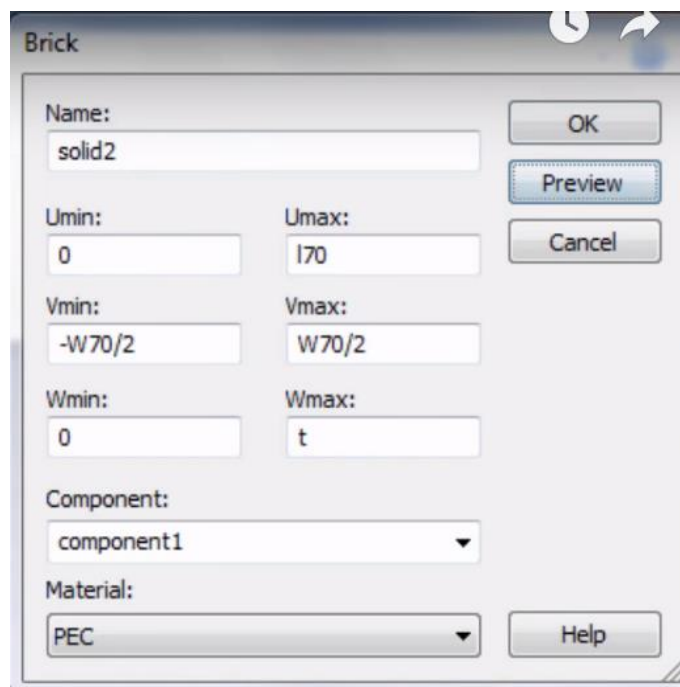
5. Pick point pada bidang seperti pada gambar di bawah ini pastikan pick pada  $z = 0$



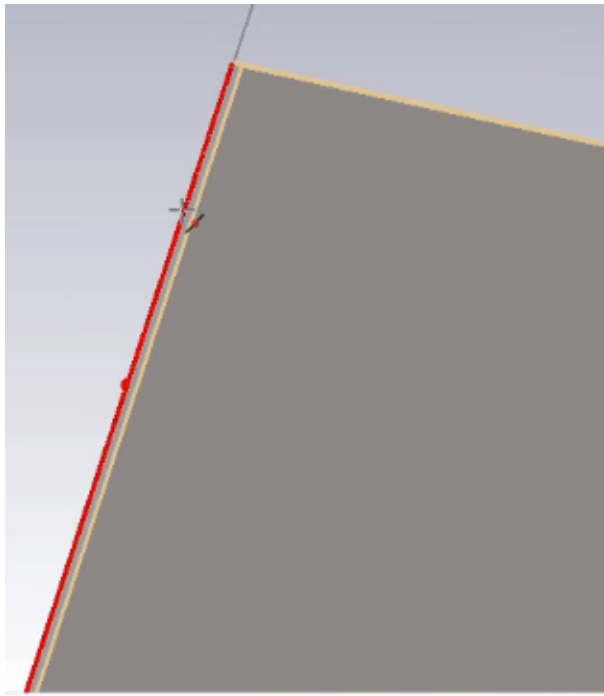
6. Klik align WCS dan hasilnya akan terlihat pada gambar di bawah ini



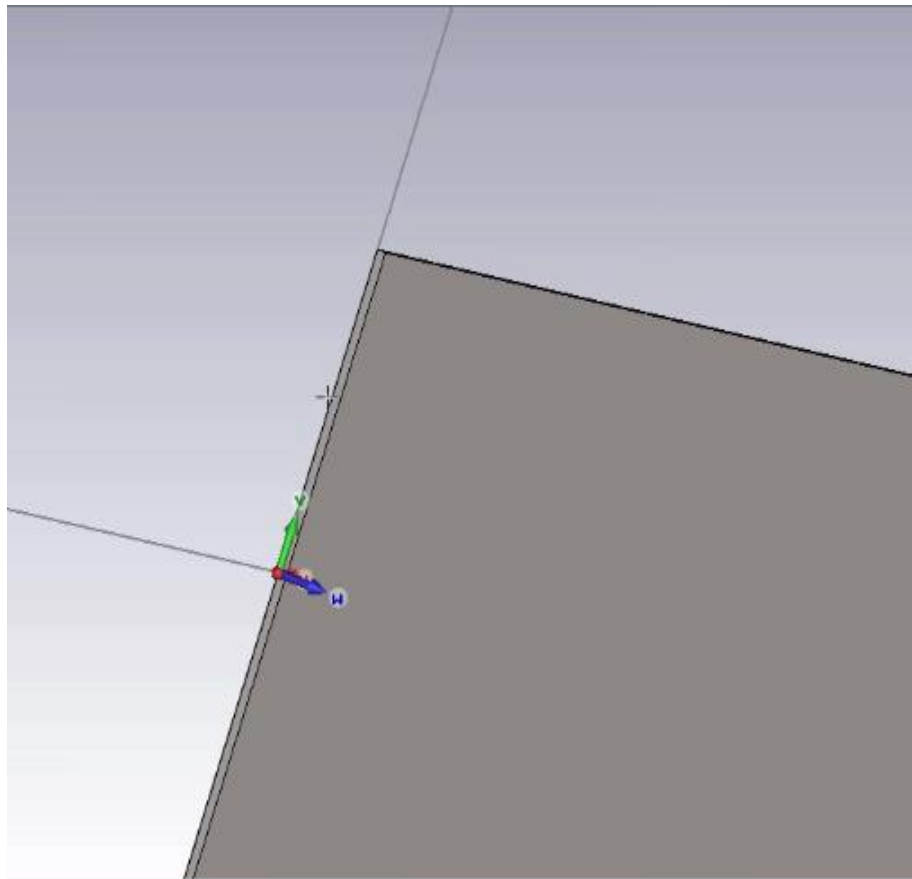
7. Klik kembali brick lalu tekan tombol ESC dan samakan parameter seperti pada gambar di bawah ini



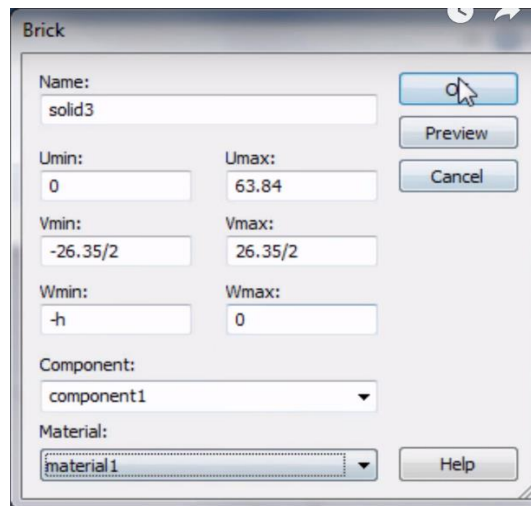
8. Pick point kembali pada brick pertama (solid 1) pada  $z = 0$ , kemudian klik local WCS.



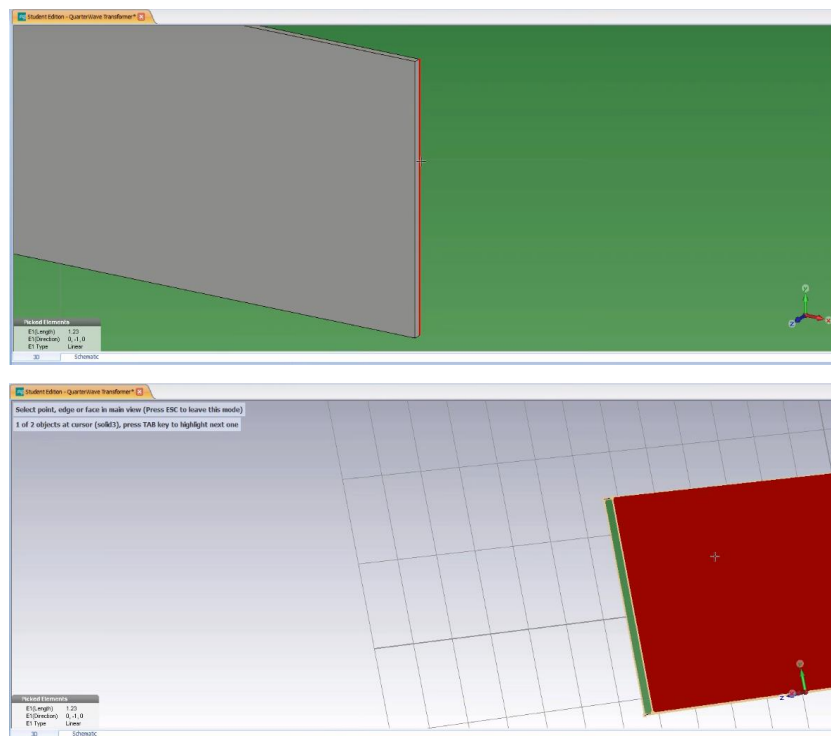
Hasil setelah diberi local WCS



9. Klik brick sesudah itu klik ESC dan masukan parameter seperti pada gambar di bawah ini

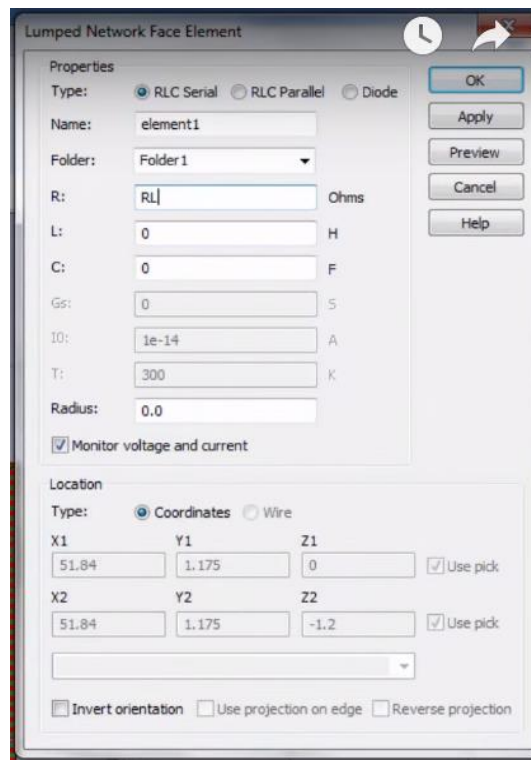


10. Pick pada brick 2 (solid 2) lalu pick face kembali bagian belakang brick 3 (solid 3) seperti pada gambar di bawah ini

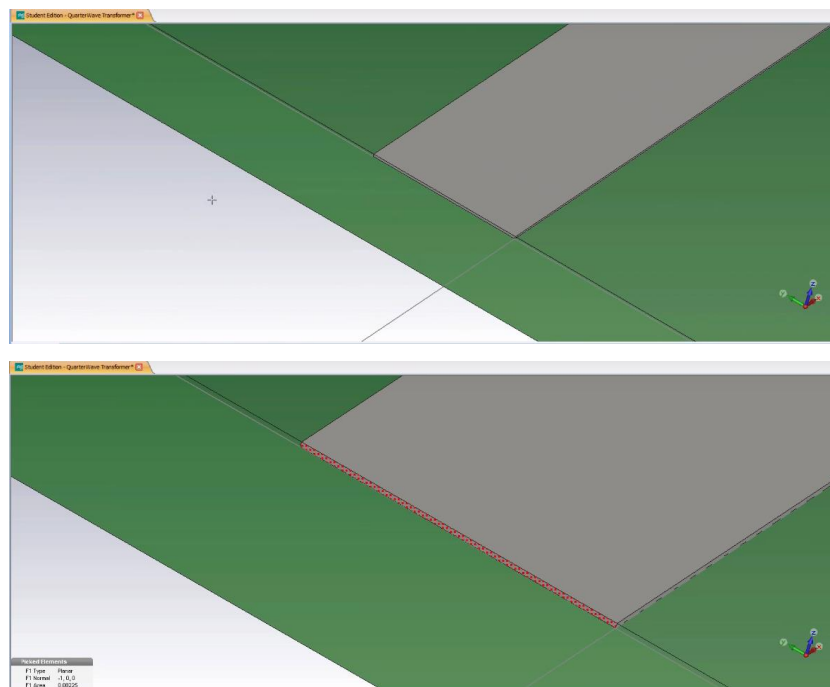


dan klik Lumped element, isi parameter seperti pada gambar di bawah ini

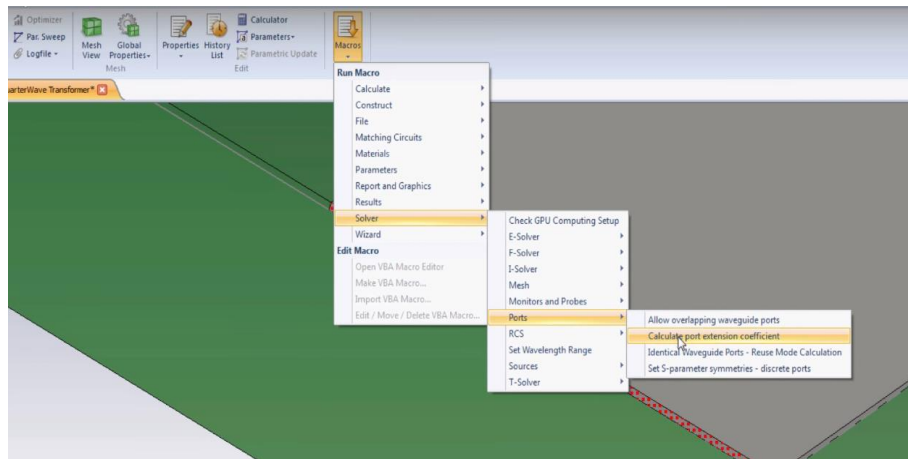




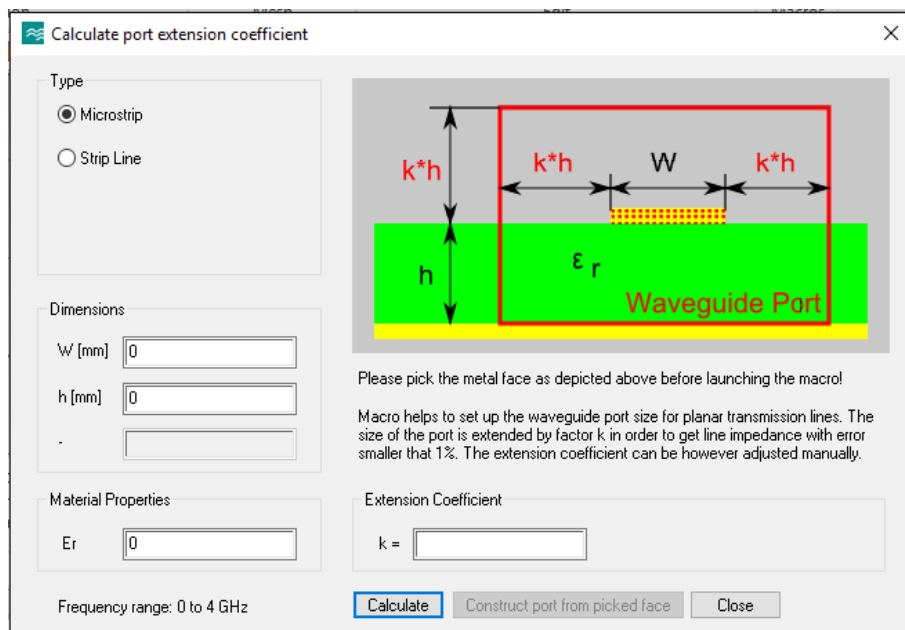
11. Klik pick lalu pick face bagian dari brick 1 (solid 1)



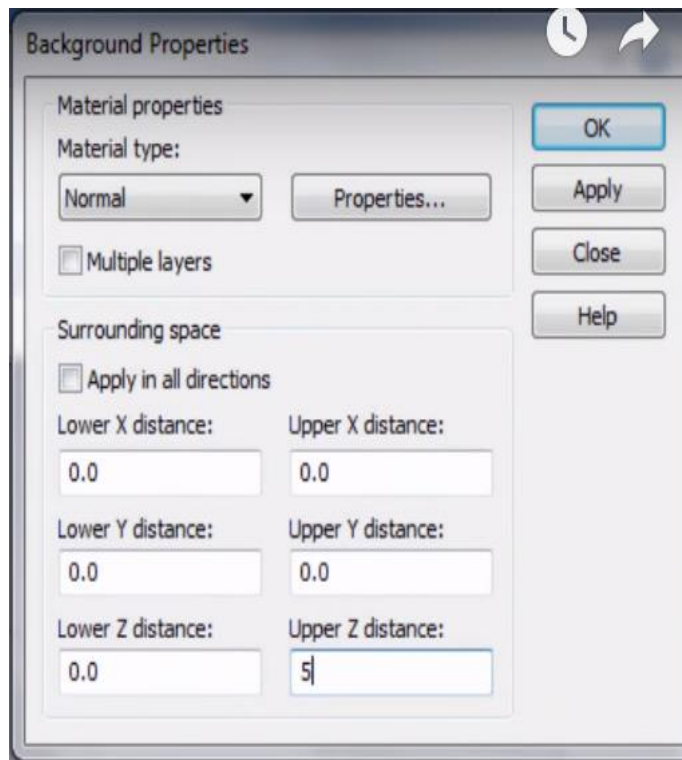
12. Klik kanan dan klik macros



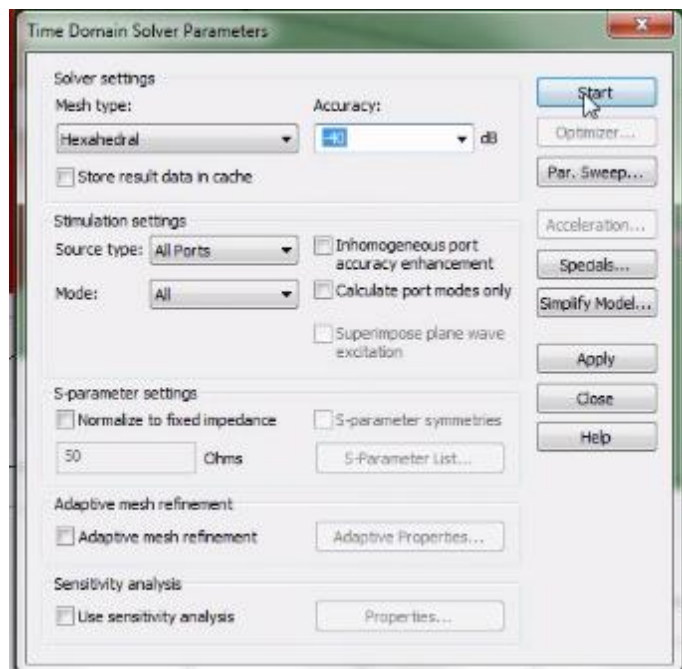
13. Samakan parameter macros dengan gambar di bawah ini, lalu klik calculate dan construct port from piked face



14. Klik background dan ubah parameter seperti gambar di bawah ini klik ok



15. Klik setup solver lalu ubah accuracy menjadi -40dB



16. Lihat dan analisis hasilnya (s-parameter, E-field (Abs))

### E. Tugas

1. Apa arti dari s-parameter dan E-field (Abs) di atas?
2. Lihat pada frekuensi 0.5 GHz dan 2 GHz?

## PERCOBAAN 9. WAVEGUIDES – RECTANGULAR WAVEGUIDE

### A. Tujuan Percobaan

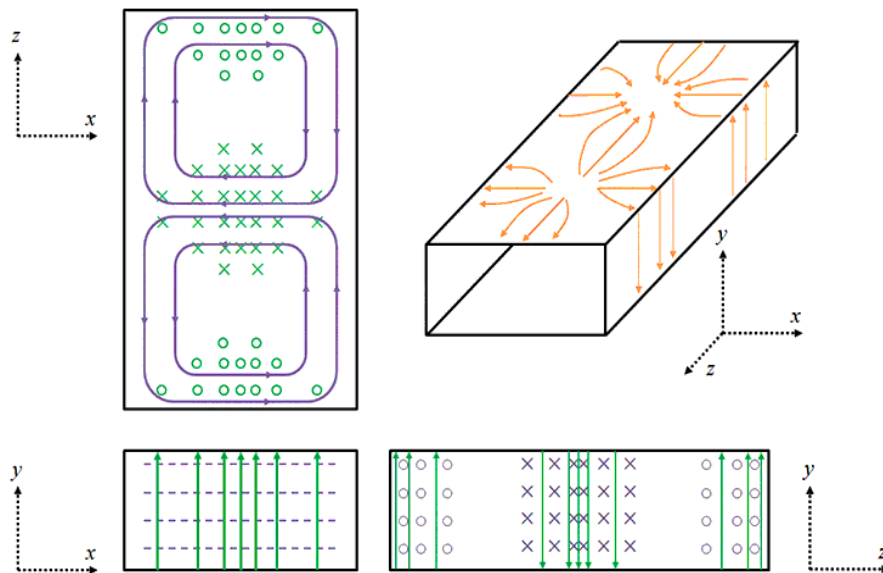
1. Mahasiswa dapat membuat dan memahami model rectangular waveguide menggunakan software CST EM STUDIO.

### B. Alat dan Bahan

1. Komputer
2. Software CST EM STUDIO (*Computer Simulation Technology*)

### C. Teori Dasar

Sebuah Waveguide berongga adalah saluran transmisi yang terlihat seperti sebuah pipa logam kosong. Mendukung propagasi *Transverse Electric* (TE) dan mode *transverse magnetik* (TM). Ada jumlah mode yang tidak terbatas yang dapat mempropagasi selama frekuensi operasi diatas frekuensi cut off. Notasi  $TE_{mn}$  dan  $TM_{mn}$  biasanya digunakan untuk menunjukkan jenis gelombang dan tipe gelombang, di mana  $m$  dan  $n$  adalah jumlah model dalam arah horisontal dan vertikal masing-masing. Mode dengan frekuensi cutoff terendah disebut model dasar atau model dominan. Untuk Rectangular Waveguide berongga mode dominan adalah  $TE_{10}$  dan E nya, bidang H dan J ditunjukkan pada Gambar di bawah ini.



Analisis elektromagnetik dari Rectangular Waveguide diketahui, dan dapat dengan mudah ditentukan dalam literatur, seperti [1]. Di sinidaftar hanya hasil akhir yang dapat digunakan untuk memverifikasi hasil simulasi.

$$f_{c_{mn}} = \frac{c}{2\pi} k_{c_{mn}} \text{ (Eq. 1)}$$

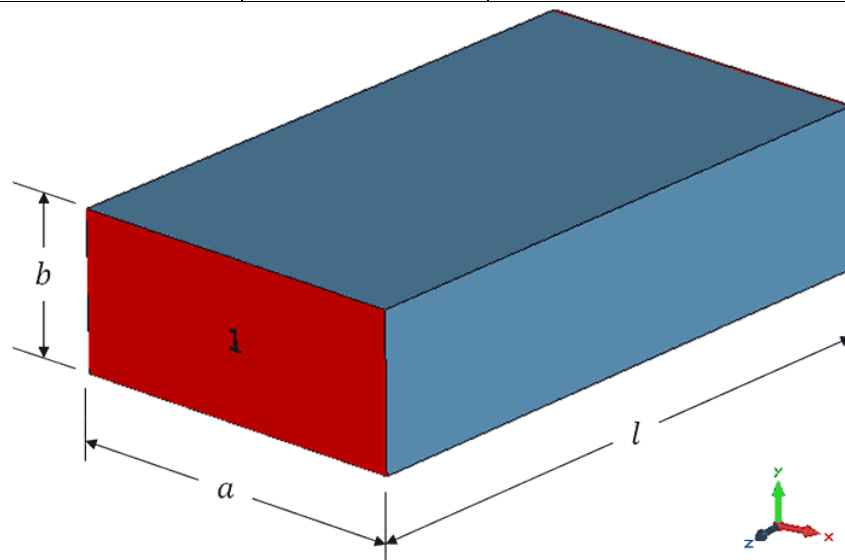
$$k_{c_{mn}} = \sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2} \text{ (Eq. 2)}$$

$$\beta_{mn} = \sqrt{k^2 - k_{c_{mn}}^2} \text{ (Eq. 3)}$$

- FC<sub>mn</sub> : frekuensi cutoff dari mode mn
- β<sub>mn</sub> : konstanta propagasi sesuai dengan mode mn
- α<sub>mn</sub> : pelemahan konstan sesuai dengan mode mn
- k : ruang bebas bilangan gelombang
- k<sub>cmn</sub> : bilangan gelombang yang sesuai dengan mode mn
- a dan b : lebar dan tinggi Waveguide masing-masing

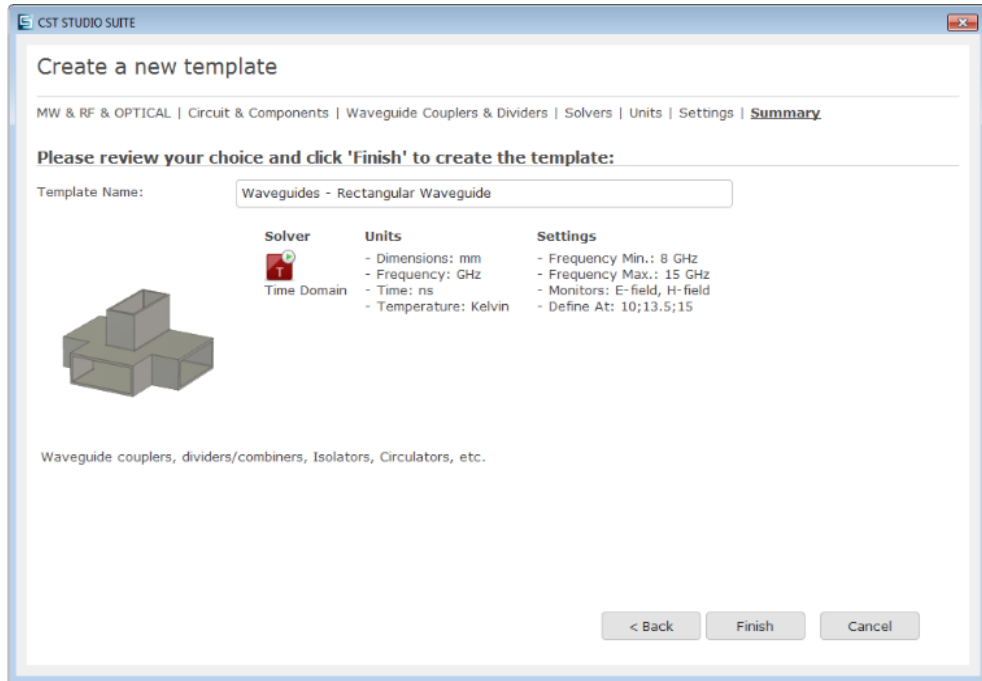
Bagian dari model Rectangulah Waveguide dalam CST STUDIO Suite® dan 3 mode pertama dihitung dan distribusi analisis medan. Dimensi yang digunakan adalah standar untuk WR-90 Waveguide. Karena *background* diatur untuk material *perfectly electrical conductor* (PEC), kita hanya perlu model hampa udara dalam Waveguide, dengan port Waveguide di setiap akhir. Kondisi batas "listrik" ke segala arah, dan model disimulasikan menggunakan *time domain solver*. Dalam model ini 3 mode pertama dihitung, dan E dan monitor H-bidang yang set-up di 10, 13,5 dan 15 GHz.

Parameter	Nilai	Deskripsi
A	22.86 mm	Dimensi Big Edge
B	10.16 mm	Dimensi Small Edge Dimension
L	40	Panjang Waveguide



#### D. Langkah Percobaan

1. Model ini hanya dibangun *vacuum brick* dengan dimensi dalam Rectangular Waveguide WR90, karena material *background* secara otomatis pengaturan oleh template adalah PEC. The "Waveguide Couplers & Divide " project template yang harus digunakan, dengan rentang frekuensi dan monitor ditentukan pada Gambar dibawah ini.

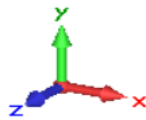
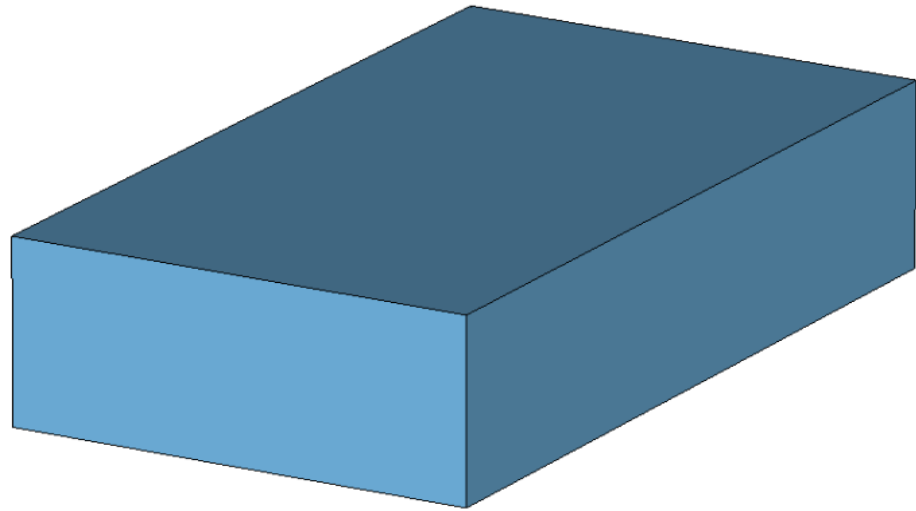


Ringkasan dari template proyek yang digunakan dalam model ini

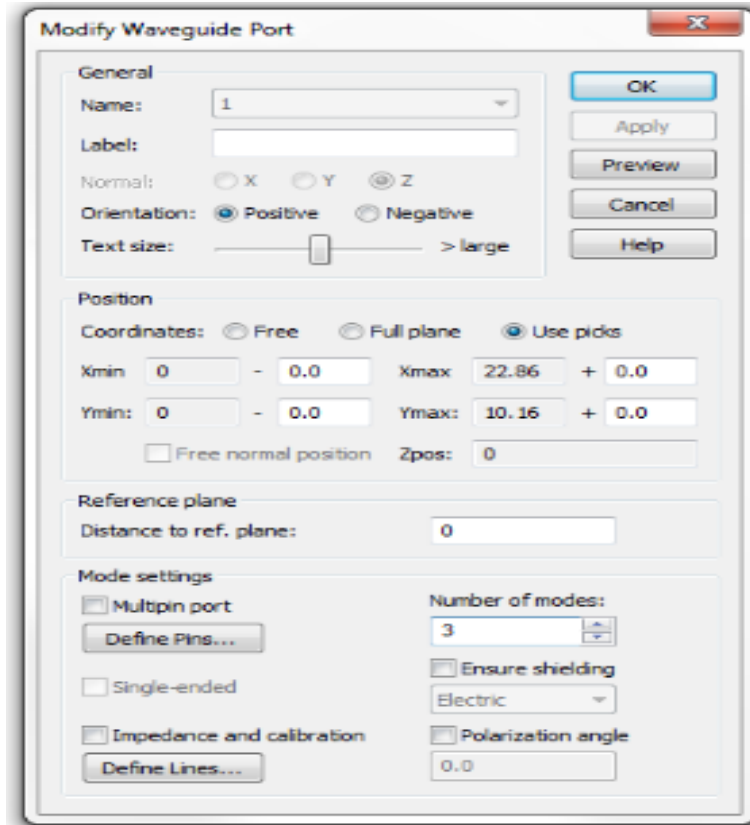
2. Pada Tabel adalah parameter yang akan digunakan dalam model ini.

Parameter	Nilai	Deskripsi
A	22.86 mm	Dimensi Big Edge
B	10.16 mm	Dimensi Small Edge Dimension
L	40	Panjang Waveguide

3. Waveguide dibuat sebagai vacuum brick dengan dimensi Tabel seperti di atas.

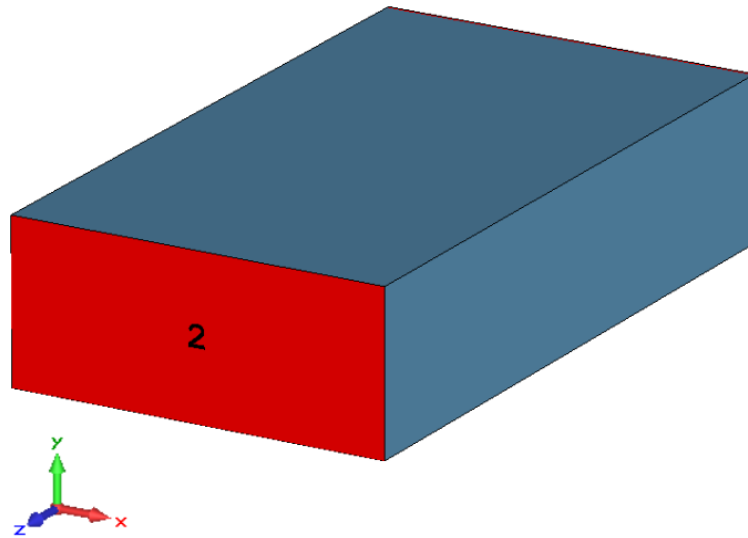


4. Port Waveguide ditempatkan di ujung kedua dari brick dengan memilih tampilan dan memilih "Waveguide Port" pada Simulasi. Ketika mendefinisikan port waveguide, itu penting bahwa jumlah mode diatur ke "3" (Gambar di bawah ini). Ini berarti bahwa mode port solver akan mengambil lebih tinggi mode orde kedua ke *account*.



Dialog box untuk pengaturan Waveguide Port

5. Model akhir akan muncul seperti yang ditunjukkan pada gambar di bawah ini.



6. Model akhir dari Hollow Rectangular Waveguide WR90



## PERCOBAAN 10. INSTALASI JARINGAN LAN

### A. Tujuan Percobaan

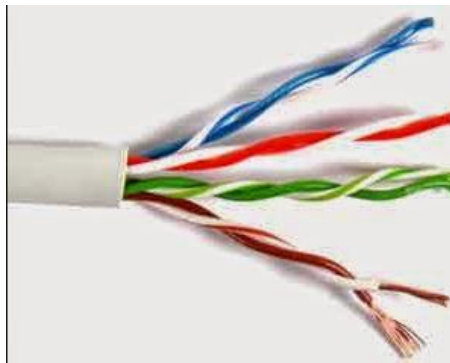
1. Mahasiswa dapat membuat kabel LAN.
2. Mahasiswa dapat memasang instalasi LAN.
3. Mahasiswa dapat mengkonfigurasi jaringan LAN.

### B. Alat dan Bahan

1. PC/Laptop dengan sistem operasi Windows (XP/7/8/8.1/10) dan memiliki LAN Card atau Ethernet Card (3 buah). Merupakan perangkat yang akan dihubungkan melalui LAN.



2. Kabel UTP CAT5 (3 x 1 meter). Media yang digunakan untuk transmisi data hingga 100 Mbps.



3. Konektor RJ45 (6 buah). Merupakan standar konektor yang dapat digunakan untuk modem atau koneksi data dalam sebuah jaringan.



4. Ethernet Switch (1 buah). Digunakan sebagai penghubung antara perangkat yang ada dalam sebuah jaringan, switch berperan sebagai sentral yang menerima data dari satu perangkat kemudian meneruskannya ke perangkat tujuan.



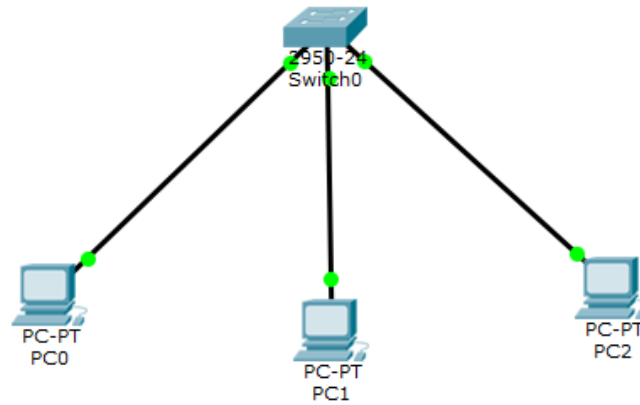
5. Tang Crimping + Cable Stripper (1 buah). Digunakan untuk memasang konektor RJ45 pada ujung kabel UTP.



6. LAN tester. Merupakan alat yang digunakan untuk melakukan pengecekan terhadap konfigurasi kabel UTP yang terpasang dengan konektor RJ45.



### C. Skema LAN (Local Area Network) yang Akan Dibangun



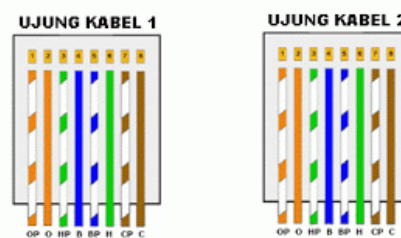
#### D. Teori Dasar

LAN (Local Area Network) merupakan jaringan computer yang mencakup wilayah kecil, seperti jaringan computer kampus, sekolah, kantor, rumah dan lainnya. LAN seringkali digunakan untuk menghubungkan komputer-komputer untuk memakai bersama suatu sumberdaya (resource) seperti printer, internet, sharing file, dan lain-lain.

#### E. Langkah Percobaan

Dalam praktikum ini ada tiga tahapan langkah kerja yaitu:

1. Menyiapkan media penghubung berupa kabel UTP dan konektor RJ45 dengan konfigurasi kabel straight through



Keterangan warna kabel:

- a. Putih Orange
- b. Orange
- c. Putih Hijau
- d. Biru
- e. Putih Biru
- f. Hijau
- g. Putih Coklat
- h. Coklat

- Tandai setiap ujung kabel sepanjang sekitar 12 mm (setengah Panjang konektor RJ45)
- Kupas setiap ujung kabel menggunakan kabel stripper, dan rapihkan ujung kabel sesuai dengan urutan warna
- Pasang konektor RJ45 pada setiap ujung kabel dengan konfigurasi straight through menggunakan tang crimping
- Test kabel UTP yang dibuat menggunakan LAN tester, untuk memastikan kabel dan konektor sudah terpasang dengan baik dan benar.

## **2. Menginstalasi PC/Laptop dengan switch menggunakan kabel UTP yang telah dibuat.**

Hubungkan Port pada Switch dan Port Ethernet card pada PC menggunakan kabel UTP.

## **3. Konfigurasi IP Address**

Setelah jaringan terinstalasi secara fisik, saatnya mengkonfigurasi jaringan dari sisi software. Dalam sebuah jaringan komputer local (LAN) setiap computer harus menggunakan IP Address dengan kelas yang sama. Pada praktikum kali ini kita akan menggunakan IP Address kelas C. Jadi, PC/Laptop pada jaringan yang kita buat akan menggunakan:

PC 1 :

IP : 192.168.1.1

Subnet Mask : 255.255.255.0

PC 2 :

IP : 192.168.1.2

Subnet Mask : 255.255.255.0

PC 3 :

IP : 192.168.1.3

Subnet Mask : 255.255.255.0

Contoh cara men-setting IP Address pada windows 10.

## PERCOBAAN 11. INSTALASI JARINGAN WLAN DENGAN CISCO PACKET TRACER

### A. Tujuan Percobaan

4. Mahasiswa dapat mengetahui software Cisco Packet Tracer.
5. Mahasiswa dapat membuat jaringan menggunakan software Cisco Packet Tracer.
6. Mahasiswa dapat mengaplikasikan jaringan WLAN dengan menggunakan software Cisco Packet Tracer

### B. Alat dan Bahan

1. Laptop
2. Software Cisco Packet Tracer
3. Modul Praktikum Teknik Telekomunikasi I

### C. Teori Dasar

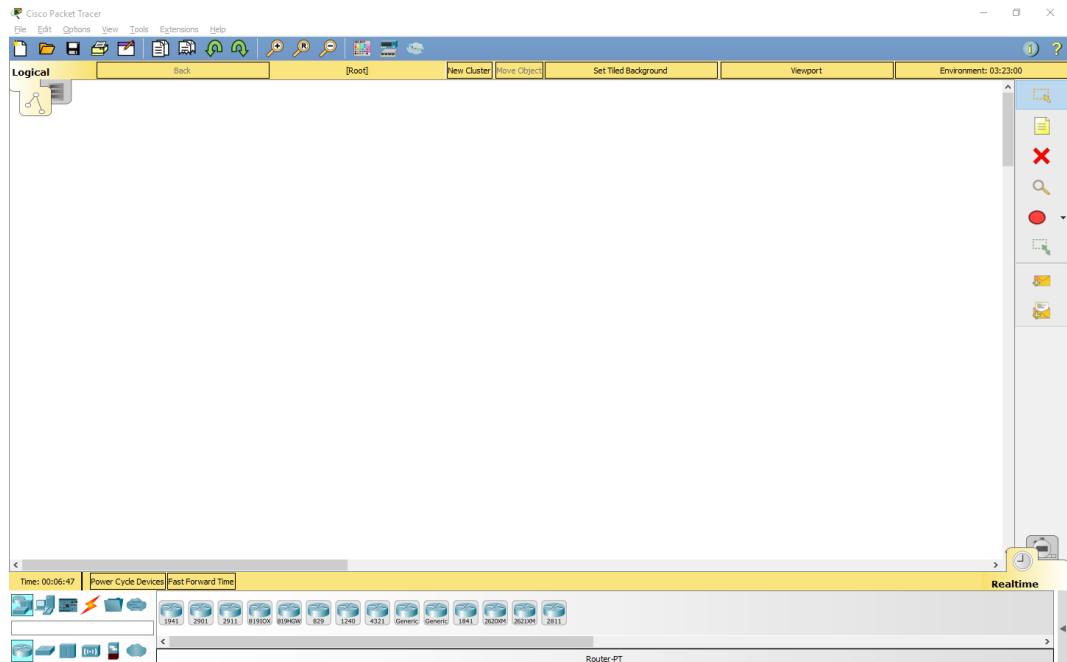
*Packet Tracer* adalah simulator alat-alat jaringan Cisco yang sering digunakan sebagai media pembelajaran dan pelatihan, dan juga dalam bidang penelitian simulasi jaringan komputer. Program ini dibuat oleh [Cisco Systems](#) dan disediakan gratis untuk fakultas, siswa dan alumni yang telah berpartisipasi di [Cisco Networking Academy](#). Tujuan utama Packet Tracer adalah untuk menyediakan alat bagi siswa dan pengajar agar dapat memahami prinsip jaringan komputer dan juga membangun skill di bidang alat-alat jaringan Cisco. Pada dasarnya Cisco Packet Tracer ini digunakan sebagai media pembelajaran bagi para pemula untuk merancang, mengkonfigurasi, dan memecahkan masalah mengenai jaringan komputer.

Singkatnya Cisco Packet Tracer memberikan kemudahan bagi kita untuk belajar bagaimana merancang, membangun dan mengkonfigurasi sebuah jaringan. mulai dari jaringan yang sederhana sampai yang kompleks. Bahkan kita juga bisa mengetahui permasalahan apa saja yang sering kali terjadi dalam sebuah jaringan hingga kita bisa menganalisa dan memperbaikinya tanpa harus membeli perangkat yang cukup mahal bagi kalangan mahasiswa yang masih dalam tahap belajar.

Untuk mengunduh *software* ini, kita bisa mendapatkannya secara gratis dengan mengunjungi link <https://www.netacad.com/courses/packet-tracer-download/> dan melakukan registrasi terlebih dahulu. Walaupun yang kita dapatkan adalah *Student Edition*, namun fiturnya sudah cukup untuk belajar. Selain itu, kita pun mendapatkan pelatihan

berupa video pembelajaran serta *file* untuk pembelajarannya yang dapat diakses secara *online*.

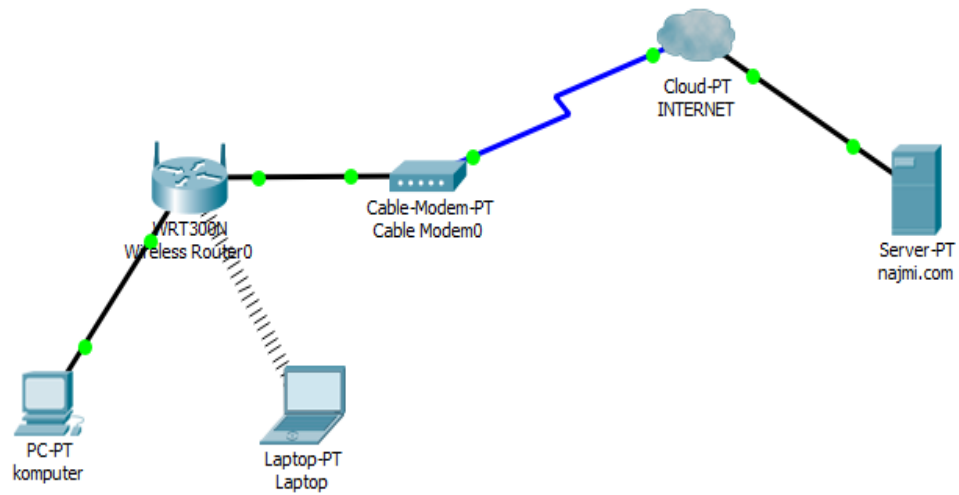
Cisco Packet Tracer pun mendukung untuk kita merancang sebuah sistem IoT, termasuk mendukung untuk simulasi dengan menambahkan beberapa sensor serta board seperti arduino. Berikut merupakan tampilan dari Cisco Packet Tracer v7.1.1



Seluruh peralatan yang akan kita pakai seperti Router, Hub, Switch, Kabel, Sensor, dan sebagainya terletak pada bagian kiri bawah, tepatnya di bawah tulisan “time”. Kita bisa memilih dengan mengklik kategori yang ada lalu pada baris bawahnya akan muncul beberapa sub-kategori. Dengan mengklik sub-kategori, maka peralatan yang dibutuhkan akan muncul di kanan berderet.

Untuk mengetahui lebih lanjut, maka akan dijelaskan dengan mensimulasikan membuat suatu jaringan sederhana (misalnya jaringan internet rumah). Dalam deskripsi, di sebuah rumah terdapat suatu wireless router yang terhubung ke cable modem dan internet, wireless router tersebut juga terhubung dengan satu computer dan satu laptop.

#### **D. Rangkaian**



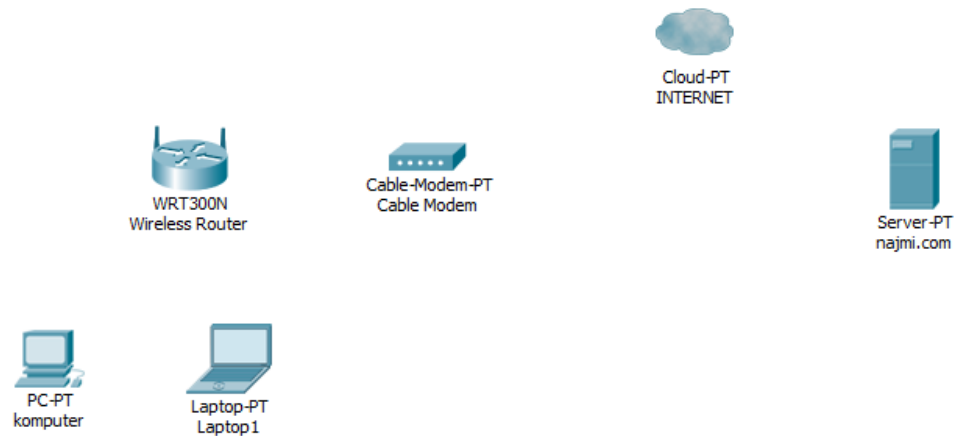
Addressing table

Device	Interface	IP Address	Subnet Mask	Default Gateway
PC	Ethernet0	DHCP		192.168.0.1
Wireless Router	LAN	192.168.0.1	255.255.255.0	
	Internet	DHCP		
najmi.com Server	Ethernet0	208.67.220.220	255.255.255.0	
Laptop	Wireless0	DHCP		

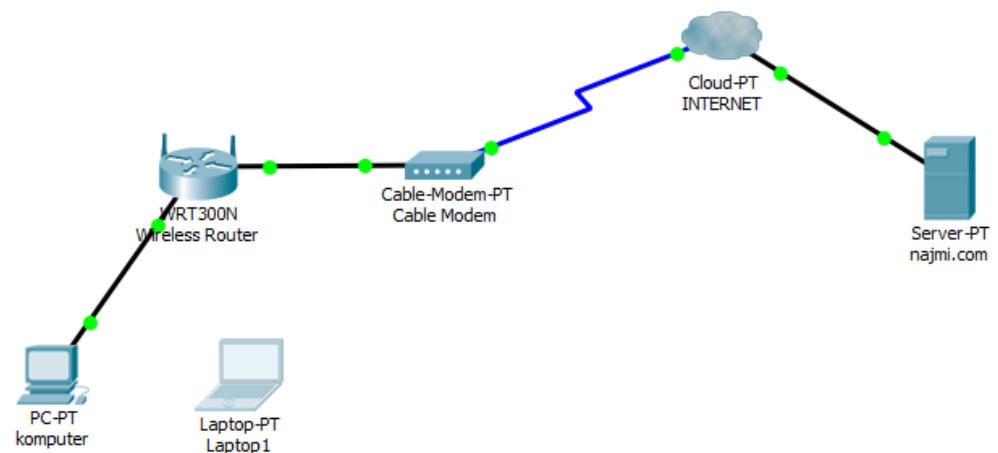
## E. Langkah Percobaan

### 1. Membuat Topologi Jaringan

- a. Menambahkan device/peralatan yang diperlukan ke lembar kerja dengan *drag* peralatan tersebut.



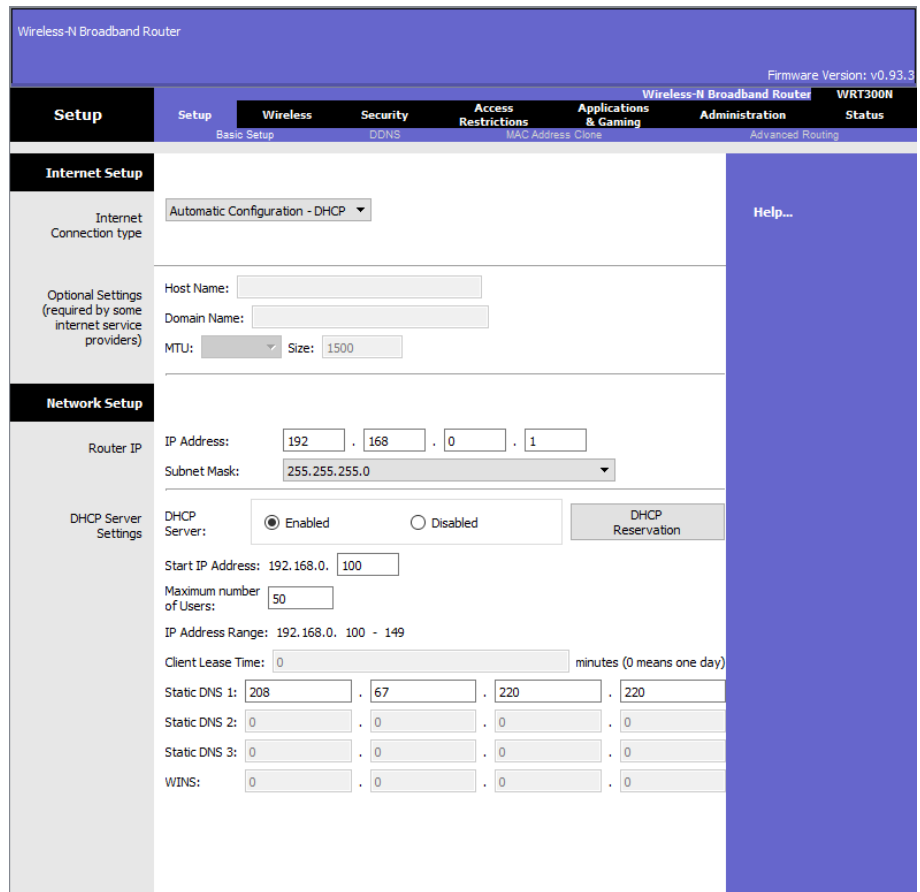
- b. Menghubungkan peralatan dengan kabel yang sesuai.
- Hubungkan dengan kabel Copper Straight-Through antara komputer (Port FastEthernet0) dengan Wireless Router(Port Ethernet1).
  - Hubungkan dengan kabel Copper Straight-Through antara Wireless Router (Port Internet) dengan Cable Modem(Port 1).
  - Hubungkan dengan kabel Coaxial antara Cable Modem (Port 0) dengan Cloud(Port Coaxial7).
  - Hubungkan dengan kabel Copper Straight-Through antara Cloud (Port Ethernet6) dengan Server(Port FastEthernet0).



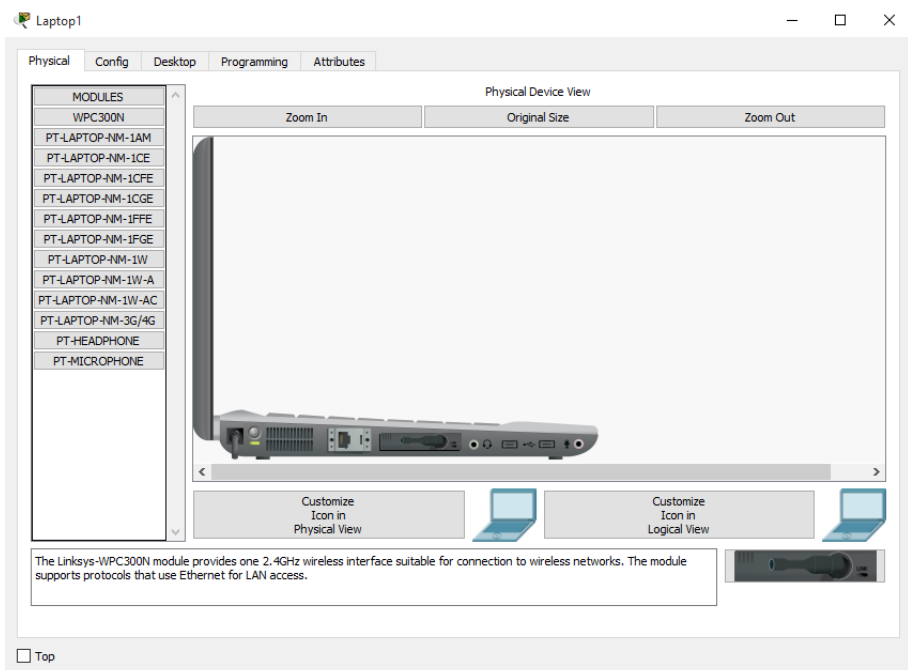
## 2. Menyusun dan Menyetel Peralatan Jaringan

- a. Menyetel Wireless Router
- Klik dua kali pada wireless router lalu pilih tab GUI.
  - Pilih tab setup. Lalu sesuaikan seperti pada gambar di bawah lalu scroll ke bawah dan pilih save setting

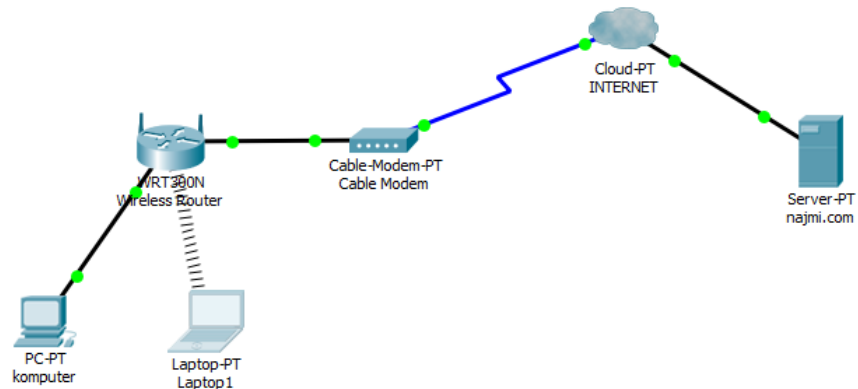




- Pilih tab wireless dan isi Network Name(SSID) sesuai keinginan lalu klik save setting
- b. Menyetel Laptop
- Klik dua kali pada laptop, pilih tab physical
  - Matikan laptop
  - Drag modul yang terpasang ke jajaran modul, lalu pasang modul WPC300N

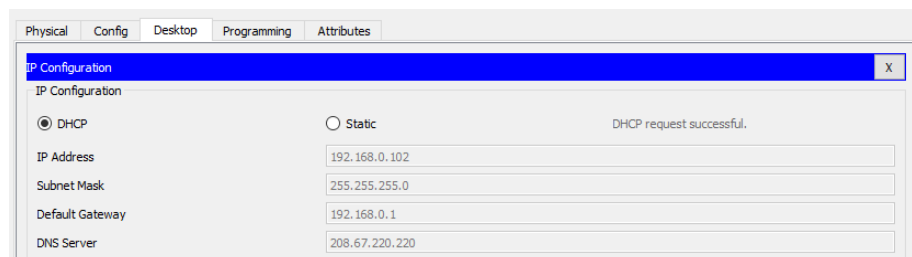


- Lalu keluar dari windows tersebut (kembali ke tampilan packet tracer)
- Klik dua kali pada laptop, lalu pilih tab Desktop kemudian PC Wireless
- Lalu pilih tab connect, jika SSID yang telah ditentukan sebelumnya belum muncul, klik refresh. Jika telah muncul klik SSID tersebut lalu klik connect disamping refresh. Maka sekarang laptop telah terhubung ke wireless router.

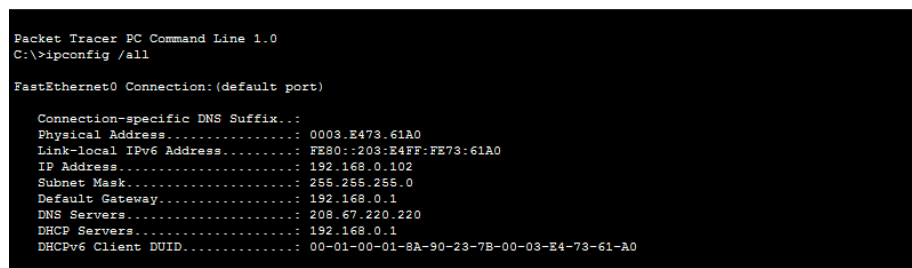


#### c. Menyetel Komputer

- Klik dua kali pada computer
- Pilih tab desktop lalu ip configuration
- Pada IP Configuration, pindahkan Dot dari static ke DHCP



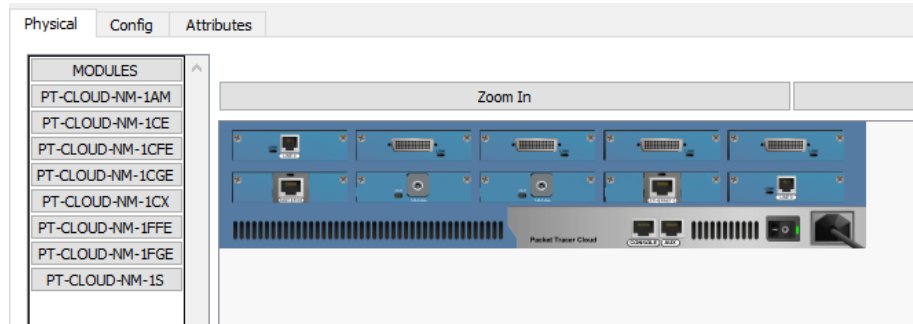
- Lalu kembali ke tab desktop (keluar dari IP Configuration)
- Klik pada command prompt
- Ketik ipconfig /all kemudian tekan Enter, untuk memastikan DHCP server telah berfungsi untuk penggunaan ip address



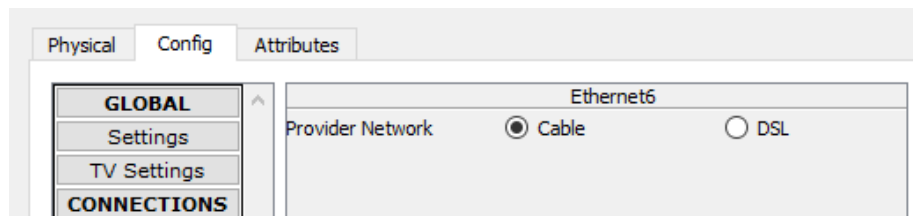
#### d. Menyusun Internet Cloud

- Klik dua kali pada cloud lalu pilih tab Physical, lalu matikan device tersebut

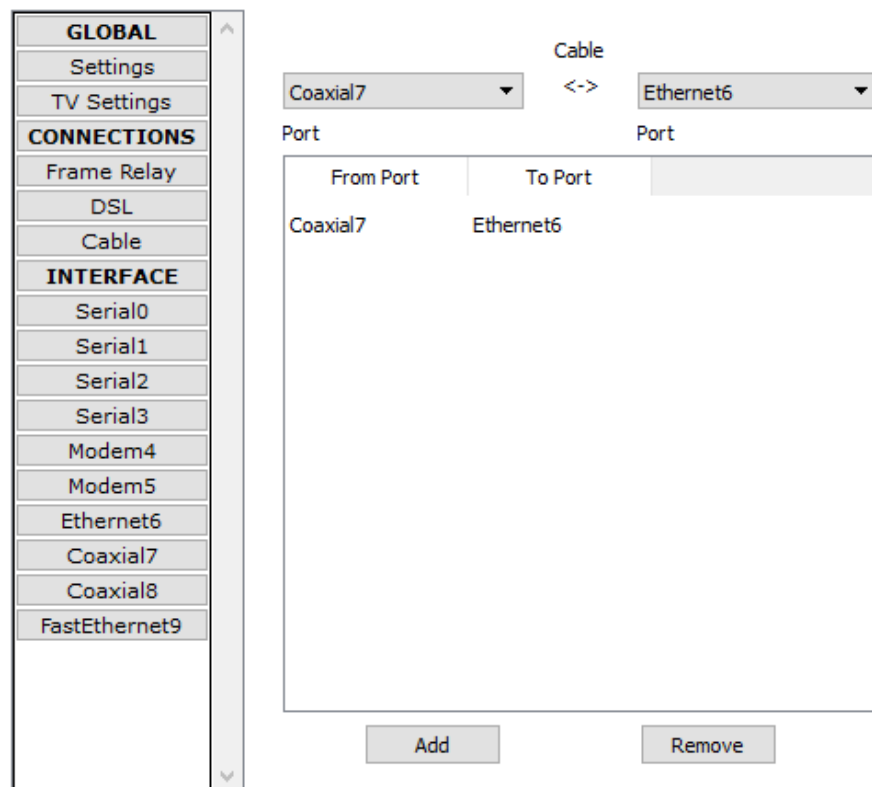
- Lalu drag modul PT-CLOUD-NM-1CFE dan PT-CLOUD-NM-1CX pada bagian device yang kosong
- Lalu hidupkan kembali device



- Lalu pilih tab config, pada barisan kiri klik ethernet6 lalu pindahkan dot ke cable



- Lalu pilih Cable pada baris kiri dibawah CONNECTIONS, pilih Coaxial7 dan Ethernet6 lalu klik Add



e. Menyetel najmi.com Server

- Klik dua kali pada Server, pilih Services tab lalu pilih DHCP pada baris kiri

- Sesuaikan isian seperti gambar di bawah lalu klik add

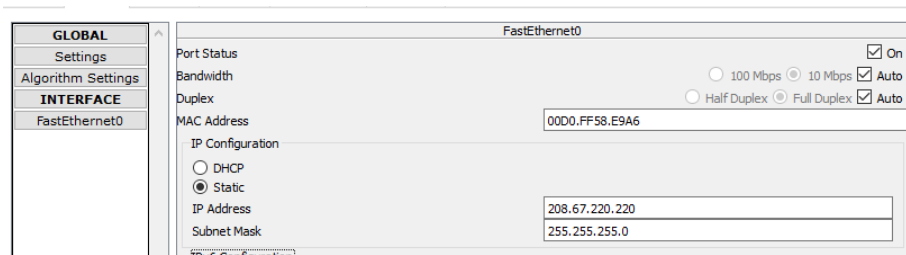
Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Address
DHCPPool	208.67.220.220	208.67.220.220	208.67.220.1	255.255.255.0	50	0.0.0.0	0.0.0.0
serverPool	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0	512	0.0.0.0	0.0.0.0

- Lalu klik DNS pada baris kiri
- Sesuaikan isian seperti gambar dibawah lalu klik add

No.	Name	Type	Detail
0	najmi.com	A Record	208.67.220.220

- Lalu pilih Config tab di baris atas lalu klik setting di baris kiri
- Sesuaikan isian seperti gambar di bawah ini

- Lalu pilih FastEthernet0 pada baris kiri
- Sesuaikan isian seperti gambar di bawah ini (MAC Address tidak perlu diganti)



### 3. Tes Koneksi Antara Client dengan Server

#### a. Memastikan komputer mendapat konfigurasi IPv4 dari DHCP

- Klik dua kali pada komputer, pilih tab Desktop lalu Command Prompt
- Ketik ipconfig /release lalu tekan enter
- Ketik ipconfig /renew lalu tekan enter, jika sesuai maka berhasil

```
C:\>ipconfig /release

IP Address. . . . . : 0.0.0.0
Subnet Mask. . . . . : 0.0.0.0
Default Gateway. . . . . : 0.0.0.0
DNS Server. . . . . : 0.0.0.0

C:\>ipconfig /renew

IP Address. . . . . : 192.168.0.102
Subnet Mask. . . . . : 255.255.255.0
Default Gateway. . . . . : 192.168.0.1
DNS Server. . . . . : 208.67.220.220
```

#### b. Mencoba menghubungi server najmi.com dengan ping

- Klik dua kali pada komputer, pilih tab Desktop lalu Command Prompt
- Ketik ping najmi.com lalu tekan enter

```
C:\>ping najmi.com

Pinging 208.67.220.220 with 32 bytes of data:

Reply from 208.67.220.220: bytes=32 time=2ms TTL=127
Reply from 208.67.220.220: bytes=32 time=1ms TTL=127
Reply from 208.67.220.220: bytes=32 time=1ms TTL=127
Reply from 208.67.220.220: bytes=32 time=1ms TTL=127

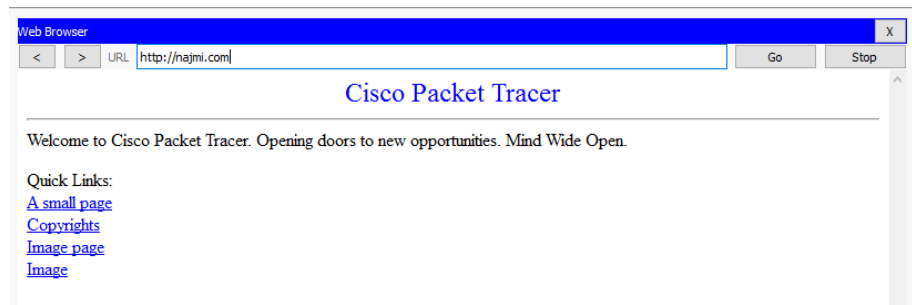
Ping statistics for 208.67.220.220:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms
```

- Jika muncul reply dari server seperti di atas, maka berhasil

#### c. Mencoba menghubungi server dengan HTTP

- Klik dua kali pada komputer, pilih tab Desktop lalu Web Browser

- Ketik najmi.com lalu tekan enter



- Jika yang muncul seperti gambar di atas, maka berhasil
- d. Mencoba menghubungi server melalui laptop dengan ping
- Klik dua kali pada laptop, pilih tab Desktop lalu Command Prompt
  - Ketik ipconfig /release lalu tekan enter
  - Ketik ipconfig /renew lalu tekan enter
  - Ketik ping najmi.com lalu tekan enter

```
C:\>ipconfig /release

IP Address.....: 0.0.0.0
Subnet Mask.....: 0.0.0.0
Default Gateway.....: 0.0.0.0
DNS Server.....: 0.0.0.0

C:\>ipconfig /renew

IP Address.....: 192.168.0.100
Subnet Mask.....: 255.255.255.0
Default Gateway.....: 192.168.0.1
DNS Server.....: 208.67.220.220

C:\>ping najmi.com

Pinging 208.67.220.220 with 32 bytes of data:

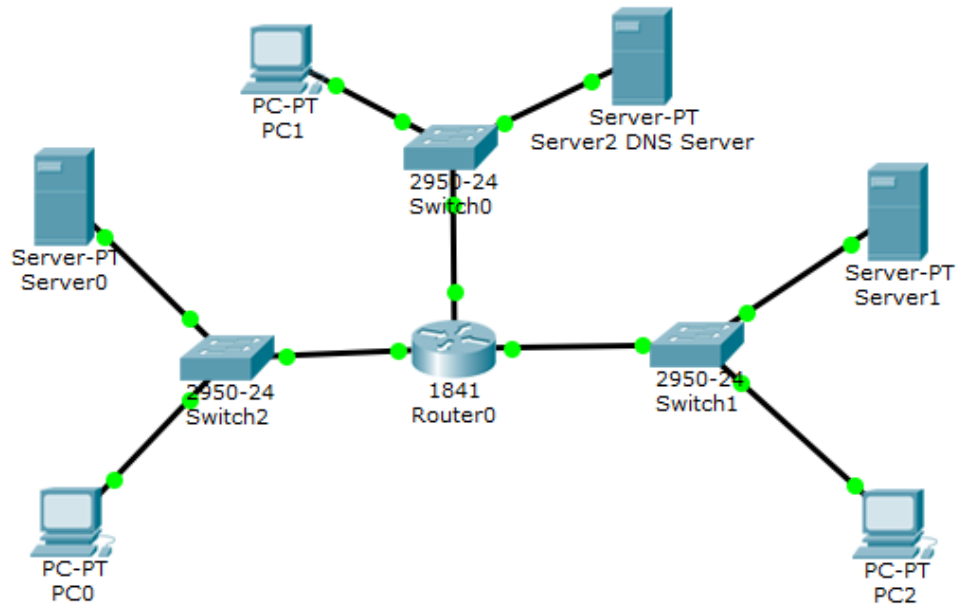
Reply from 208.67.220.220: bytes=32 time=10ms TTL=127
Reply from 208.67.220.220: bytes=32 time=10ms TTL=127
Reply from 208.67.220.220: bytes=32 time=14ms TTL=127
Reply from 208.67.220.220: bytes=32 time=10ms TTL=127

Ping statistics for 208.67.220.220:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 10ms, Maximum = 14ms, Average = 11ms
```

- Apabila setelah /renew sesuai dengan yang telah ditentukan dan juga ketika ping ke najmi.com telah mendapatkan reply seperti gambar di atas, maka koneksi telah berhasil.

## F. Tugas

1. Coba buat interkoneksi antara 3 buah network yang terhubung pada sebuah router. Di network-1 terdapat DNS Server dan 1 PC, di network-2 terdapat HTTP Server (domain bebas) dan 1 PC, di network-3 terdapat HTTP Server (domain bebas) dan 1 PC. Lakukan konfigurasi sedemikian sehingga setiap workstation bisa mengakses layanan server-server yang ada pada tiga network tersebut. Ilustrasi pada gambar dibawah ini!



## PERCOBAAN 12. ARDUINO

### A. Tujuan Percobaan

1. Mahasiswa dapat memahami konsep dasar mikrokontroler arduino.
2. Mahasiswa dapat memahami blok komponen arduino.
3. Mahasiswa dapat membuat program menggunakan arduino IDE.
4. Mahasiswa dapat membuat konsep dan aplikasi arduino.

### B. Bahan

- |                      |            |
|----------------------|------------|
| 1. Arduino Uno       | x1         |
| 2. <i>Buzzer</i>     | x1         |
| 3. Relay             | x1         |
| 4. Transistor 2N3904 | x1         |
| 5. Project Board     | x1         |
| 6. LED               | x2         |
| 7. Trimpot 50k       | x1         |
| 8. Push button       | x2         |
| 9. Kabel Jumper      | secukupnya |

### C. Alat

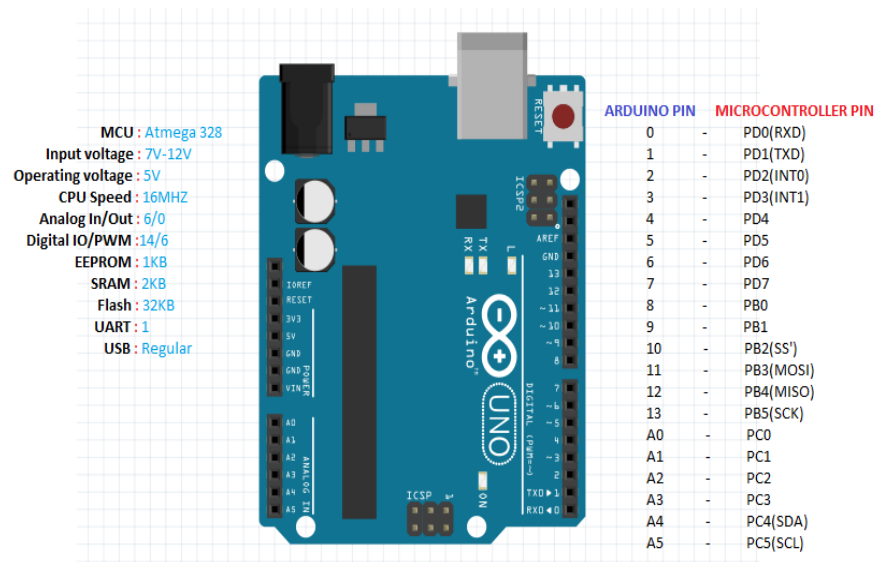
1. Tang pemotong (opsional)
2. Multimeter (opsional)
3. Obeng minus
4. Laptop dengan software Arduino IDE.

### D. Teori Dasar

1. Arduino

Arduino adalah salah satu kit mikrokontroler yang berbasis pada Atmega328. Modul ini sudah dilengkapi dengan berbagai hal yang dibutuhkan untuk mendukung mikrokontroler untuk bekerja, hanya sambungkan ke power suply atau sambungkan melalui kabel USB ke PC selanjutnya Arduino Uno ini sudah siap digunakan. Arduino Uno ini memiliki 14 pin digital input/output, 6 analog input, sebuah resonator keramik 16MHz, koneksi USB, konektor power input, ICSP header, dan sebuah tombol reset.





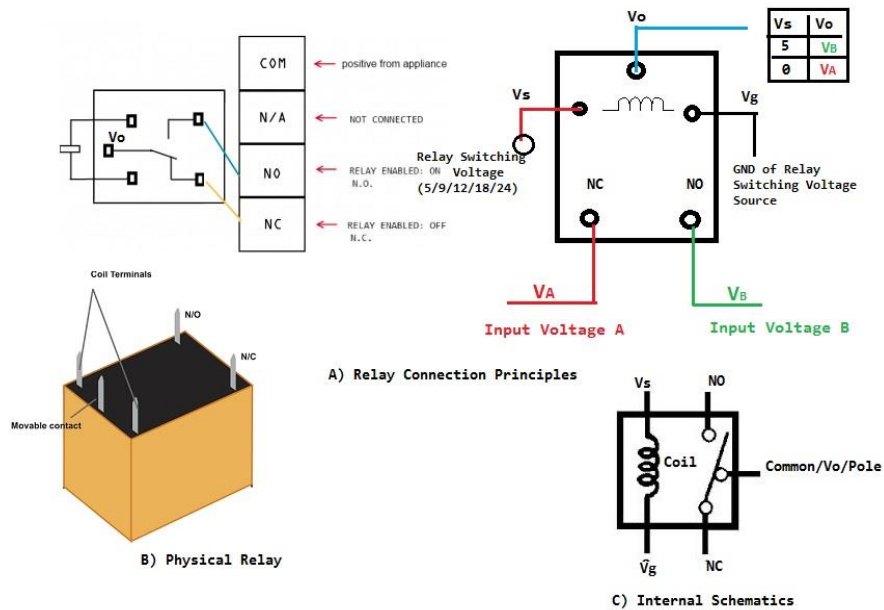
Gambar 1. Detail Arduino Uno tampak atas.

## 2. Relay

Relay terdiri dari *coil* dan *switch* (Pole, No, Nc), untuk mengaktifkan switch ini tergantung dari pencatuan coil, coil sendiri memiliki tegangan catu yang berbeda-beda, pada praktikum ini menggunakan relay dengan coil 5V DC, namun pada umumnya terdapat coil dengan tegangan catu 5V,12V,24V,48V. Ketika coil dicatu maka Pole akan tersambung pada No (normaly open) pada kasus ini normaly open berarti ketika coil tidak di catu maka titik pole dan No tidak terhubung. Berartu jetuja coil tidak di catu maka Pole akan tersambung pada Nc (normaly close).



Gambar 2. Tampilan fisik relay 5V.



Gambar 3. Gambaran prinsip kerja Relay.

### 3. Buzzer Melody

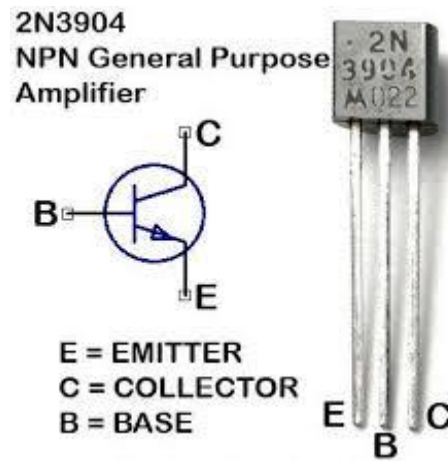
*Buzzer* adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Perangkat elektronika ini terbuat dari *elemen piezoceramics* yang diletakkan pada suatu diafragma yang mengubah getaran/vibrasi menjadi gelombang suara. *Buzzer* menggunakan resonansi untuk memperkuat intensitas suara. Berbeda dengan *active buzzer*, *buzzer melody* akan bersuara lantang bila diberi frekuensi tertentu. Tegangan catu *buzzer* juga berbeda-beda, pada praktek ini menggunakan *buzzer 5v*.



Gambar 4. Tampilan fisik *Buzzer*.

### 4. Transistor 2N3904

Transistor ini adalah jenis NPN berarti transistor akan bekerja apabila *base emitter* diberi bias forward dan basis *colector* diberi bias reverse, berarti bahwa ketika ada arus yang mengalir dari *base* ke *emitter* maka ada arus yang mengalir dari *colector* ke *emitter*.



Gambar 5. Konfigurasi kaki transistor 2N3904.

## 5. LED

LED adalah diode yang mengemisikan cahaya ketika diberi bias forward, artinya *anode* diberi tegangan yang lebih tinggi dibanding tegangan pada *cattode*.



Gambar 6. Tampilan Fisik LED.

## 6. Trimpot

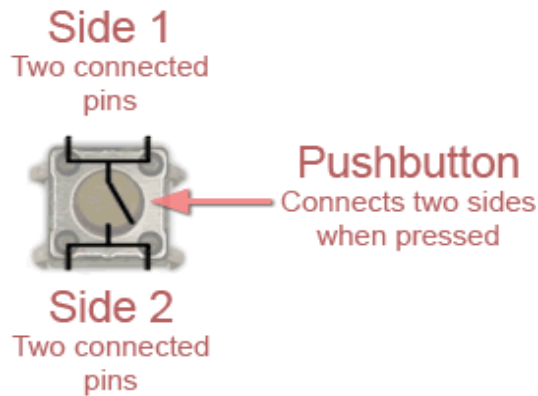
Trimpot adalah golongan *variable* resistor, dimana nilai resistansinya akan berubah-ubah ketika kita memutarkannya. Prinsipnya adalah pembagi tegangan, trimpot terdiri dari 2 buah resistor yang diseri.



Gambar 7. Tampilan fisik Trimpot.

## 7. Push Button

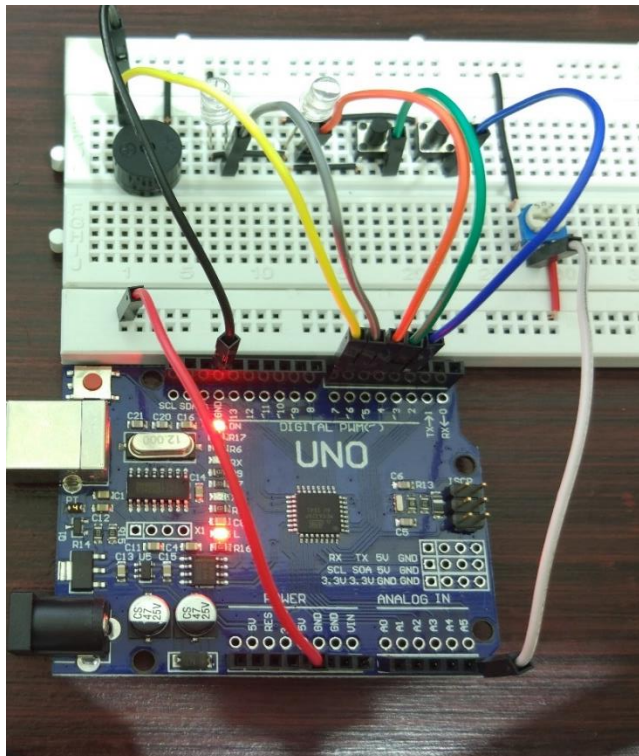
Push button memiliki jenis push off dan push on. Pada percobaan ini kita menggunakan push ON artinya ketika ditekan maka kedua sisi akan tersambung. Sebaliknya jika push off, ketika ditekan maka kedua sisi akan terputus.



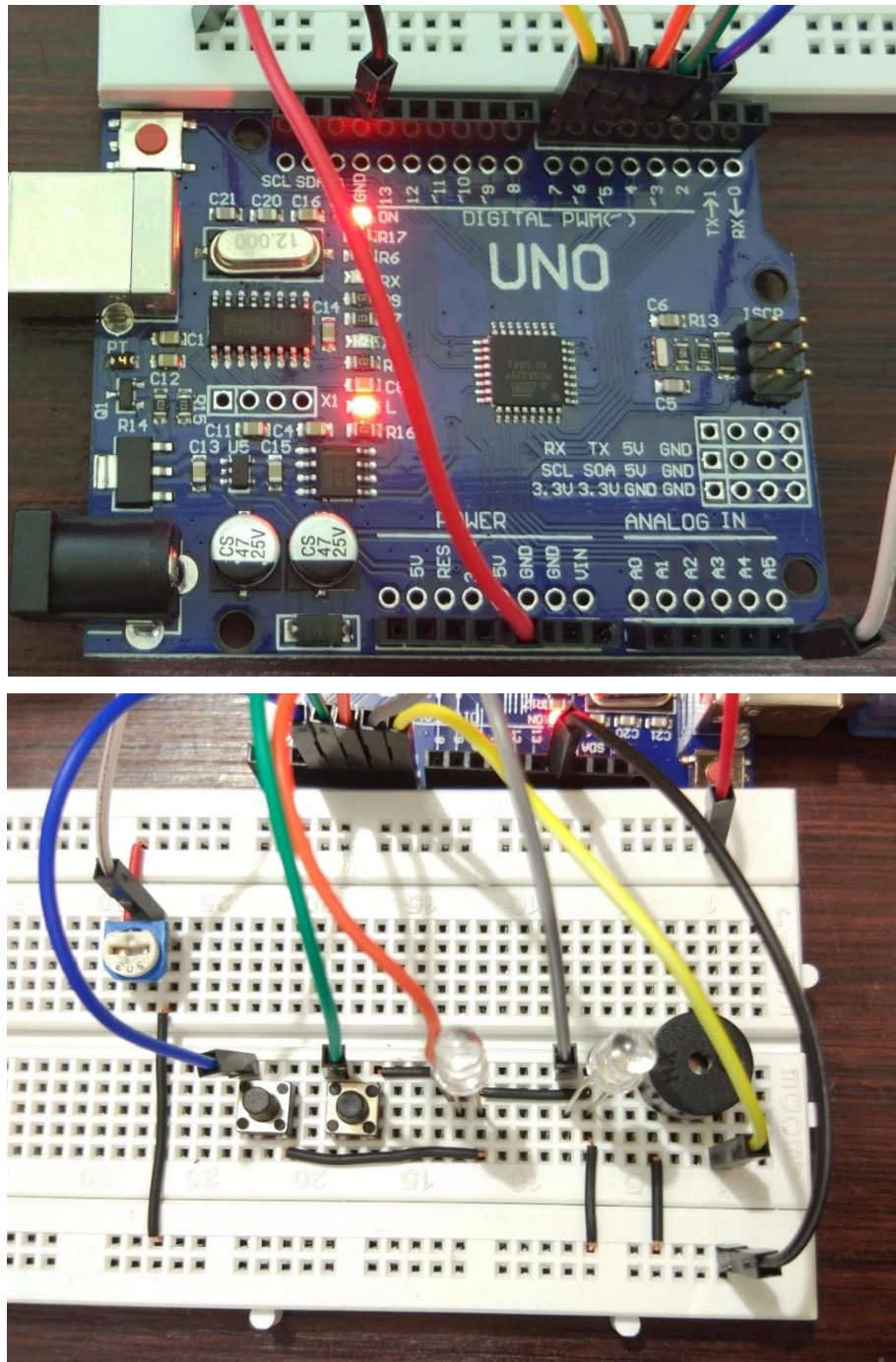
Gambar 8. Gambaran prinsip kerja pushbutton.

### E. Langkah Percobaan

1. Siapkan alat dan bahan, kemudian susun rangkaian dibawah pada project board, perhatikan warna kabel dan sambungannya seperti gambar 9 dan 10. Sambungkan usb terakhir setelah rangkaian tersusun.



Gambar 9. Sambungan rangkaian tampak atas



Gambar 10. Sambungan rangkaian LED, *buzzer*, *pushbutton*.

Keterangan:

- Kabel Hitam adalah kabel ground terhubung ke ground arduino. Setiap kabel hitam terhubung dengan negative komponen, seperti katoda LED, negative buzzer, salah satu bagian kaki push button.
- Kabel merah adalah kabel positif 5V terhubung ke pin 5V arduino.
- Kabel biru tersambung antara pin nomor 3 arduino ke salah satu kaki push button 1, sisi lain dari kaki yang terhubung ke negatif.

- Kabel hijau tersambung antara pin nomor 4 arduino ke salah satu kaki push buton 2 sisi lain dari kaki yang terhubung ke negatif.
- Kabel orange tersambung antara pin nomor 5 arduino ke anoda LED1.
- Kabel abu tersambung antara pin nomor 6 arduino ke anoda LED2.
- Kabel kuning tersambung antara pin nomor 7 arduino ke positif buzzer.
- Kabel putih tersambung antara kaki analog 5 (A5) arduino ke kaki tengah dari trimpot.

2. Apabila sudah tersusun, sambungkan kabel usb dari pc ke arduino. Buka Arduino IDE pada laptop, kemudian ikuti script dibawah.

**a. Menyalakan LED sintaks decision.**

- **Part 1 (menggunakan sintaks IF else)**

```

////////////////////////////////////
// Menyalakan LED menggunakan push button //
// dengan IF //
// part 1 //
////////////////////////////////////

#define PB1 3
#define PB2 4
#define LED1 5
#define LED2 6

int pushbutton1,pushbutton2;

void setup()
{
  pinMode(PB1, INPUT);
  pinMode(PB2, INPUT);
  digitalWrite(PB1, HIGH);
  digitalWrite(PB2, HIGH);
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
}

void loop()
{
  pushbutton1=digitalRead(PB1);
  pushbutton2=digitalRead(PB2);

  if(pushbutton1==0)
  {
    digitalWrite(LED1,HIGH);
  }
  else
  {
    digitalWrite(LED1,LOW);
  }
}

```

```

        if(pushbutton2==0)
        {
            digitalWrite(LED2,HIGH);
        }
        else
        {
            digitalWrite(LED2,LOW);
        }
    }
}

```

- **Part 2 (menggunakan sintaks FOR)**

```

////////////////////////////////////
// Menyalakan LED berulang kali ketika //
// tombol ditekan menggunakan for //
// part 2 //
////////////////////////////////////

#define PB1 3
#define PB2 4
#define LED1 5
#define LED2 6

int pushbutton1,pushbutton2;

void setup()
{
    pinMode(PB1, INPUT);
    pinMode(PB2, INPUT);
    digitalWrite(PB1, HIGH);
    digitalWrite(PB2, HIGH);
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
}

void loop()
{
    pushbutton1=digitalRead(PB1);
    pushbutton2=digitalRead(PB2);

    if(pushbutton1==0)
    {
        for(int i=0; i<5; i++)
        {
            digitalWrite(LED1,HIGH);
            delay(500);
            digitalWrite(LED1,LOW);
            delay(500);
        }
    }

    if(pushbutton2==0)
    {
        for(int i=0; i<5; i++)
        {
            digitalWrite(LED2,HIGH);
            delay(500);
            digitalWrite(LED2,LOW);
            delay(500);
        }
    }
}

```

```
    }  
  }  
}
```

- **Part 3 (menggunakan sintaks while)**

```
////////////////////////////////////  
// Menyalakan LED berkedip kemudian //  
// menyala berulang menggunakan while //  
// part 3 //  
////////////////////////////////////  
  
#define PB1 3  
#define PB2 4  
#define LED1 5  
#define LED2 6  
  
int pushbutton1,pushbutton2;  
char count;  
  
void setup()  
{  
  pinMode(PB1, INPUT);  
  pinMode(PB2, INPUT);  
  digitalWrite(PB1, HIGH);  
  digitalWrite(PB2, HIGH);  
  pinMode(LED1, OUTPUT);  
  pinMode(LED2, OUTPUT);  
}  
  
void loop()  
{  
  
  pushbutton1=digitalRead(PB1);  
  pushbutton2=digitalRead(PB2);  
  
  while(count<10)  
  {  
    count=count+1;  
    delay(1000);  
    digitalWrite(LED1,HIGH);  
    delay(200);  
    digitalWrite(LED1,LOW);  
    digitalWrite(LED2,HIGH);  
    delay(200);  
    digitalWrite(LED2,LOW);  
  }  
  digitalWrite(LED1,HIGH);  
  digitalWrite(LED2,HIGH);  
  delay(500);  
  digitalWrite(LED1,LOW);  
  digitalWrite(LED2,LOW);  
  delay(500);  
}
```

- **Part 4 (menggunakan sintaks switch case)**

```
////////////////////////////////////  
// Menyalakan LED bergantian dengan tombol //  
// menggunakan switch case //  
// part 4 //  
////////////////////////////////////
```



```

#define PB1 3
#define PB2 4
#define LED1 5
#define LED2 6

int pushbutton1,pushbutton2;
char count;

void setup()
{
  pinMode(PB1, INPUT);
  pinMode(PB2, INPUT);
  digitalWrite(PB1, HIGH);
  digitalWrite(PB2, HIGH);
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);

}

void loop()
{

  pushbutton1=digitalRead(PB1);
  pushbutton2=digitalRead(PB2);

  switch(pushbutton1)
  {
  case 1:
    digitalWrite(LED1,HIGH);
    digitalWrite(LED2,HIGH);
    break;
  case 0:
    for(int i=0;i<10;i++)
    {
      digitalWrite(LED1,HIGH);
      digitalWrite(LED2,HIGH);
      delay(100);
      digitalWrite(LED1,LOW);
      digitalWrite(LED2,LOW);
      delay(100);
    }
    break;
}

switch(pushbutton2)
{
  case 1:
    digitalWrite(LED1,HIGH);
    digitalWrite(LED2,HIGH);
    break;
  case 0:
    for(int i=0;i<10;i++)
    {
      digitalWrite(LED1,HIGH);
      digitalWrite(LED2,LOW);
      delay(100);
      digitalWrite(LED1,LOW);
      digitalWrite(LED2,HIGH);
      delay(100);
    }
    break;
}
}

```

## b. Analog input dan PWM

- **Part 1 (serial monitor analog input menggunakan trimpot)**

```
////////////////////////////////////  
// Analog input dan PWM //  
// input Trimpot dan serial monitor //  
// part 1 //  
////////////////////////////////////  
  
#define PB1 3  
#define PB2 4  
#define LED1 5  
#define LED2 6  
#define trimpot A5  
int variabelADC;  
  
void setup()  
{  
  Serial.begin(9600);  
  pinMode(trimpot, INPUT);  
}  
  
void loop() {  
  variabelADC=analogRead(trimpot);  
  Serial.print("Nilai Analog 10bit dalam desimal : ");  
  Serial.println(variabelADC);  
  delay(100);  
}
```

Buka serial monitor dengan klik Tools > Serial Monitor

Putar-putar trimpot

- **Part 2 (mengatur intensitas LED menggunakan trimpot)**

```
////////////////////////////////////  
// Analog input dan PWM atur LED dengan //  
// input Trimpot dan serial monitor //  
// part 2 //  
////////////////////////////////////  
  
#define PB1 3  
#define PB2 4  
#define LED1 5  
#define LED2 6  
#define trimpot A5  
int variabelADC, ADCbaru;  
float vo;  
  
void setup()  
{  
  Serial.begin(9600);  
  pinMode(trimpot, INPUT);  
}  
  
void loop()  
{  
  variabelADC=analogRead(trimpot);  
  vo=(5*variabelADC)/1023;  
  ADCbaru=map(variabelADC,0,1023,0,255);
```

```

    analogWrite(LED1,ADCbaru);
    analogWrite(LED2,ADCbaru);
    Serial.print("Nilai ADC baru : ");
    Serial.print(ADCbaru);
    Serial.print(" Nilai tegangan : ");
    Serial.print(vo);
    Serial.println(" V ");

    delay(100);
}

```

Buka serial monitor dengan klik Tools > Serial Monitor

Putar-putar trimpot

- **Part 3 (mengatur kecepatan perpindahan LED menggunakan trimpot)**

```

////////////////////////////////////
// Analog input dan PWM atur kecepatan LED//
// input Trimpot dan serialmonitor //
// part3 //
////////////////////////////////////

#define PB1 3
#define PB2 4
#define LED1 5
#define LED2 6
#define trimpot A5
int variabelADC, ADCbaru;

void setup()
{
    Serial.begin(9600);
    pinMode(trimpot,INPUT);
    pinMode(LED1,OUTPUT);
    pinMode(LED2,OUTPUT);
}

void loop()
{
    variabelADC=analogRead(trimpot);
    ADCbaru=map(variabelADC,0,1023,0,2000);
    digitalWrite(LED1,HIGH);
    digitalWrite(LED2,LOW);
    delay(ADCbaru);
    digitalWrite(LED1,LOW);
    digitalWrite(LED2,HIGH);
    delay(ADCbaru);
    digitalWrite(LED1,HIGH);
    digitalWrite(LED2,HIGH);
    delay(ADCbaru);
    digitalWrite(LED1,LOW);
    digitalWrite(LED2,LOW);
    Serial.print("Nilai delay: ");
    Serial.print(ADCbaru);
    Serial.println(" ms");
    delay(100);
}

```

Buka serial monitor dengan klik Tools > Serial Monitor

Putar-putar trimpot

### c. Buzzer

- **Part 1 (mengatur frekuensi buzzer)**

```

////////////////////////////////////
// Buzzer atur frekuensi //
// sweep frekuensi //
// part1 //
////////////////////////////////////

#define PB1 3
#define PB2 4
#define LED1 5
#define LED2 6
#define buzzer 7
int freq=0;

void setup() {
  pinMode(buzzer,OUTPUT);
}

void loop()
{
  awal:
  tone(buzzer,freq,100);
  freq=freq+100;
  delay(200);
  if(freq>=3500)
  {
    freq=3500;
    goto akhir;
  }
  goto awal;

  akhir:
  tone(buzzer,freq,100);
  freq=freq-100;
  delay(200);
  if(freq<=0)
  {
    goto awal;
  }
  goto akhir;
}

```

- **Part 2 (membuat melodi lagu menggunakan buzzer)**

```

////////////////////////////////////
// Buzzer atur frekuensi memnbuat lagu //
// nada pitch.h manual //
// part 2 //
////////////////////////////////////

#define C6 1047
#define D6 1175
#define E6 1319
#define F6 1397

```

```

#define G6 1568
#define A6 1760
#define C7 2093
#define B6 1976

#define buzzer 7

void setup()
{
  pinMode(buzzer, OUTPUT);
}

void loop()
{
  tone(buzzer, C6, 500);
  delay(1000);
  tone(buzzer, D6, 200);
  delay(1000);
  tone(buzzer, E6, 400);
  delay(1000);
  tone(buzzer, F6, 400);
  delay(1000);
  tone(buzzer, G6, 600);
  delay(1000);
  tone(buzzer, E6, 200);
  delay(1000);
  tone(buzzer, C6, 500);
  delay(1000);
  tone(buzzer, A6, 500);
  delay(1000);
  tone(buzzer, C7, 200);
  delay(1000);
  tone(buzzer, B6, 200);
  delay(1000);
  tone(buzzer, A6, 500);
  delay(1000);
  tone(buzzer, G6, 500);
  delay(1000);
  noTone(buzzer);
  delay(1000);
}

```

Ganti nada lihat di lampiran untuk nada-nada lainnya

- **Part 3 (merubah frekuensi buzzer menggunakan potensiometer)**

```

////////////////////////////////////
// Buzzer          Atur frekuensi dengan
//
// input Trimpot dan serial monitor      //
// part3                                     //
////////////////////////////////////

#define PB1 3

```

```

#define PB2 4
#define LED1 5
#define LED2 6
#define trimpot A5
#define buzzer 7
int variabelADC, freq;

void setup() {
  Serial.begin(9600);
  pinMode(trimpot, INPUT);
  pinMode(buzzer, OUTPUT);
}

void loop()
{
  variabelADC=analogRead(trimpot);
  freq=map(variabelADC, 0, 1023, 0, 4000);
  Serial.print("frekuensi : ");
  Serial.print(freq);
  Serial.println(" Hz");
  tone(buzzer, freq, 200);
  delay(1000);
}

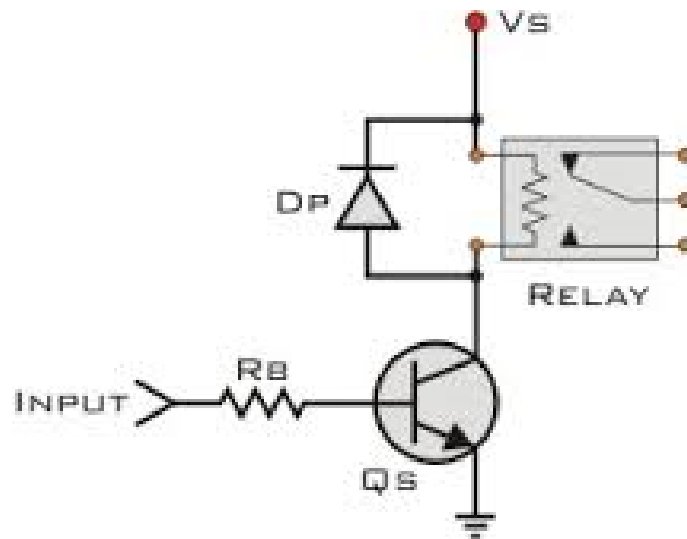
```

Buka serial monitor dengan klik Tools > Serial Monitor  
Putar-putar trimpot

## F. Tantangan

Dari praktikum diatas, pikirkan sebuah konsep program berikut aplikasinya menggunakan bahan-bahan yang telah dicoba dan sebagai tambahannya gunakan relay 5V untuk membuat proyek di pertemuan selanjutnya.





## G. Penutup

Demikian Jobsheet ini dibuat dengan sebaik yang penulis bisa, mohon maaf apabila terdapat kesalahan. Penulis menerima kritik dan sarannya mengenai jobsheet ini, terimakasih. Saran penulis, masih banyak ilmu yang dapat dikembangkan dari jobsheet ini, selamat mencoba proyek lainnya.