

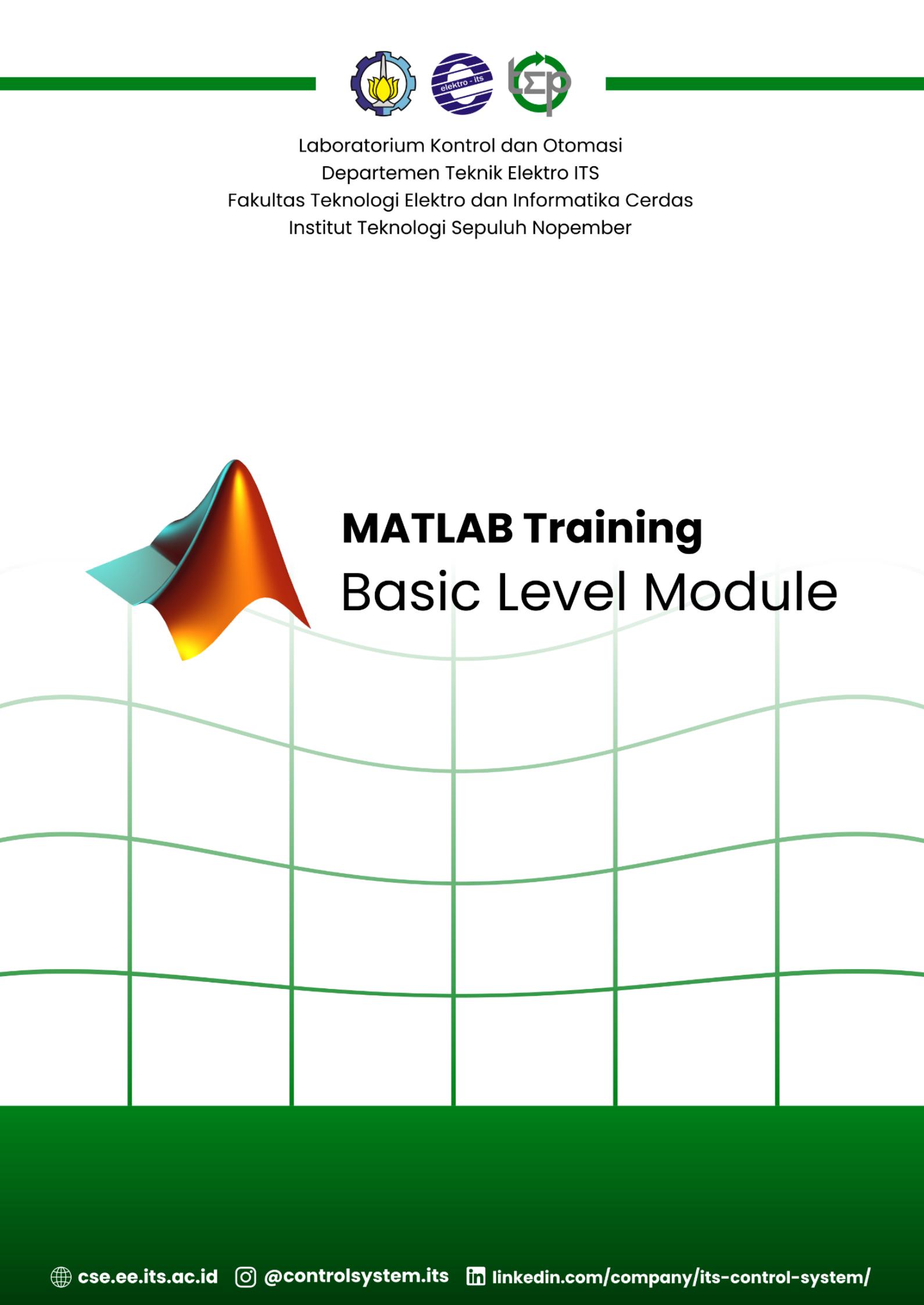


Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember



# **MATLAB Training**

## **Basic Level Module**



## Daftar Isi

<b>Daftar Isi</b>	<b>1</b>
<b>BAB 1. Pendahuluan Matlab</b>	<b>4</b>
1.1. Instalasi MATLAB dan Aktivasi Lisensi ITS	4
1.2. Memulai Penulisan Script	6
1.3. Penggunaan Live Editor	7
1.4. Run and Stop Program	7
1.5. Instalasi Toolbox	8
1.6. If else, While Loop, dan For Loop	9
1.6.1. The if Statement	9
1.6.2. For Loops	12
1.6.3. While Loops	12
1.6.4. While Loops dan For Loops	13
1.7. Plot Grafik	15
1.7.1. Menambahkan Judul dan Label Sumbu	15
1.7.2. Plot Banyak Grafik	16
1.7.3. Menentukan Gaya Garis dan Warna pada Grafik	18
1.8. Import	23
<b>BAB 2. Aljabar Linier</b>	<b>25</b>
2.1. Pendahuluan	25
2.2. Operasi Matriks	25
2.2.1. Membentuk Vektor dengan Matriks	25
2.2.1.1. Menuliskan secara langsung	25
2.2.1.2. Menggunakan colon operator	27
2.2.1.3. Menggunakan linspace	27
2.2.1.4. Membuat vektor dengan vektor	28
2.2.2. Membentuk Matriks Khusus dengan Syntax	29
2.2.2.1. Matriks Zeros	29
2.2.2.2. Matriks Ones	29
2.2.2.3. Matriks Eye	29
2.2.2.4. Matriks Diagonal	30
2.2.2.5. Matriks Magic	31
2.2.2.6. Matriks Random	32
2.2.3. Menentukan Ukuran Matriks	32
2.2.4. Modifikasi Elemen-Elemen Matriks	34
2.2.4.1. Ekstraksi Matriks	34
2.2.4.2. Membalik Elemen pada Matriks	36
2.2.4.3. Membalik Elemen pada Matriks	37



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

2.2.4.4. Menghapus Elemen pada Matriks	38
2.2.4.5. Menambahkan Elemen pada Matriks	39
2.2.4.6. Mengganti Elemen pada Matriks	39
2.2.4.7. Mengubah Dimensi Matriks	40
2.2.5. Transpose Matriks	40
2.2.6. Inverse Matriks	41
2.2.7. Determinan	42
2.2.8. Rank	43
2.2.9. Operasi Aritmatika	44
2.2.9.1. Penjumlahan Matriks	44
2.2.9.2. Pengurangan Matriks	44
2.2.9.3. Perkalian Matriks	45
2.2.9.4. Pembagian Matriks	45
2.2.10. Operasi Vektor	46
2.2.10.2. Cross Product	46
2.2.11. Eigenvalue dan Eigenvector	47
2.3. Penyelesaian Sistem Persamaan Linier	47
2.3.1. Mendefinisikan Sistem Persamaan Linier	47
2.3.2. Menyelesaikan Sistem Persamaan Linier	48
2.3.2.1. Linsolve	48
2.3.2.2. Solve	49
<b>BAB 3. Persamaan Diferensial</b>	<b>50</b>
3.1. Solusi Persamaan	50
3.2. Solusi Numerik	52
3.2.1. Metode euler	52
3.2.2. Metode runge-kutta orde 4	53
3.3. ODE23 and ODE45 Solver Matlab	54
3.3.1. ODE23	55
3.3.2. ODE45	55
3.4. Tabel Perbandingan Solusi Numerik	56
<b>BAB 4. Transformasi</b>	<b>57</b>
4.1. Pendahuluan	57
4.2. Transformasi Fourier Kontinu	57
4.3. Transformasi Fourier Balik (Inverse)	62
4.4. Transformasi Laplace	63
4.5. Transformasi Laplace Balik (Inverse)	67
<b>BAB 5. Pemodelan Sistem Elektrik</b>	<b>70</b>
5.1. Pendahuluan	70
5.2. Sistem Elektrik	70

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

<b>BAB 6. Pengenalan SIMULINK</b>	<b>75</b>
6.1. Pendahuluan	75
6.2. Lingkungan Grafis	75
6.3. Melihat Sinyal	79
6.4. Algoritma Dasar	82
<b>LAMPIRAN</b>	<b>84</b>

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS

Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://www.instagram.com/@controlsystem.its) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)



## BAB 1. Pendahuluan Matlab

### 1.1. Instalasi MATLAB dan Aktivasi Lisensi ITS

Sebelum memulai instalasi Matlab, ada beberapa hal yang perlu diperhatikan selama proses instalasi Matlab, yaitu:

- Matlab dapat diakses gratis oleh Mahasiswa ITS dengan lisensi pelajar
- Matlab dapat diaktifasi menggunakan email ITS dengan domain @mhs.its.ac.id dan pastikan email ITS tersebut aktif dan bisa diakses sebelum melakukan instalasi Matlab
- Pada saat instalasi pastikan terhubung dengan internet

Dan berikut merupakan langkah-langkah dalam instalasi Matlab :

- a) Buat akun baru pada <https://www.mathworks.com/mwaccount/register>

MathWorks Account

Create MathWorks Account

Email Address: name.nrp@mhs.its.ac.id

To access your organization's MATLAB license, use your work or university email.

Location: Indonesia

Which best describes you?: Student

Are you at least 13 years or older?  Yes  No

- b) Buka email dan lakukan verifikasi

MathWorks

Thank you for registering with MathWorks!

Next, please verify this email address for your MathWorks Account.

[Verify your email](#)

Alternatively, copy and paste the following link into your browser:

<https://www.mathworks.com/mwaccount/register/verify>

MathWorks Customer Service Team

Note. jika tidak ada email verifikasi yang masuk ketika mengakses melalui gmail, coba akses email melalui <https://outlook.office365.com/mail/>

c) Isi dan lengkapi data diri dalam form profil

MathWorks Account

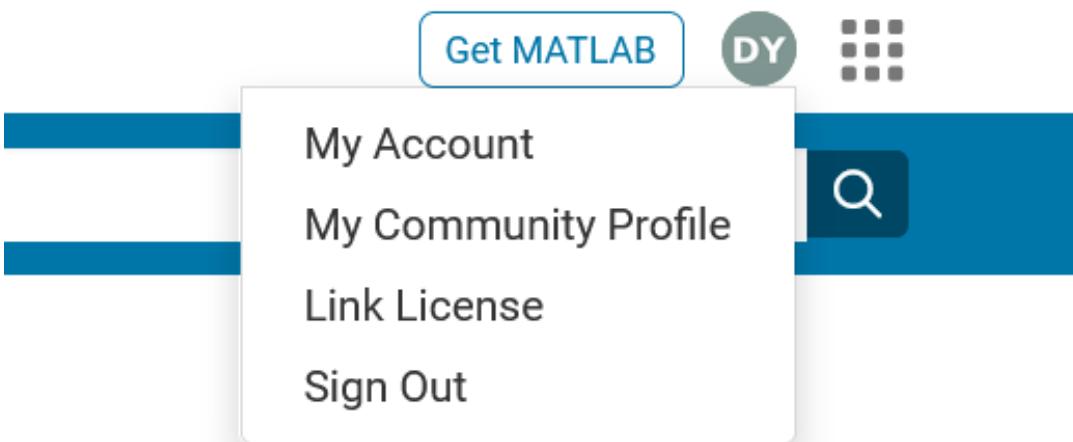
First Name	name	<input checked="" type="checkbox"/>
Last Name	last name	<input checked="" type="checkbox"/>
Password	*****	<input type="button" value="eye"/>
Password Confirmation	*****	<input type="button" value="eye"/>
What describes your role?	Student (Undergraduate)	<input type="button" value="dropdown"/>
Department	Electrical and Computer Engineering	<input type="button" value="dropdown"/>

**Location of your School/University**

Please provide the following information in English.

Location of School/University	Indonesia	<input type="button" value="dropdown"/>
School / University	Institut Teknologi Sepuluh Nopember	<input checked="" type="checkbox"/>

d) Pada pojok kanan atas klik bagian My Account



e) Akan muncul tampilan dashboard dan lisensi yang bisa digunakan

MathWorks Account

Search MathWorks.com

File ▾ | Security Settings ▾ | Quotes | Orders | Community Profile

**My Software**

License	Label	Option	Use	
40755015	MATLAB (Individual)	Total Headcount	Academic	<input type="button" value="dropdown"/> <input type="button" value="X"/> <input type="button" value="Cart"/>

[+ Link an additional license](#)  
[+ Get a trial](#)



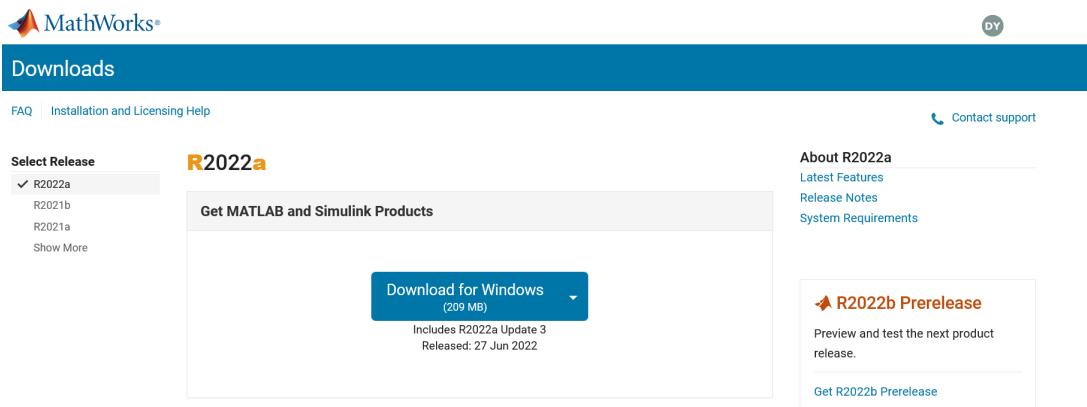
# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

- f) Untuk mengunduh software Matlab klik ikon unduh seperti berikut  
**My Software**

License	Label	Option	Use
40755015	MATLAB (Individual)	Total Headcount	Academic   

- g) Pilih versi matlab yang diinginkan (direkomendasikan versi terbaru) lalu unduh

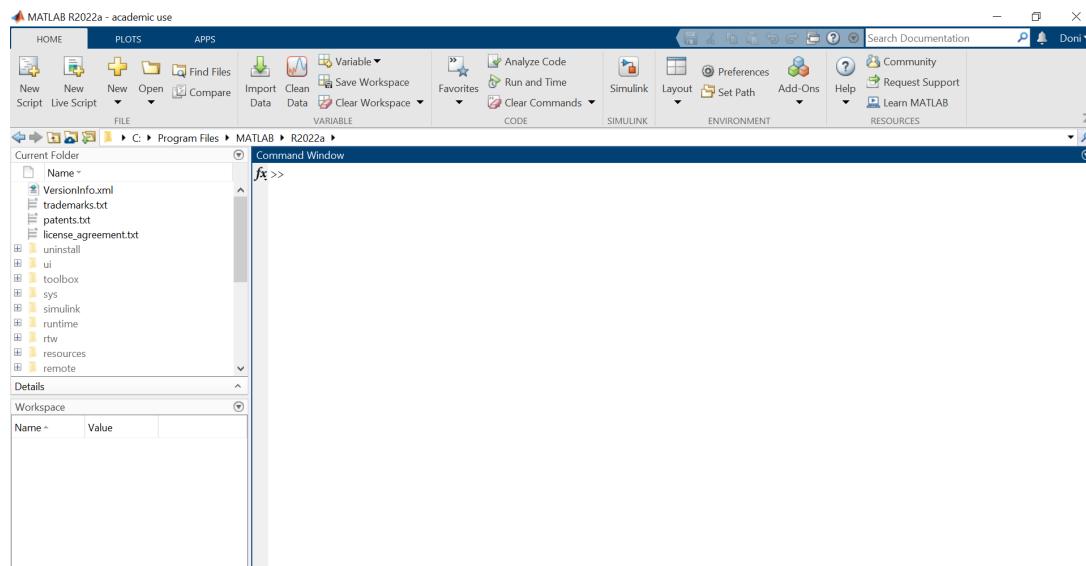


The screenshot shows the MathWorks Downloads page. On the left, a sidebar lists releases: R2022a (selected), R2021b, R2021a, and Show More. The main area displays the R2022a product page with a large 'Download for Windows' button highlighted with a green circle. Below it, text indicates it includes R2022a Update 3 and was released on June 27, 2022. To the right, there's information about the R2022b Prerelease and a link to get it.

- h) Setelah file selesai diunduh, ikuti langkah-langkah instalasi sampai Matlab terinstall.

## 1.2. Memulai Penulisan Script

Setelah melakukan instalasi Matlab dan Matlab dapat dibuka maka tampilan awal yang pertama kali terlihat dinamakan command window dan workspace



Di dalam command window tersebutlah penulisan perintah dapat dituliskan dan dijalankan. Sedangkan untuk workspace berisi penggunaan variabel atau data yang telah digunakan sebelumnya. Berikut beberapa daftar perintah atau script umum yang biasa digunakan pada Matlab

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Command	Fungsi
clc	Membersihkan command window
clear	Membersihkan data yang tersimpan pada workspace
ans	Menampilkan hasil terakhir yang tidak dideklarasikan
help	Menampilkan petunjuk pemakaian fungsi
commandhistory	Menampilkan riwayat perintah yang telah dituliskan
clf	Menghapus figur

## 1.3. Penggunaan Live Editor

Terdapat dua penggunaan editor pada Matlab yaitu editor dan live editor. Dari segi cara penggunaan sebenarnya dua editor ini bisa dibilang sama tetapi secara fitur memiliki perbedaan. Perbedaan yang paling mencolok adalah pada live editor baik hasil, gambar, grafik, dan lainnya dapat dilihat secara langsung pada bagian akhir syntax

```
K = 0.7;
zeta = 0.5;
omega_n = 2;
s = tf('s');
Gs = K*omega_n^2/(s^2+2*zeta*omega_n*s+omega_n^2)

Ts =
2.8
-----
s^2 + 2 s + 4
Continuous-time transfer function.

Ts = 0.1;
Gz_0 = c2d(Gs,Ts,"zoh")

Gz_0 =
0.01307 z + 0.01222
-----
z^2 - 1.783 z + 0.8187
Sample time: 0.1 seconds
Discrete-time transfer function.

Gz_1 = c2d(Gs,Ts,"foh")
```

## 1.4. Run and Stop Program

Untuk menjalankan dan menghentikan program pada Matlab dapat dilakukan dengan mengklik tombol run dan stop

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



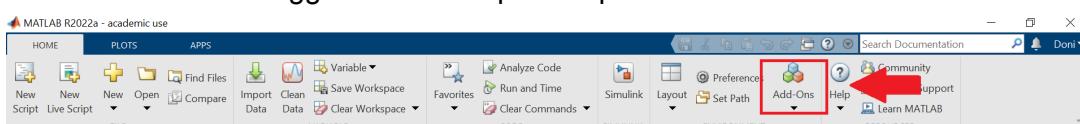
# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember



## 1.5. Instalasi Toolbox

Salah satu keunggulan Matlab yang membuat para pemakainya nyaman dalam memakai software ini terletak pada toolbox yang diberikan. Toolbox ini berisi berbagai macam fungsi Matlab yang telah dikembangkan oleh Mathworks (perusahaan yang bertanggung jawab dalam mengelola dan mengembangkan Matlab). Toolbox yang ada pada matlab ini dapat diinstall pada saat pertama kali Matlab diinstall atau jika ingin menambahkan toolbox ketika Matlab sudah terinstall dapat dilakukan dengan mengklik tombol add-Ons sehingga muncul tampilan seperti berikut



Dan tinggal dipilih toolbox mana yang ingin diinstall dan digunakan

The screenshot shows the MathWorks Add-On Explorer interface. On the left, there's a sidebar with filters for 'For You' (My Products: 108), 'Filter by Source' (MathWorks: 390, Community: 43,911), and 'Filter by Category'. Under 'Using MATLAB', it lists 'MATLAB' (11,785) and 'Simulink' (866). Under 'Workflows', it lists 'Parallel Computing' (196), 'Reporting and Database Access' (205), 'Systems Engineering' (21), 'Code Generation' (336), and 'Application Deployment' (124). The main area displays four toolbox cards: 'Bluetooth Toolbox' (Included in Your License), 'DSP HDL Toolbox' (Included in Your License), 'Industrial Communication Toolbox' (Included in Your License), and 'Wireless Testbench' (Included in Your License). A green 'Community Toolboxes' button is at the bottom.

Untuk ToolBox yang sering digunakan dalam Teknik Sistem Pengaturan adalah

1. Symbolic Math Toolbox
2. Simulink
3. Global Optimization Toolbox
4. Control System Toolbox
5. Deep Learning Toolbox
6. Fuzzy Logic Toolbox
7. Simscape
8. Simscape Electrical
9. Statistics and Machine Learning Toolbox
10. Optimization Toolbox
11. Signal Processing Toolbox
12. DSP System Toolbox

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



## 1.6. If else, While Loop, dan For Loop

Dalam bahasa pemrograman tentunya sudah tidak asing dengan istilah looping atau perulangan dan statement. Sama halnya seperti bahasa pemrograman yang lain, pada Matlab juga mengenal istilah tersebut dan berikut cara penggunaan masing-masingnya.

### 1.6.1. The if Statement

If statement digunakan untuk mengeksekusi **commands** berdasarkan **condition** yang diberikan terpenuhi atau tidak. Bentuk if statement:

```
if condition  
    commands  
end
```

*Condition* adalah boolean (true/false) dan *commands* adalah set MATLAB commands. Jika *condition* bernilai true, maka *commands* di dalam statement (sebelum *end*) akan dieksekusi; jika *condition* bernilai false, maka *commands* di dalam statement tidak akan dieksekusi.

#### Contoh 1 : Penggunaan dasar

Perhatikan if statement berikut:

```
x = 2;      % Input nilai x (ubah-ubah nilai ini)  
  
if x >= 0    % Set kondisi  
    val = x      % Mendefinisikan variabel val sama dengan x dan  
    men-display  
  
end
```

val = 2

Saat program dijalankan, jika x lebih besar atau sama dengan 0, maka variabel val akan sama dengan x; jika x negatif, maka tidak ada hasil apapun. Coba dengan mengganti-ganti nilai x

#### Contoh 2 : else command

Untuk menjalankan *commands* jika if statement dalam kondisi false, dapat menggunakan else. Script berikut akan menjalankan 2nd set-of-commands jika if dalam kondisi false.



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

```
if condition
    1st set-of-commands
else
    2nd set-of-commands
end
```

Contoh program berikut

```
x = -10;      % Input nilai x (ubah-ubah nilai ini)
if x >= 0
    y = x;
else
    y = -x;
end
y
```

y = 10

Jika  $x \geq 0$  maka  $y = x$ ; jika kondisi tersebut tidak dipenuhi maka  $y = -x$ .

### Contoh 3 : elseif command

elseif command digunakan untuk menjalankan beberapa command berdasarkan lebih dari satu kondisi.

```
if 1st condition
    1st set-of-commands
elseif 2nd condition
    2nd set-of-commands
...
elseif nth condition
    nth set-of-commands
else
    final set-of-commands
end
```

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://@controlsystem.its) [linkedin.com/company/its-control-system/](https://linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Ketika program dijalankan, saat if 1st condition bernilai true maka 1st set of commands akan dieksekusi; jika if 1st condition bernilai false dan hanya elseif 2nd condition yang bernilai true maka 2nd set of commands akan dieksekusi; begitupun selanjutnya.

Contohnya dapat menggunakan fungsi piecewise berikut

$$y = f(x) = \begin{cases} -x, & \text{if } x < 0 \\ \pi, & \text{if } x = 0 \\ x, & \text{if } 0 < x \end{cases}$$

```
x = 2;      % Input nilai x (ubah-ubah nilai ini)
if x < 0
    y = -x;
elseif x > 0
    y = x;
else
    y = pi;
end
y
```

y = 10

Saat nilai x positif atau negatif maka akan mengembalikan nilai y yaitu mutlak dari x, saat kedua kondisi tersebut tidak dipenuhi maka y akan mengembalikan nilai pi

## Contoh 4 : Operator logika

Kita bisa menggunakan operasi logika dengan **if statement**, contoh script berikut akan melakukan pengecekan apakah x bernilai positif atau negatif, jika memenuhi maka y akan sama dengan x, namun jika kondisi tersebut tidak terpenuhi maka y akan sama dengan pi

```
x = 0;
if (x < 0) || (x > 0) % Cek apakah x positif atau negatif
    y = x
else
    y = pi
end
```

y = 3.1416

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



Selain OR (||) kita juga menggunakan operasi logika lain seperti AND (&&), NOT (~)

## 1.6.2. For Loops

Script for loop yang bisa digunakan di MATLAB:

```
for index = index_vector  
    commands  
end
```

- command for menandakan dimulainya loop
- command end menandakan loop berakhir
- **commands** akan dieksekusi berulang sesuai dengan banyaknya loop yang didefinisikan
- index vektor dalam bentuk vektor merupakan definisi banyaknya loop
- index adalah variabel yang menyimpan banyaknya loop index vector

### Contoh: Menjumlahkan elemen-elemen vektor

Pada script di bawah akan dilakukan perhitungan penjumlahan semua elemen vektor [1 2 3 4 5 6]

```
jumlah = 0; % Mendefinisikan variabel nilai awal  
  
for i = 1:6 % i adalah vektor dari 1 sampai 6 dengan increment 1, i  
dapat diubah i = [1 2 3 4 5 6]  
  
jumlah = jumlah + i % perhitungan  
  
end % penanda looping berakhir
```

```
jumlah = 1  
  
jumlah = 3  
  
jumlah = 6  
  
jumlah = 10  
  
jumlah = 15  
  
jumlah = 21
```

## 1.6.3. While Loops

Script while loop yang bisa digunakan di MATLAB:

```
while condition
```



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

commands

end

Pada while loop, commands akan terus dijalankan saat condition bernilai true. Loop akan berakhir hanya jika condition bernilai false. Sehingga dalam while loop kita tidak perlu mendefinisikan banyaknya iterasi terlebih dahulu.

## Contoh : Penjumlahan

while loop berikut akan menghitung dari 0 sampai 4

```
n = 0; % Mendefinisikan variabel nilai awal
while n < 4 % Set kondisi
    n = n + 1 % Assign a new value to "n"
end % Exit the "while" loop
```

n = 1

n = 2

n = 3

n = 4

Penjumlahan n+1 akan terus berjalan saat n kurang dari 4, saat n lebih dari atau sama dengan 4 maka looping akan berakhir.

### 1.6.4. While Loops dan For Loops

Loops berikut akan menjumlahkan semua elemen vektor:

Menggunakan for loop:

```
x = 1:10; % Mendefinisikan array vektor dari 1 sampai 10
sum = 0; % Nilai awal
for i = 1:length(x) % i dari 1 sampai 10
    sum = sum + x(i) % menjumlahkan sum dengan x(i), x(i) mengakses
    nilai x pada data ke i
end
```

sum = 1

sum = 3

sum = 6

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

```
sum = 10
```

```
sum = 15
```

```
sum = 21
```

```
sum = 28
```

```
sum = 36
```

```
sum = 45
```

```
sum = 55
```

Menggunakan while loop:

```
a = 1:10; % Mendefinisikan array vektor dari 1 sampai 10
sum_a = 0; % Nilai awal
k = 1; % Deklarasi k untuk menghitung
while k <= length(a) % Set kondisi
    sum_a = sum_a + a(k) % menjumlahkan sum_a dengan a(k), a(k) mengakses
    nilai a pada data ke k
    k = k + 1; % Increment k
end
```

```
sum_a = 1
```

```
sum_a = 3
```

```
sum_a = 6
```

```
sum_a = 10
```

```
sum_a = 15
```

```
sum_a = 21
```

```
sum_a = 28
```

```
sum_a = 36
```

```
sum_a = 45
```

```
sum_a = 55
```

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

## 1.7. Plot Grafik

Plot Grafik merupakan salah satu fitur paling sering digunakan pada Matlab dikarenakan Matlab mempunyai kapabilitas yang kuat dalam segi pembuatan grafik dan cocok untuk digunakan para peneliti dan engineer. Ada beberapa perintah yang dapat digunakan untuk membuat plot grafik diantaranya

Kode	Fungsi
plot	Membuat plot
fplot	Membuat plot dengan fungsi $y = f(x)$
loglog	Membuat plot skala logaritma
semilogx	Membuat plot dengan sumbu x skala logaritma
semilogy	Membuat plot dengan sumbu y skala logaritma
polarplot	Membuat plot dengan koordinat polar

Selanjutnya akan diberikan beberapa contoh penggunaan plot grafik yang biasa digunakan pada Matlab.

### 1.7.1. Menambahkan Judul dan Label Sumbu

Penamaan judul dan pemberian label pada grafik juga merupakan hal yang tidak kalah penting dibanding lainnya, bahkan hal ini merupakan langkah awal yang seharusnya dilakukan dalam melakukan plot grafik

```
x = linspace(0,10,150);
y = cos(5*x);
plot(x,y)

%% Untuk memberikan judul pada grafik
%% title ('{nama judul}')
title('2-D Line Plot')

%% Untuk memberikan label pada masing masing sumbu
%% xlabel('{nama label sumbu x}') | ylabel('{nama label sumbu y}')
xlabel('x')
ylabel('cos(5x)')
```

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

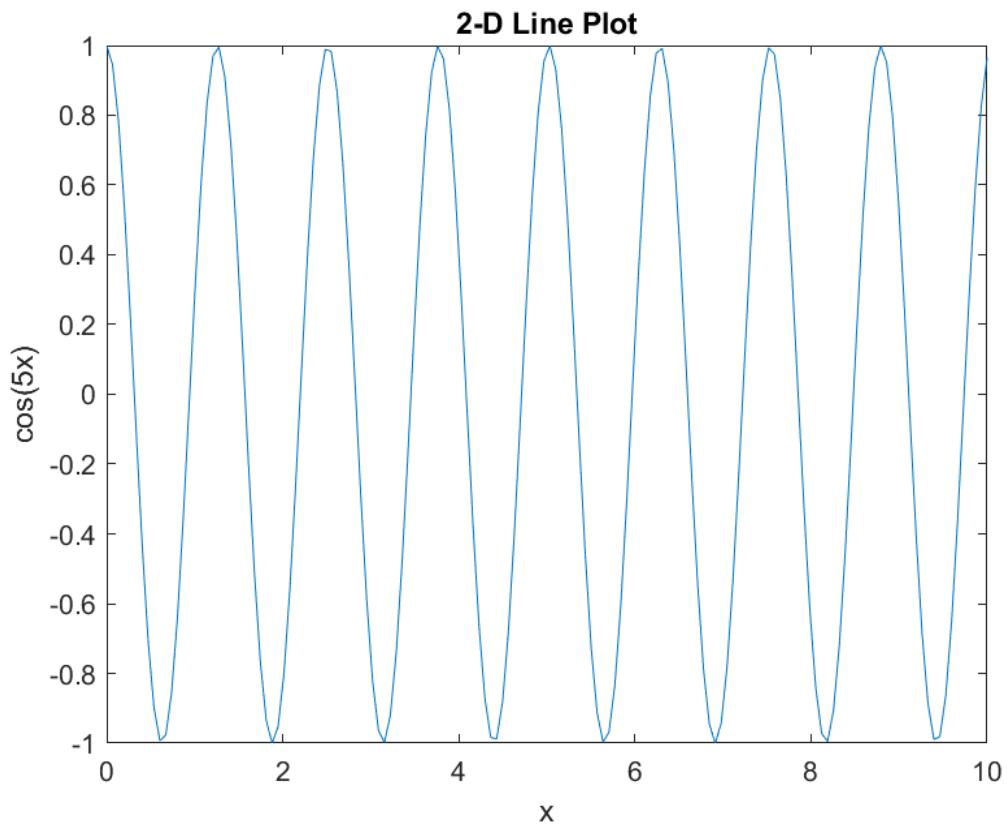


# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember



#### 1.7.2. Plot Banyak Grafik

```
x = linspace(-2*pi,2*pi);
y1 = sin(x);
y2 = cos(x);
figure
plot(x,y1,x,y2) %% Untuk plot beberapa grafik dalam 1 gambar
```

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS

Keputih, Sukolilo, Surabaya

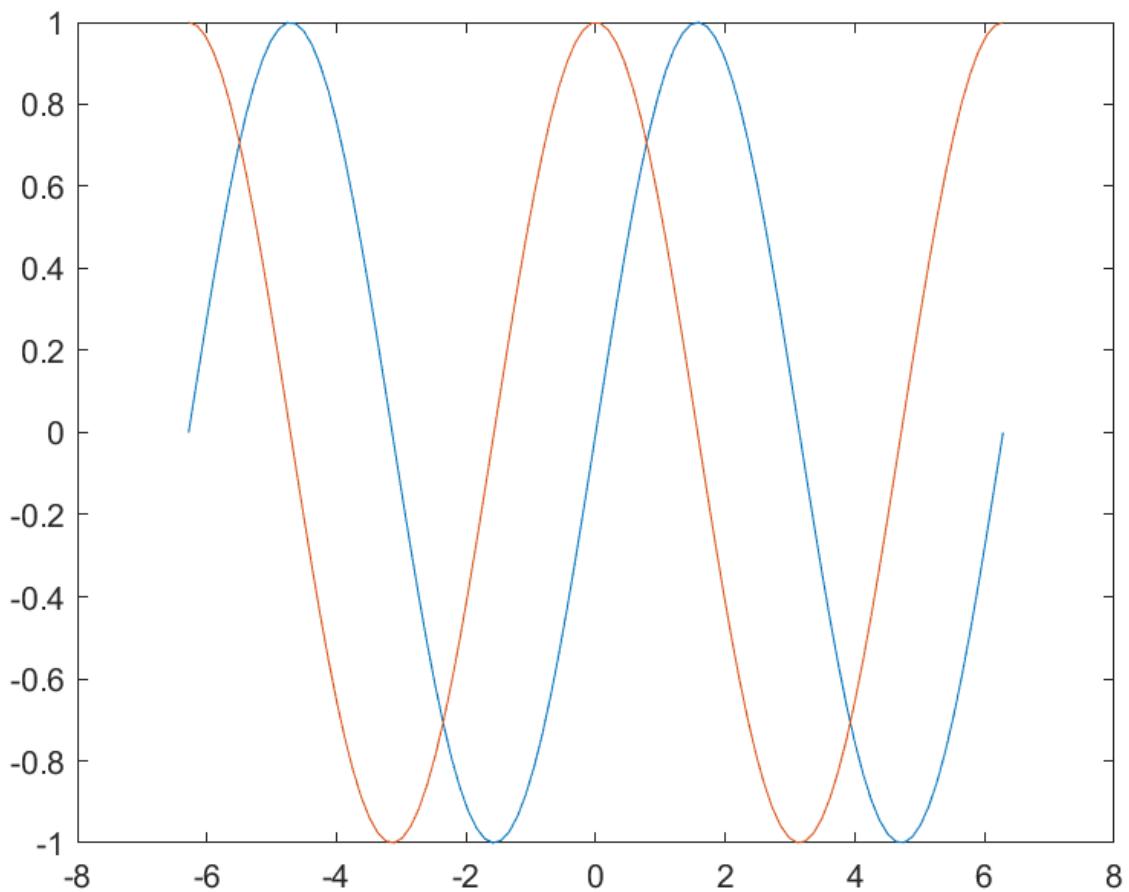


# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember



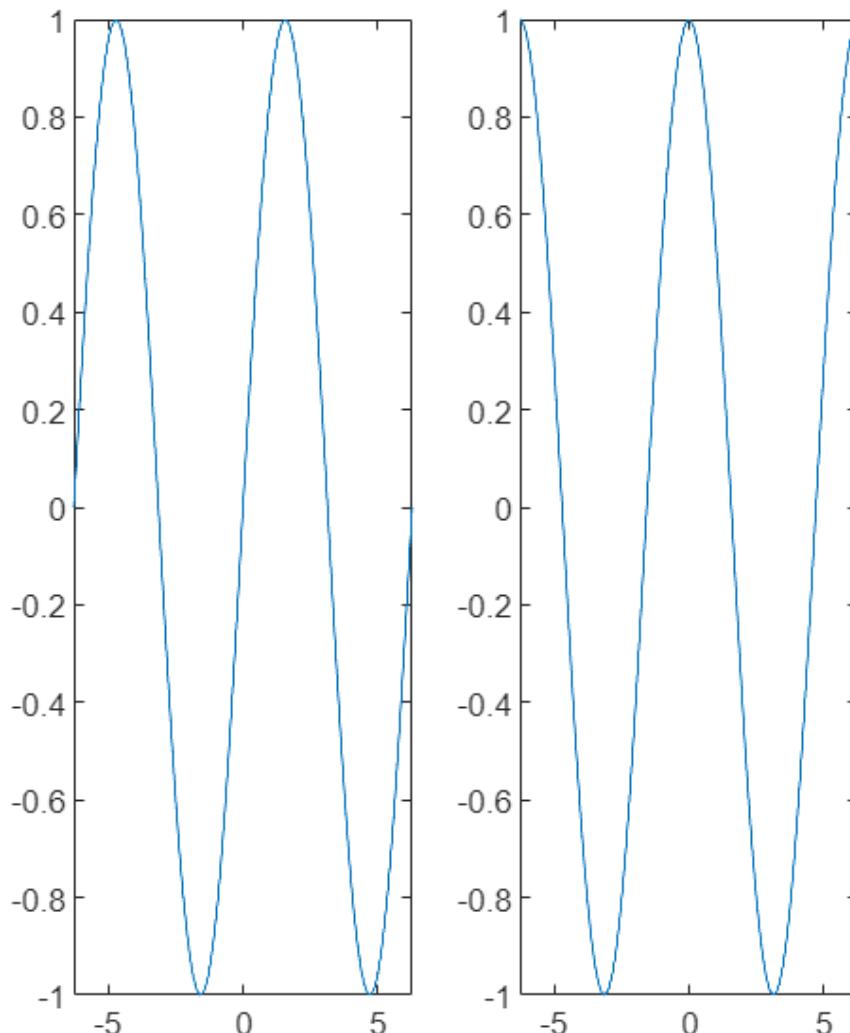
figure

```
% Untuk membagi beberapa grafik dalam beberapa gambar  
% subplot (jumlah baris yang diinginkan, jumlah kolom yang  
diinginkan, letak grafik)  
subplot(1,3,1)  
plot(x,y1)  
subplot(1,3,2)  
plot(x,y2)
```

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS

Keputih, Sukolilo, Surabaya



### 1.7.3. Menentukan Gaya Garis dan Warna pada Grafik

```
x = 0:pi/100:2*pi;
y1 = sin(x);
y2 = sin(x-0.25);
y3 = sin(x-0.5);

figure
% Untuk membuat gaya berbeda pada garis
% plot (x,y, '{kode gaya garis}')
plot(x,y1,x,y2, '--', x,y3, ':')
```

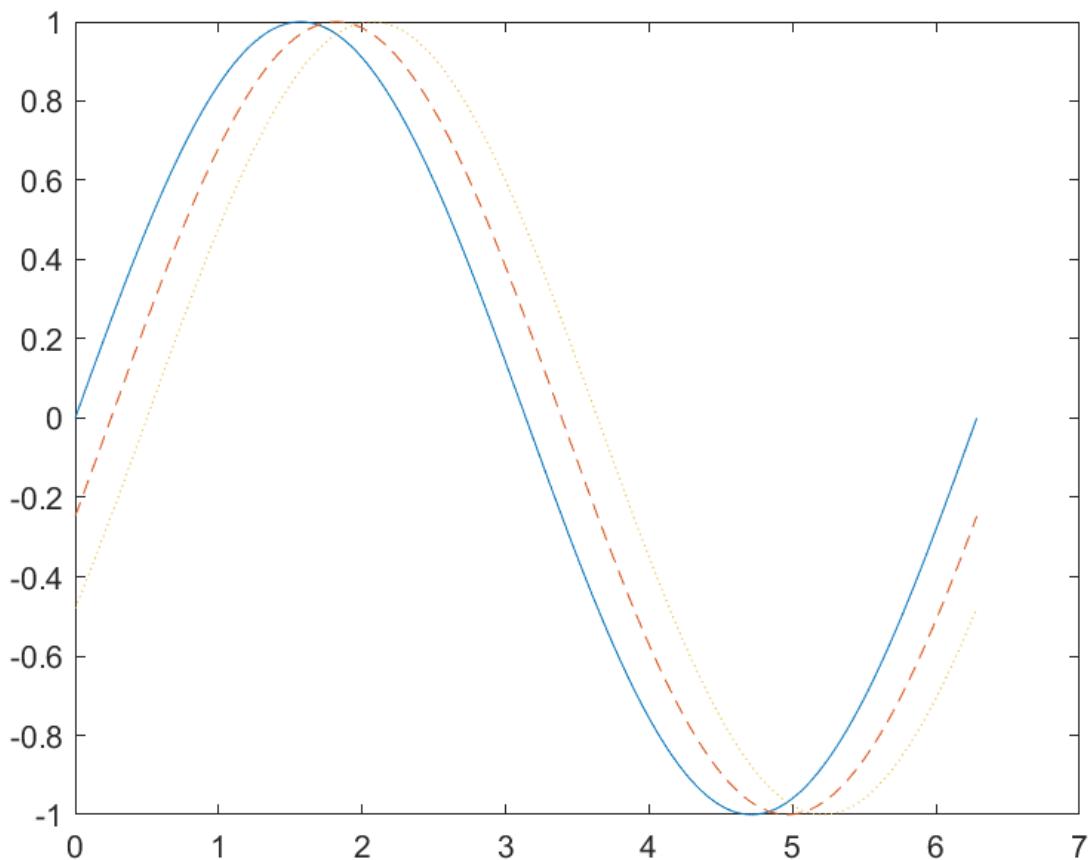


# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember



```
x = 0:pi/10:2*pi;
y1 = sin(x);
y2 = sin(x-0.25);
y3 = sin(x-0.5);

figure
% Untuk memberikan warna pada tiap grafik
% plot (x,y, '{kode warna}{kode gaya garis}')
plot(x,y1, 'g', x,y2, 'b--o', x,y3, 'c*')
```

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS

Keputih, Sukolilo, Surabaya

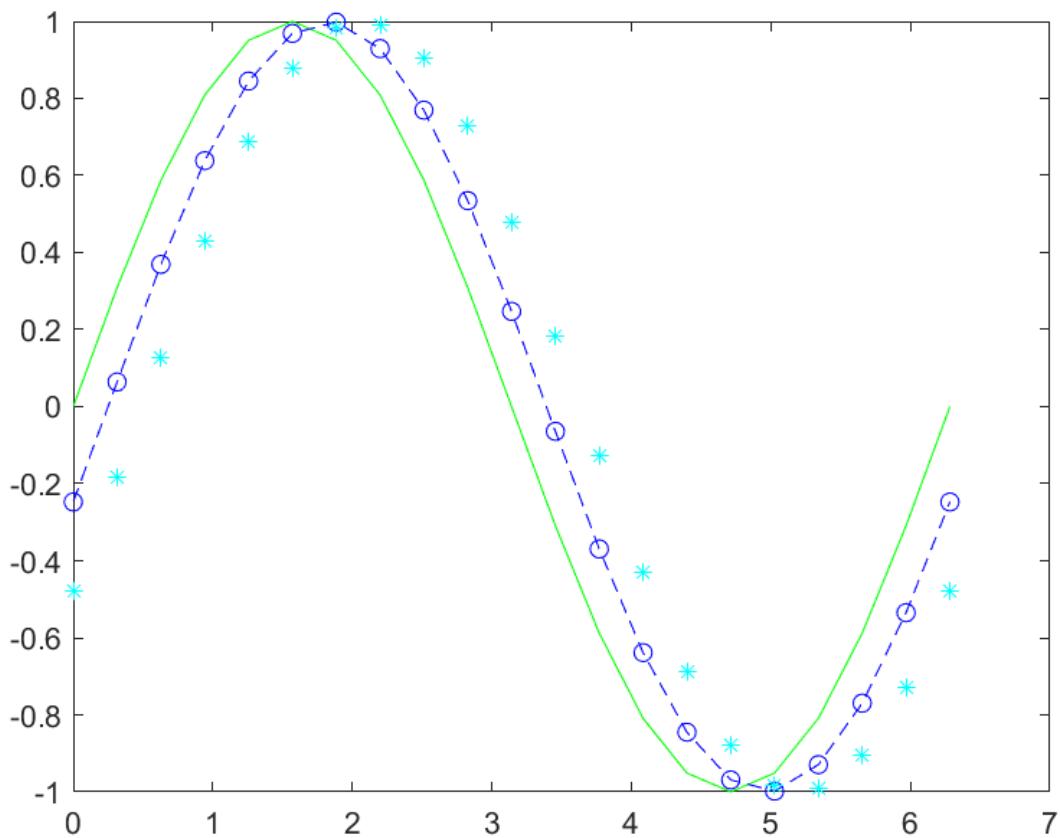


# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember



Untuk list style garis dan penanda yang dapat digunakan adalah sebagai berikut dan untuk warna yang dapat digunakan dapat dikombinasikan sedemikian rupa sehingga menghasilkan beragam warna. Untuk lebih lengkapnya dapat dilihat pada link berikut <https://www.mathworks.com/help/matlab/ref/plot.html>

Line Style	Description	Resulting Line
'-'	Solid line	_____
'--'	Dashed line	- - - - -
'.'	Dotted line	.....
'-.'	Dash-dotted line	- - - - -

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Marker	Description	Resulting Marker
'o'	Circle	○
'+'	Plus sign	+
'*' or ''	Asterisk	*
::	Point	•
'x'	Cross	×
--	Horizontal line	—
' '	Vertical line	
's'	Square	□
'd'	Diamond	◇
'^'	Upward-pointing triangle	△
'v'	Downward-pointing triangle	▽

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS

Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

'>'	Right-pointing triangle	▷
'<'	Left-pointing triangle	◁
'p'	Pentagram	☆
'h'	Hexagram	★

Colour Name	Short Name	RGB Triplet	Appearance
'red'	'r'	[1 0 0]	
'green'	'g'	[0 1 0]	
'blue'	'b'	[0 0 1]	
'cyan'	'c'	[0 1 1]	
'yellow'	'y'	[1 1 0]	
'black'	'k'	[0 0 0]	
'white'	'w'	[1 1 1]	

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS

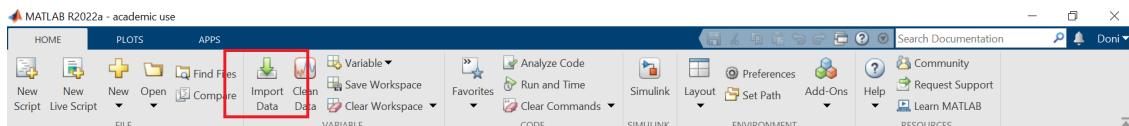
Keputih, Sukolilo, Surabaya



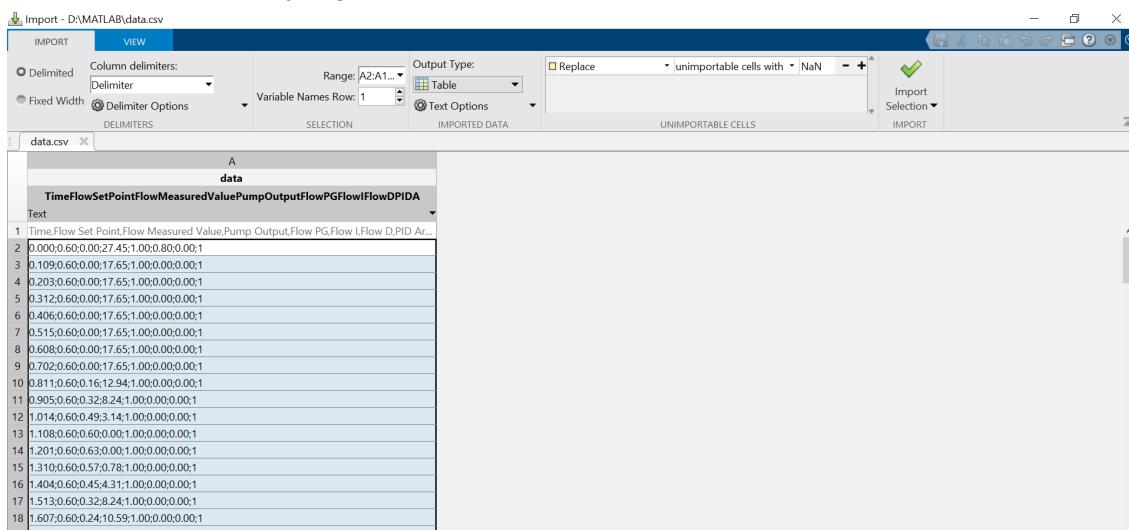
## 1.8. Import

Matlab yang merupakan sebagai salah satu software pengolahan data terbaik tentunya harus dimanfaatkan dengan baik. Dan sebelum melakukan pengolahan dan analisis data maka data yang telah didapatkan sebelumnya haruslah di import terlebih dahulu, berikut beberapa cara untuk melakukan import data pada Matlab.

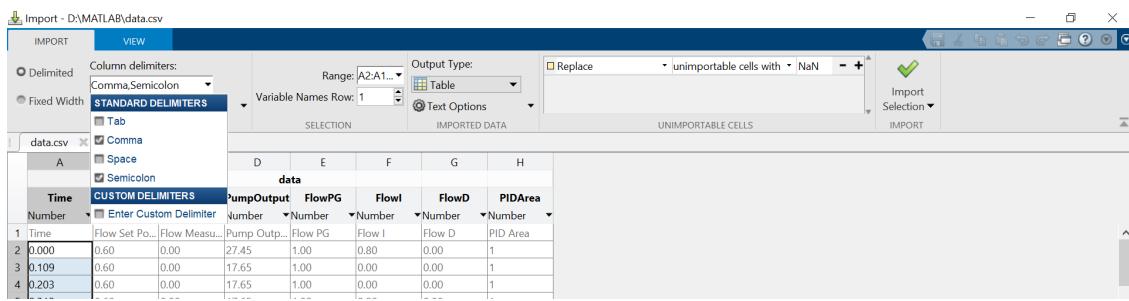
Cara pertama dapat dilakukan dengan cara menekan ikon import data pada bagian tab home



Setelah memilih data yang akan diimport akan muncul tab seperti berikut



Untuk memperbaiki data yang masih berantakan dapat dilakukan dengan memilih pada Column delimiters, column delimiters merupakan tanda pemisah antara satu data dengan data lainnya dan dapat dipilih berdasarkan tanda pemisah yang ada pada pilihan atau dapat di custom



Lalu untuk memberi nama pada setiap variabel data dapat dipilih pada Variable Names Row (artinya nama tiap variabel ada pada baris ke-) jika pada data sudah tertulis nama tiap variabelnya



The screenshot shows the MATLAB Import Data tool interface. The 'Import' tab is active. Under 'DELIMITERS', 'Delimited' is selected with 'Comma,Semicolon' chosen as the column delimiter. Under 'SELECTION', 'Variable Names Row: 1' is selected. Under 'IMPORTED DATA', 'Table' is selected as the output type. The data in the 'data.csv' file is displayed in a table format with columns: Time, FlowSetPoint, FlowMeasu..., PumpOutput, FlowPG, FlowI, FlowD, PIDArea. The first row contains column names, and subsequent rows contain numerical data.

Untuk jenis data yang akan diimport nantinya dapat dipilih pada Output Type

This screenshot shows the same MATLAB Import Data tool interface as the previous one, but with a different output type selection. Under 'IMPORTED DATA', 'Column vectors' is selected instead of 'Table'. The rest of the interface and data preview remain the same.

Dan setelah memilih data yang ingin diimport maka selanjutnya import data tersebut

This screenshot shows the MATLAB Import Data tool with a context menu open over the first row of the data table. The menu includes options such as 'Import Selection', 'Import Data', 'Generate Live Script', 'Generate Script', and 'Generate Function'. The 'Import Data' option is highlighted.

Data yang telah diimport dapat dilihat pada workspace

This screenshot shows the MATLAB Workspace browser. It lists two variables: 'FlowMeasuredValue' and 'Time'. Both are of type '153x1 double'. This indicates that the imported data has been successfully loaded into the MATLAB workspace.

Cara kedua yaitu dengan menggunakan syntax yang telah disediakan oleh matlab, yaitu

```
A = importdata('filename')
```



## BAB 2. Aljabar Linier

### 2.1. Pendahuluan

Persoalan matematis dapat diselesaikan menggunakan Symbolic Math Toolbox™ yang terdapat pada MATLAB. Persoalan matematis yang dapat diselesaikan mencakup penyederhanaan persamaan, penyelesaian persamaan, diferensial, integral, transformasi, dan operasi aljabar linier.

Aljabar linier merupakan cabang ilmu matematika yang mempelajari mengenai persamaan linier. Pada Bab II akan dibahas secara khusus bagaimana MATLAB, lebih detailnya Symbolic Math Toolbox™, membantu dalam penyelesaian aljabar linier dengan tingkat presisi komputasi yang tinggi melalui syntax-syntax yang tersedia

### 2.2. Operasi Matriks

Pada bagian ini akan dibahas mengenai pengoperasian matriks pada MATLAB, mulai dari membentuk matriks hingga operasi aritmatika matriks pada MATLAB.

#### 2.2.1. Membentuk Vektor dengan Matriks

##### 2.2.1.1. Menuliskan secara langsung

Membentuk Matriks  $1 \times 1$ ,

```
A = [3]
```

A = 3

Vektor baris, Matriks  $1 \times n$ ,

```
A = [1 2 3]
```

A = 1x3

1      2      3

Vektor kolom, Matriks  $n \times 1$ ,

```
A = [1;2;3]
```

A = 3x1

1

2

3



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

A = [1

2

3]

A = 3x1

1

2

3

A = [1 2 3]'

A = 3x1

1

2

3

Matriks n x n,

A = [1 2 3; 4 5 6; 7 8 9]

A = 3x3

1      2      3

4      5      6

7      8      9

A = [1 2 3

4 5 6

7 8 9]

A = 3x3

1      2      3

4      5      6

7      8      9

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS

Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

## 2.2.1.2. Menggunakan Colon Operator

Opsi lain untuk membentuk suatu vektor dapat menggunakan colon operator ":".

Membuat vektor baris dimulai dari 1 dan diakhiri 10, dengan increment 1, kita bisa menuliskan dengan script berikut:

```
1:10 % generate vector baris dari 1 sampai 10
```

A = 1x10

1    2    3    4    5    6    7    8    9    10

Membuat vektor baris dengan increment angka desimal, contoh 0.1

```
1.5:0.1:2 % increment 0.1
```

A = 1x6

1.5000    1.6000    1.7000    1.8000    1.9000    2.0000

Selain increment, kita bisa membuat decrement. Membuat vektor baris dari 2 ke 1.5 dengan decrement -0.1

```
2:-0.1:1.5 % decrement negative
```

A = 1x6

2.0000    1.9000    1.8000    1.7000    1.6000    1.5000

Jika kita membuat angka terakhir tidak bisa dicapai dengan increment, maka perhitungan MATLAB tidak akan berakhir di angka terakhir tersebut

```
2.7:0.1:pi % berhenti di 3.1, bukan 3.1416...
```

A = 1x5

2.7000    2.8000    2.9000    3.0000    3.1000

Colon operator akan berguna saat kita membutuhkan spacing yang spesifik antara elemen vektor yang berurutan

## 2.2.1.3. Menggunakan Linspace

linspace dapat digunakan untuk membuat vektor baris yang terpisah merata secara linear. linspace(x1,x2,n) berarti vektor baris sebanyak n elemen yang terpisah merata secara linear antara x1 dan x2. Jarak antara dua titik sebesar  $(x_2-x_1)/(n-1)$ .

Membuat vektor baris yang terpisah secara merata sebanyak 15 elemen dengan elemen pertama adalah 1 dan elemen terakhir adalah 10

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

```
linspace(1,10,15) % vektor baris yang terpisah secara merata sebanyak  
15
```

Membuat vektor baris yang terpisah secara merata sebanyak 10 elemen dengan elemen pertama adalah 0 dan elemen terakhir adalah  $2\pi$

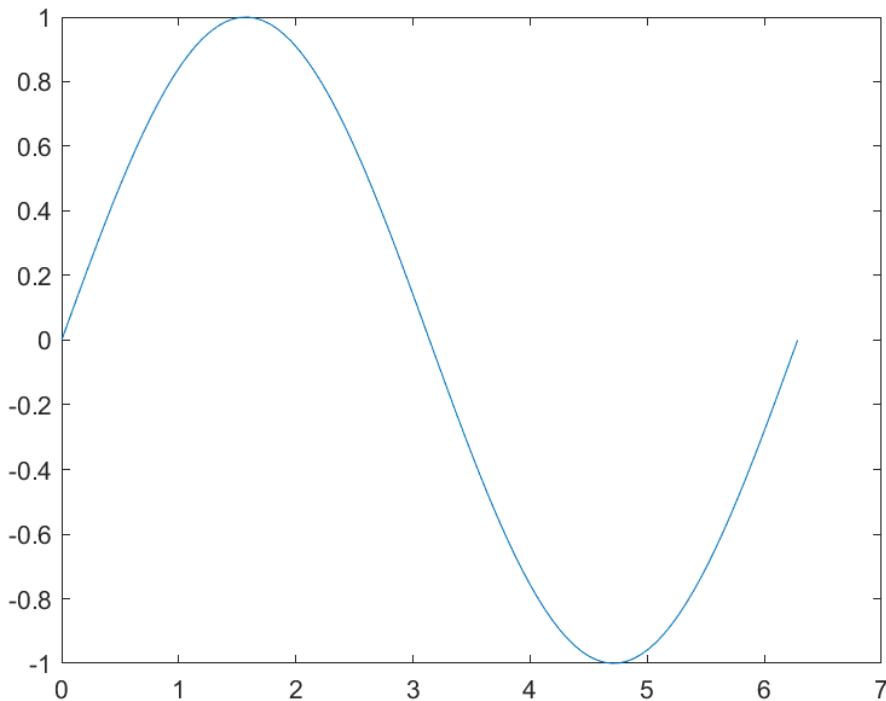
```
linspace(0,2*pi,10) % vektor baris yang terpisah secara merata  
sebanyak 10
```

`linspace` akan berguna jika ingin mendefinisikan vektor dengan elemen bilangan irasional

## 2.2.1.4. Membuat Vektor dengan Vektor

Membuat vektor menggunakan vektor. Contoh, kita dapat membuat vektor nilai sinus dari  $x = 0$  ke  $x = 2\pi$ , menggunakan script berikut:

```
X = linspace(0,2*pi,100);  
Y = sin(X); % Y adalah vektor  
plot(X,Y)
```



Y adalah vektor dengan ukuran yang sama dengan X.

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

## 2.2.2. Membentuk Matriks Khusus dengan Syntax

Selanjutnya akan dipelajari untuk pembentukan matriks-matriks khusus yang memiliki syntax tersendiri pada MATLAB.

### 2.2.2.1. Matriks Zeros

Matriks zeros atau umumnya disebut juga dengan matriks nol, dimana semua elemen matriksnya bernilai nol (0).

```
n = 3 %deklarasi variabel untuk ukuran matriks di code selanjutnya
```

```
n = 3
```

```
z = zeros(n) %membentuk matriks zeros n x n
```

```
z = 3x3
```

```
0 0 0  
0 0 0  
0 0 0
```

### 2.2.2.2. Matriks Ones

Syntax ones digunakan untuk membentuk matriks satuan, dimana semua elemen matriksnya bernilai satu (1).

```
o = ones(n)
```

```
o = 3x3
```

```
1 1 1  
1 1 1  
1 1 1
```

### 2.2.2.3. Matriks Eye

Matriks Eye umumnya lebih dikenal dengan sebutan matriks identitas.

```
i = eye(n)
```

```
i = 3x3
```

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://@controlsystem.its) [linkedin.com/company/its-control-system/](https://linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

## 2.2.2.4. Matriks Diagonal

Dapat dilakukan pembuatan matriks yang hanya terisi diagonalnya saja pada MATLAB dengan menggunakan *syntax diag*.

```
d = diag([1 2 3])
```

d = 3x3

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{matrix}$$

```
d = diag([1 1 1]) %membuat matriks identitas melalui syntax diag
```

d = 3x3

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

Syntax *diag* juga dapat digunakan untuk mendapatkan nilai diagonal utama suatu matriks

```
A %memanggil matriks A
```

A = 3x3

$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$

```
d = diag(A)
```

d = 3x1

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

1

5

9

Penggunaan *syntax diag* dua kali dapat membentuk matriks diagonal yang terdiri dari nilai diagonal utama dari matriks aslinya

```
d = diag(diag(A))
```

d = 3x3

1	0	0
0	5	0
0	0	9

## 2.2.2.5. Matriks Magic

Matriks *magic* merupakan matriks acak yang elemen penyusunnya adalah bilangan bulat dari 1 hingga  $n^2$  dengan jumlah baris dan kolom yang sama.

```
m = magic(n)
```

m = 3x3

8	1	6
3	5	7
4	9	2

```
n = 4
```

n = 4

```
m = magic(n)
```

m = 4x4

16	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

## 2.2.2.6. Matriks Random

Melalui MATLAB dapat dibentuk matriks dengan elemen penyusunnya dipilih secara acak, n adalah dimensi dari matriks dan elemen penyusun matriksnya adalah bilangan antara 0 sampai 1.

```
n = 3 %deklarasi variabel untuk ukuran matriks di code selanjutnya
```

```
n = 3
```

```
r = rand(n)
```

```
r = 3x3
```

```
0.9001 0.7803 0.4039  
0.3692 0.3897 0.0965  
0.1112 0.2417 0.1320
```

```
r = randn(n) %bilangan acak yang terdistribusi normal
```

```
r = 3x3
```

```
2.5260 -1.2571 0.7914  
1.6555 -0.8655 -1.3320  
0.3075 -0.1765 -2.3299
```

## 2.2.3. Menentukan Ukuran Matriks

*Length*, menghitung panjang dimensi array terbesar pada suatu matriks

```
m %memanggil matriks m
```

```
m = 4x4
```

```
16 2 3 13  
5 11 10 8  
9 7 6 12  
4 14 15 1
```

```
l = length(m)
```

```
l = 4
```

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

```
x = rand(3,7) %membentuk matriks random dengan ukuran 3 x 7
```

x = 3x7

0.1690	0.6477	0.2963	0.6868	0.6256	0.9294	0.4359
0.6491	0.4509	0.7447	0.1835	0.7802	0.7757	0.4468
0.7317	0.5470	0.1890	0.3685	0.0811	0.4868	0.3063

```
l = length(x)
```

l = 7

```
w = rand(8,4) %membentuk matriks random ukuran 8,4
```

w = 8x4

0.4909	0.2417	0.2963	0.6868
0.4893	0.4039	0.7447	0.1835
0.3377	0.0965	0.1890	0.3685
0.9001	0.1320	0.2963	0.6868
0.3692	0.9421	0.7447	0.1835
0.1112	0.5470	0.1890	0.3685
0.7803	0.6477	0.2963	0.6868
0.6491	0.4509	0.7447	0.1835

```
l = length(w)
```

l = 8

Dari dua contoh diketahui bahwa syntax length mengambil nilai terbesar antara kolom atau baris suatu matriks. Bila yang diukur adalah sebuah vektor, maka nilai length sama dengan jumlah elemen yang ada pada vektor tersebut. Bila mengukur himpunan kosong, nilai length sama dengan nol.

**Size**, mengukur dimensi matriks dalam bentuk vektor baris ([A,B]), artinya matriks yang diukur memiliki ukuran A x B.

```
sz = size(m)
```

m = 1x2

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://@controlsystem.its) [linkedin.com/company/its-control-system/](https://linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

4 4

```
sz = size(x)
```

m = 1x2

3 7

max dan min, digunakan untuk menentukan elemen terbesar dan terkecil pada matriks

```
x %memanggil matriks x
```

x = 3x7

0.1690	0.6477	0.2963	0.6868	0.6256	0.9294	0.4359
0.6491	0.4509	0.7447	0.1835	0.7802	0.7757	0.4468
0.7317	0.5470	0.1890	0.3685	0.0811	0.4868	0.3063

```
max(x) %mengambil nilai maksimal per kolom
```

ans = 1x7

0.7317	0.6477	0.7447	0.6868	0.7802	0.9294	0.4468
--------	--------	--------	--------	--------	--------	--------

```
min(x) %mengambil nilai minimal per kolom
```

ans = 1x7

0.1690	0.4509	0.1890	0.1835	0.0811	0.4868	0.3063
--------	--------	--------	--------	--------	--------	--------

## 2.2.4. Modifikasi Elemen-Elemen Matriks

Pada bagian ini akan dibahas mengenai cara melakukan ekstraksi elemen-elemen pada matriks, membalik elemen pada matriks, menghapus elemen pada matriks, menambahkan elemen pada matriks, mengganti elemen pada matriks, dan mengubah dimensi matriks.

### 2.2.4.1. Ekstraksi Matriks

MATLAB memungkinkan penggunanya untuk mengambil elemen-elemen tertentu dari suatu matriks.

```
A %memanggil matriks A
```

A = 3x3

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://@controlsystem.its) [linkedin.com/company/its-control-system/](https://linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

1	2	3
4	5	6
7	8	9

```
e = A(1,2) %A(baris,kolom)
```

```
e = 2
```

```
e = A(1,end) %A(baris, elemen pada kolom terakhir)
```

```
e = 3
```

```
e = A(end,1) %A(elemen pada baris terakhir, kolom)
```

```
e = 7
```

```
e = A(1,:) %ekstraksi semua elemen pada baris pertama
```

```
e = 1x3
```

1	2	3
---	---	---

```
e = A(:,1) %ekstraksi semua elemen pada kolom pertama
```

```
e = 3x1
```

1
4
7

```
e = A(1:2,3) %ekstraksi submatriks
```

```
e = 2x1
```

3
6

```
e = A(1:2,2:3) %ekstraksi submatriks
```

```
e = 2x2
```

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://twitter.com/@controlsystem.its) [linkedin.com/company/its-control-system/](https://linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

2      3

5      6

## 2.2.4.2. Membalik Elemen pada Matriks

Selanjutnya, tools MATLAB juga memungkinkan untuk membalik elemen pada suatu matriks

```
A %memanggil matriks A
```

A = 3x3

1	2	3
4	5	6
7	8	9

```
A1 = A([2 1 3], :) %membalik baris 1 dan 2 dari matriks A
```

A1 = 3x3

4	5	6
1	2	3
7	8	9

```
A2 = A(:,[2 1 3]) %membalik kolom 1 dan 2 dari matriks A
```

A2 = 3x3

2	1	3
5	4	6
8	7	9

```
A3 = A([2 1 3], [2 1 3]) %membalik baris dan kolom 1 dan 2 dari  
matriks A
```

A3 = 3x3

5	4	6
2	1	3
8	7	9

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

## 2.2.4.3. Membalik Elemen pada Matriks

Selanjutnya, tools MATLAB juga memungkinkan untuk membalik elemen pada suatu matriks

```
A % memanggil matriks A
```

```
A = 3x3
```

1	2	3
4	5	6
7	8	9

```
A1 = A([2 1 3], :) % membalik baris 1 dan 2 dari matriks A
```

```
A1 = 3x3
```

4	5	6
1	2	3
7	8	9

```
A2 = A(:,[2 1 3]) % membalik kolom 1 dan 2 dari matriks A
```

```
A2 = 3x3
```

2	1	3
5	4	6
8	7	9

```
A3 = A([2 1 3], [2 1 3]) % membalik baris dan kolom 1 dan 2 dari  
matriks A
```

```
A3 = 3x3
```

5	4	6
2	1	3
8	7	9

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://www.instagram.com/@controlsystem.its) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

## 2.2.4.4. Menghapus Elemen pada Matriks

MATLAB memungkinkan pengguna untuk menghapus baris dan kolom dari suatu matriks dengan menggunakan operator kosong

```
B = magic(5) %membuat matriks magic 5 x 5
```

```
B = 5x5
```

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

```
B(1,:) = [] %menghapus baris pertama
```

```
B = 4x5
```

23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

```
B(:,1) = [] %menghapus kolom pertama
```

```
B = 4x4
```

5	7	14	16
6	13	20	22
12	19	21	3
18	25	2	9

```
B([1 2],:) = [] %menghapus baris pertama dan kedua
```

```
B = 2x4
```

12	19	21	3
18	25	2	9

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

## 2.2.4.5. Menambahkan Elemen pada Matriks

Selain menghapus, MATLAB juga memiliki fungsi untuk menambahkan elemen pada matriks

```
B = [[6 13 20 22];B(1,:);B(2,:)] %menambahkan satu baris pada matriks B
```

B = 3x4

6	13	20	22
12	19	21	3
18	25	2	9

```
B = [[23 4 10]' B(:,1) B(:,2) B(:,3) B(:,4)] %menambahkan satu kolom pada matriks B
```

B = 3x5

23	6	13	20	22
4	12	19	21	3
10	18	25	2	9

## 2.2.4.6. Mengganti Elemen pada Matriks

Untuk mengganti elemen dapat dilakukan teknik berikut

```
A % memanggil matriks A
```

A = 3x3

1	2	3
4	5	6
7	8	9

```
A(1,1) = 0 %mengganti elemen pada baris pertama kolom pertama dengan 0
```

A = 3x3

0	2	3
4	5	6
7	8	9

```
A(1,:) = 1 % baris pertama dengan 1
```

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

A = 3x3

1	1	1
4	5	6
7	8	9

A(1,:) = [1 2 4] % mengganti baris pertama dengan 1, 2, dan 4

A = 3x3

1	2	4
4	5	6
7	8	9

## 2.2.4.7. Mengubah Dimensi Matriks

Untuk mengubah dimensi matriks, digunakan *syntax reshape*

reshape(A,1,9) % reshape(A,n,m) (m=kolom baru n=baris baru)

ans = 1x9

1	4	7	2	5	8	4	6	9
---	---	---	---	---	---	---	---	---

## 2.2.5. Transpose Matriks

Melakukan operasi transpose matriks pada MATLAB memungkinkan dengan menggunakan operator quote '

a = [1 2 3;4 5 6;7 8 9]

a = 3x3

1	2	3
4	5	6
7	8	9

a = a'

a = 3x3

1	4	7
2	5	8

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

3      6      9

## 2.2.6. Inverse Matriks

Seperti yang diketahui rumus invers matriks dibagi berdasarkan jumlah ordo. Misalkan memiliki sebuah matriks ordo 2 sebagai berikut

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$A^{-1} = \frac{1}{\det A} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Sedangkan untuk matriks ordo tinggi, rumus yang digunakan berbeda. Misalkan sebuah matriks ordo 3 sebagai berikut

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$A^{-1} = \frac{1}{\det A} \text{adj}(A)$$

Melalui MATLAB, dapat menghitung invers matriks dengan lebih sederhana dan berlaku untuk semua ordo, sebagai berikut

```
X = [1 0 2;-1 5 0;0 3 -9]
```

X = 3x3

1	0	2
-1	5	0
0	3	-9

```
Y = inv(X)
```

Y = 3x3

0.8824	-0.1176	0.1961
0.1765	0.1765	0.0392
0.0588	0.0588	-0.0980

```
Z = X^(-1)
```

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://www.instagram.com/@controlsystem.its) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Z = 3x3

$$\begin{array}{ccc} 0.8824 & -0.1176 & 0.1961 \\ 0.1765 & 0.1765 & 0.0392 \\ 0.0588 & 0.0588 & -0.0980 \end{array}$$

%pembuktian dengan hasil invers \* matriks original = matriks identitas

Y1 = Y\*X

Y1 = 3x3

$$\begin{array}{ccc} 1.0000 & 0.0000 & -0.0000 \\ 0 & 1.0000 & -0.0000 \\ 0 & -0.0000 & 1.0000 \end{array}$$

Z1 = Z\*X

Z1 = 3x3

$$\begin{array}{ccc} 1.0000 & 0.0000 & -0.0000 \\ 0 & 1.0000 & -0.0000 \\ 0 & -0.0000 & 1.0000 \end{array}$$

## 2.2.7. Determinan

Selain invers, MATLAB juga dapat menghitung determinan dari suatu matriks

X

X = 3x3

$$\begin{array}{ccc} 1 & 0 & 2 \\ -1 & 5 & 0 \\ 0 & 3 & -9 \end{array}$$

X1 = det(X)

X1 = -51

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://www.instagram.com/@controlsystem.its) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

```
P = [1 1 2; 1 1 2; 10 8 15]
```

P = 3x3

1	1	2
1	1	2
10	8	15

```
P1 = det(P) % karena det = 0, maka matriks P adalah matriks singular
```

P1 = 0

## 2.2.8. Rank

Rank adalah jumlah kolom yang *linearly independent* dalam sebuah matriks.

```
X
```

X = 3x3

1	0	2
-1	5	0
0	3	-9

```
k1 = rank(X)
```

k1 = 3

```
P = [1 1 2; 1 1 2; 10 8 15]
```

P = 3x3

1	1	2
1	1	2
10	8	15

```
k2 = rank(P)
```

k2 = 2

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://www.instagram.com/@controlsystem.its) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

## 2.2.9. Operasi Aritmatika

Pada bagian ini akan dibahas mengenai empat jenis operasi, yaitu penjumlahan, pengurangan, perkalian, dan pembagian matriks

### 2.2.9.1. Penjumlahan Matriks

Sebelum melakukan penjumlahan matriks, ada syarat yang harus dipenuhi, yaitu dimensi antar matriks yang akan dijumlahkan harus sama

$$A = [1 \ 3 \ 4; \ 2 \ 7 \ 9; \ 5 \ 6 \ 8]$$

A = 3x3

$$\begin{matrix} 1 & 3 & 4 \\ 2 & 7 & 9 \\ 5 & 6 & 8 \end{matrix}$$

$$B = [7 \ 3 \ 5; \ 6 \ 9 \ 1; \ 2 \ 4 \ 8]$$

B = 3x3

$$\begin{matrix} 7 & 3 & 5 \\ 6 & 9 & 1 \\ 2 & 4 & 8 \end{matrix}$$

$$S = A+B$$

S = 3x3

$$\begin{matrix} 8 & 6 & 9 \\ 8 & 16 & 10 \\ 7 & 10 & 16 \end{matrix}$$

### 2.2.9.2. Pengurangan Matriks

Operasi pengurangan matriks memiliki syarat yang sama dengan operasi penjumlahan

$$M = A-B$$

M = 3x3

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

-6	0	-1
-4	-2	8
-3	2	0

## 2.2.9.3. Perkalian Matriks

Operasi perkalian matriks memiliki syarat, yaitu jumlah kolom matriks pertama sama dengan jumlah baris matriks kedua

$$C = A * B$$

$$C = 3 \times 3$$

33	46	40
74	105	89
87	101	95

$$C1 = A.*B \text{ % perkalian per elemen}$$

$$C1 = 3 \times 3$$

7	9	20
12	63	9
10	24	64

## 2.2.9.4. Pembagian Matriks

Teknik pembagian dibagi menjadi dua yaitu *left division* dan *right division*. *Left division* digunakan untuk mendapatkan nilai X dari persoalan  $A * X = B$ , sedangkan *right division* digunakan untuk mendapatkan nilai X dari persoalan  $X * A = B$ . Syarat untuk melakukan operasi pembagian matriks adalah jumlah kolom matriks pertama sama dengan jumlah baris matriks kedua, serta determinan matriks kedua tidak sama dengan nol. Misalkan kita memiliki persoalan  $A * B = C$ ,

$$D1 = A \setminus C$$

$$D1 = 3 \times 3$$

7.0000	3.0000	5.0000
6.0000	9.0000	1.0000
2.0000	4.0000	8.0000

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

D2 = C/B

D2 = 3x3

1.0000	3.0000	8.0000
2.0000	7.0000	9.0000
5.0000	6.0000	8.0000

D3 = A./B % pembagian per elemen

D1 = 3x3

0.1429	1.0000	0.8000
0.3333	0.7778	9.0000
2.5000	1.5000	1.0000

## 2.2.10. Operasi Vektor

### 2.2.10.1. Dot Product

Terdapat syntax *dot*, yang dapat digunakan untuk memperoleh nilai *dot product* dari dua matriks

dotproduct = dot(A,B)

dotproduct = 1x3

29	96	93
----	----	----

% dotproduct(1) = dot product dari A(:,1) dan B(:,1)

### 2.2.10.2. Cross Product

Terdapat syntax *cross*, yang dapat digunakan untuk memperoleh nilai *cross product* dari dua matriks

crossproduct = cross(A,B)

crossproduct = 3x3

-26	-26	64
33	6	8
-8	8	-41

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

```
% crossproduct(1) = cross product dari A(:,1) dan B(:,1)
```

## 2.2.11. Eigenvalue dan Eigenvector

Melalui MATLAB juga bisa didapatkan nilai eigen dan vektor eigen dengan menggunakan syntax *eig*

```
N = magic(3)
```

N = 3x3

8	1	6
3	5	7
4	9	2

```
[V,D] = eig(N)
```

```
% eigenvalue dinyatakan dalam diagonal matriks D
```

```
% eigenvector dinyatakan dalam matriks V
```

V = 3x3

-0.5774	-0.8131	-0.3416
-0.5774	0.4714	-0.4714
-0.5774	0.3416	0.8131

D = 3x3

15.0000	0	0
0	4.8990	0
0	0	-4.8990

## 2.3. Penyelesaian Sistem Persamaan Linier

Pada bagian ini akan dibahas mengenai bagaimana mendefinisikan persamaan linier hingga bagaimana menemukan solusi dari suatu persamaan linier.

### 2.3.1. Mendefinisikan Sistem Persamaan Linier

Misalkan diketahui sebuah persamaan linier sebagai berikut

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://@controlsystem.its) [linkedin.com/company/its-control-system/](https://linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

$$\begin{aligned}2x + y + z &= 2 \\-x + y - z &= 3 \\x + 2y + 3z &= -10\end{aligned}$$

Untuk mendefinisikan persamaan linier ke MATLAB dapat digunakan teknik sebagai berikut

```
syms x y z %membuat simbol variabel
```

```
p1 = 2*x + y + z == 2
```

```
p1 =
```

$$2x + y + z = 2$$

```
%pada MATLAB, simbol = adalah sama, sedangkan simbol == adalah sama dengan
```

```
p2 = -x + y - z == 3
```

```
p2 =
```

$$-x + y - z = 3$$

```
p3 = x + 2*y + 3*z == -10
```

```
p3 =
```

$$x + 2y + 3z = -10$$

## 2.3.2. Menyelesaikan Sistem Persamaan Linier

Selanjutnya, untuk menyelesaikan persoalan persamaan linier, terdapat dua teknik yang dapat digunakan

### 2.3.2.1. Linsolve

Syntax *linsolve* digunakan untuk menyelesaikan persamaan linier dalam bentuk matriks  $AX = B$ . Untuk mengonversi persamaan ke bentuk matriks, dapat menggunakan syntax *equationsToMatrix*.

```
[A,B] = equationsToMatrix([p1,p2,p3], [x,y,z])
```

```
A =
```

$$\begin{matrix} 2 & 1 & 1 \end{matrix}$$

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

-1      1      -1

1      2      3

B =

2

3

-10

```
X = linsolve(A,B)
```

X =

3

1

-5

Dari X, dapat diketahui nilai x = 3, y = 1, dan z = -5

### 2.3.2.2. Solve

Syntax `solve` digunakan untuk menyelesaikan persamaan linier tanpa perlu mengonversi bentuk persamaan menjadi matriks.

```
sol = solve([p1, p2, p3], [x, y, z]);
```

```
x = sol.x
```

x =

3

```
y = sol.y
```

y =

1

```
z = sol.z
```

z = -5

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)



## BAB 3. Persamaan Diferensial

### 3.1. Solusi Persamaan

Untuk mendapatkan solusi analitis persamaan diferensial di Matlab dapat menggunakan fungsi `dsolve` dengan atau tanpa kondisi awal. Sebagai contoh digunakan persamaan diferensial

#### 3.1.1. PD orde 1

$$\frac{dy}{dt} = -0.5y$$

Mendefinisikan fungsi `y`

```
syms y(t)
```

Mendefinisikan PD yang akan diselesaikan dengan fungsi `diff` dan persamaan dengan `==`

```
ode = diff(y,t) == -0.5*y
```

`ode(t) =`

$$\frac{\partial}{\partial t} y(t) = -\frac{y(t)}{2}$$

Cari solusi PD tersebut dengan fungsi `dsolve`

```
ySol(t) = dsolve(ode)
```

`ySol(t) =`

$$C_1 e^{-\frac{t}{2}}$$

Pada penyelesaian persamaan diferensial tanpa kondisi awal masih terdapat nilai konstanta  $C_1$ , untuk mendapatkan nilai konstanta tersebut maka dibutuhkan nilai kondisi awal  $y(0)$ . Dimisalkan  $y(0) = 0.5$ , maka dilakukan sedikit perubahan untuk fungsi `dsolve` sebagai berikut:

```
cond = y(0) == 0.5;
ySol(t) = dsolve(ode,cond)
```



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

$$ySol(t) =$$

$$\frac{e^{-\frac{t}{2}}}{2}$$

### 3.1.2. PD orde 2 dengan kondisi awal

$$\frac{d^2y}{dt^2} = \sin(3t) + y$$

$$y(0) = 1$$

$$y'(0) = 0$$

Untuk kondisi awal  $y'(0)=0$  turunkan fungsi y Dy=diff(y,t) lalu definisikan Dy(0)==0.

```
syms y(t)
Dy = diff(y,t);
ode = diff(y,t,2) == sin(3*t)+y;
cond1 = y(0) == 1;
cond2 = Dy(0) == 0;
```

Solusi PD untuk y dapat dicari dengan

```
conds = [cond1 cond2];
ySol(t) = dsolve(ode,conds)
```

$$ySol(t) =$$

$$\frac{7 e^{-t}}{20} - \frac{\sin(3 t)}{10} + \frac{13 e^t}{20}$$

### 3.1.3. Beberapa contoh PD

PD orde 1

$$\frac{dy}{dx} = x + y$$

$$y(0) = 0$$

```
syms y(x)
```

```
ode = diff(y,x) == x+y;
```

```
cond = y(0) == 0;
```

```
ySol(x) = dsolve(ode,cond)
```

$$ySol(x) = e^x - x - 1$$

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



<p>PD orde 2</p> $\frac{d^2y}{dt^2} + 0.8 \frac{dy}{dt} + 0.2y = 0$	<pre>syms y(t) ode = diff(y,t,2)+0.8*diff(y,t)+0.2*y == 0; ySol(t) = dsolve(ode)</pre> <p><math>ySol(t) =</math></p> $C_1 \cos\left(\frac{t}{5}\right) e^{-\frac{2t}{5}} - C_2 \sin\left(\frac{t}{5}\right) e^{-\frac{2t}{5}}$
<p>PD non linear tanpa kondisi awal</p> $\frac{dy}{dt} - \frac{1}{y} = 0$ <p>Persamaan ini memiliki lebih dari satu penyelesaian</p>	<pre>syms y(t) ode = diff(y,t)-1/y == 0; ySol(t) = dsolve(ode)</pre> <p><math>ySol(t) =</math></p> $\begin{pmatrix} \sqrt{2} & \sqrt{C_1+t} \\ -\sqrt{2} & \sqrt{C_1+t} \end{pmatrix}$

### 3.2. Solusi Numerik

Teknik numerik dapat digunakan untuk menyelesaikan persamaan diferensial ketika metode analitik sulit digunakan atau saat kita bekerja dengan data empirik.

#### 3.2.1. Metode euler

Algoritma metode euler:

- Definisikan  $y' = f(x_i, y_i)$
- Inisialisasi  $x_0, y_0$
- Input batas akhir nilai  $x_{final}$
- Tentukan step  $h$  dan jumlah  $N = (x_{final} - x_0)/h$
- Untuk  $n = 0, 1, \dots, N$  hitung

$$x_{n+1} = x_n + h$$

$$y_{n+1} = y_n + h f(x_n, y_n)$$

- Output  $x_{n+1}, y_{n+1}$

Contoh untuk  $y' = x + y$  dengan kondisi awal  $y(0) = 0$  dan  $h = 0.2$

```
clear;clc
```



```
f = @(x,y)(x+y); % Mendefinisikan persamaan diferensial
x(1) = 0; % Inisialisasi x
y(1) = 0; % Inisialisasi y
xn = 1; % Batas akhir x
h = 0.2; % Step size
N = (xn-x(1))/h; % Banyaknya iterasi
for n=1:N
    y(n+1) = y(n) + h*f(x(n),y(n));
    x(n+1) = x(n) + h;
end
```

Tabel hasil perhitungan dan perbandingan metode euler dan analitik tiap iterasi

x	y (euler)	y (analitis)	error
0.000000	0.000000	0.000000	0.000000
0.200000	0.000000	0.021403	0.021403
0.400000	0.040000	0.091825	0.051825
0.600000	0.128000	0.222119	0.094119
0.800000	0.273600	0.425541	0.151941
1.000000	0.488320	0.718282	0.229962

$$mean\ error = 0.0915417$$

### 3.2.2. Metode runge-kutta orde 4

Algoritma metode runge-kutta orde 4:

- Definisikan  $y' = f(x_i, y_i)$
- Inisialisasi  $x_0, y_0$
- Tentukan step  $h$  dan jumlah  $N = (x_{final} + x_0)/h$
- Untuk  $n = 0, 1, \dots, N$  hitung

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1) \\ k_3 &= hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2) \\ k_4 &= hf(x_n + h, y_n + k_3) \\ x_{n+1} &= x_n + h \\ y_{n+1} &= y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned}$$

- Output  $x_{n+1}, y_{n+1}$

Contoh untuk  $y' = x + y$  dengan kondisi awal  $y(0) = 0$  dan  $h = 0.2$



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

```
clear;clc

f = @(x,y)(x+y);      % Mendefinisikan persamaan diferensial

x(1) = 0;                % Inisialisasi x

y(1) = 0;                % Inisialisasi y

xn = 1;                  % Batas akhir x

h = 0.2;                  % Step size

N = (xn-x(1))/h;        % Banyaknya iterasi

for n=1:N            %menghitung solusi dari x0 sampai xn dengan step size h

%formula RK orde 4

    k1 = h*f(x(n),y(n));

    k2 = h*f(x(n)+0.5*h,y(n)+0.5*k1);

    k3 = h*f(x(n)+0.5*h,y(n)+0.5*k2);

    k4 = h*f(x(n)+h,y(n)+k3);

    y(n+1) = y(n) + (1/6)*(k1 + 2*k2 + 2*k3 + k4);

    x(n+1) = x(n) + h;           %increment x dengan step size h

end
```

Tabel hasil perhitungan dan perbandingan metode RK4 dan analitik tiap iterasi

x	y (RK4)	y (analitis)	error
0.000000	0.000000	0.000000	0.000000
0.200000	0.021400	0.021403	0.000003
0.400000	0.091818	0.091825	0.000007
0.600000	0.222106	0.222119	0.000012
0.800000	0.425521	0.425541	0.000020
1.000000	0.718251	0.718282	0.000031

$$\text{mean error} = 0.000012$$

### 3.3. ODE23 and ODE45 Solver Matlab

ODE23 dan ODE45 adalah fitur ODE solver numerik yang ada di Matlab. Dalam metode ini akan diselesaikan Persamaan Diferensial orde pertama yaitu:

$$\frac{dy}{dx} = x + y$$

$$y(0) = 0$$

$$h = 0.2$$

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

### 3.3.1. ODE23

Fungsi yang digunakan dalam metode ini adalah  $[x,y] = \text{ode23}(\text{odefun},\text{tspan},y_0)$ . Dengan  $x$  adalah evaluation point yang dikembalikan dalam bentuk vektor array,  $y$  adalah solusi dalam bentuk vektor array,  $\text{odefun}$  adalah fungsi yang akan diselesaikan,  $\text{tspan}$  adalah interval  $x$  antara  $[x_0 \ xf]$  apabila ingin diberi step size  $h$  maka menjadi  $[x_0:h:xf]$ , dan  $y_0$  adalah kondisi awal dari persamaan diferensial.

```
clear;clc  
  
dy = @(x,y)(x+y); % Persamaan Diferensial  
  
x0 = 0; % Inisialisasi x / interval awal  
  
xn = 1; % interval akhir  
  
y0 = 0; % initial condition y(0)  
  
h = 0.2; % step size  
  
[x,y] = ode23(dy,[x0:h:xn],y0);
```

Tabel hasil perhitungan dan perbandingan ode23 dan analitik tiap iterasi

x	y (ode23)	y (analitis)	error
0.000000	0.000000	0.000000	0.000000
0.200000	0.021402	0.021403	0.000001
0.400000	0.091815	0.091825	0.000010
0.600000	0.222093	0.222119	0.000026
0.800000	0.425492	0.425541	0.000049
1.000000	0.718207	0.718282	0.000075

*mean error = 0.000027*

### 3.3.2. ODE45

Fungsi yang digunakan dalam metode ini adalah  $[x,y] = \text{ode45}(\text{odefun},\text{tspan},y_0)$ . Dengan  $x$  adalah evaluation point yang dikembalikan dalam bentuk vektor array,  $y$  adalah solusi dalam bentuk vektor array,  $\text{odefun}$  adalah fungsi yang akan diselesaikan,  $\text{tspan}$  adalah interval  $x$  antara  $[x_0 \ xf]$  apabila ingin diberi step size  $h$  maka menjadi  $[x_0:h:xf]$ , dan  $y_0$  adalah kondisi awal dari persamaan diferensial.

```
clear;clc  
  
dy = @(x,y)(x+y); % Persamaan Diferensial  
  
x0 = 0; % Inisialisasi x / interval awal  
  
xn = 1; % interval akhir  
  
y0 = 0; % initial condition y(0)  
  
h = 0.2; % step size
```

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

```
[x,y] = ode45(dy,[x0:h:xn],y0);
```

Tabel hasil perhitungan dan perbandingan ode45 dan analitik tiap iterasi

x	y (ode45)	y (analitis)	error
0.000000	0.000000	0.000000	0.000000
0.200000	0.021403	0.021403	0.000000
0.400000	0.091825	0.091825	0.000000
0.600000	0.222119	0.222119	0.000000
0.800000	0.425541	0.425541	0.000000
1.000000	0.718282	0.718282	0.000000

*mean error = 0*

### 3.4. Tabel Perbandingan Solusi Numerik

Metode	Mean error
Euler	0.0915417
Runge-Kutta Orde 4	0.000012
ode23	0.000027
ode45	0

Berdasarkan keempat metode yang digunakan, solusi dengan ode45 adalah yang terbaik berdasarkan MAE.

*\*script untuk perbandingan tabel ada di lampiran*

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



## BAB 4. Transformasi

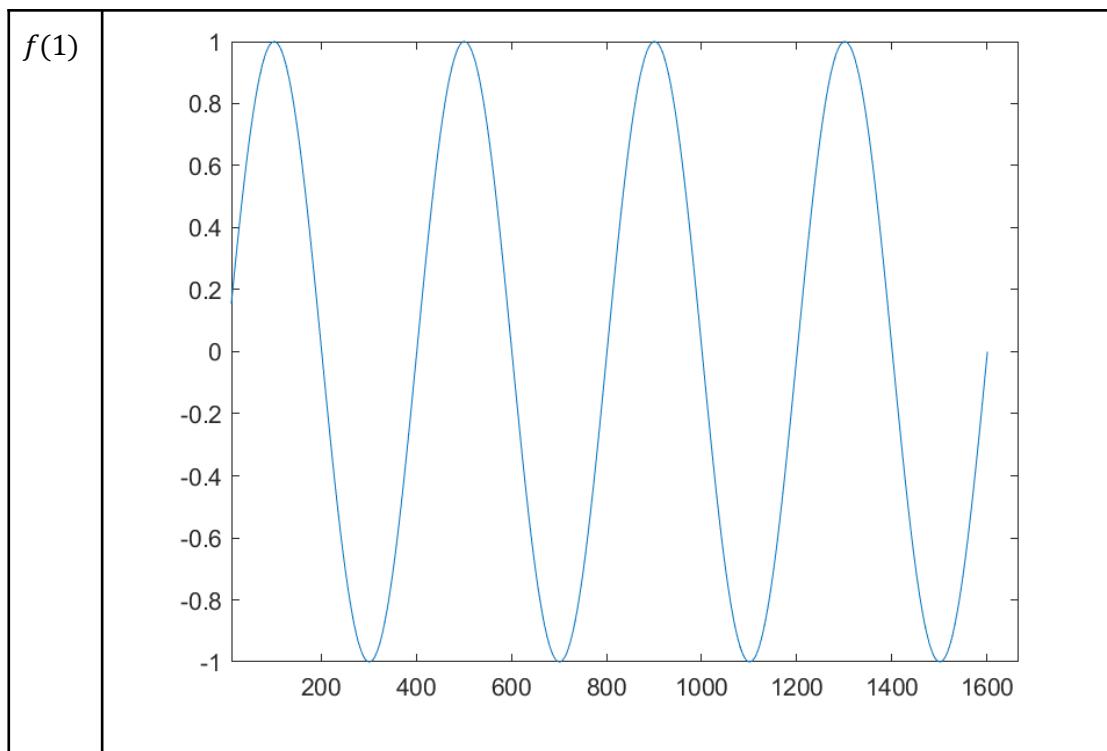
### 4.1. Pendahuluan

Transformasi atau transformasi integral merupakan suatu metode matematis yang dapat merepresentasikan suatu fungsi ke dalam bentuk lain atau dari suatu domain ke domain lainnya. Ada berbagai macam transformasi integral yang dapat dipelajari diantaranya adalah transformasi fourier, transformasi laplace, transformasi weierstrass, transformasi mellin, dan lainnya. Pada bab kali ini akan dibahas dua macam transformasi yang umumnya akan dipelajari di teknik elektro yaitu transformasi fourier dan transformasi laplace.

### 4.2. Transformasi Fourier Kontinu

Transformasi Fourier sendiri merupakan salah satu transformasi integral yang dapat merepresentasikan domain waktu  $x(t)$  ke dalam domain frekuensi  $X(\omega)$ . Penemu dari konsep Fourier, Joseph Fourier, mengemukakan bahwa "Setiap fungsi periodik (sinyal) dapat dibentuk dari penjumlahan gelombang-gelombang sinus dan cosinus". Sebagai contoh, sinyal kotak merupakan penjumlahan dari fungsi-fungsi sinus sebagai berikut

$$f(x) = \sin(x) + \frac{\sin(3x)}{3} + \frac{\sin(5x)}{5} + \frac{\sin(7x)}{7} + \frac{\sin(9x)}{9} + \dots$$



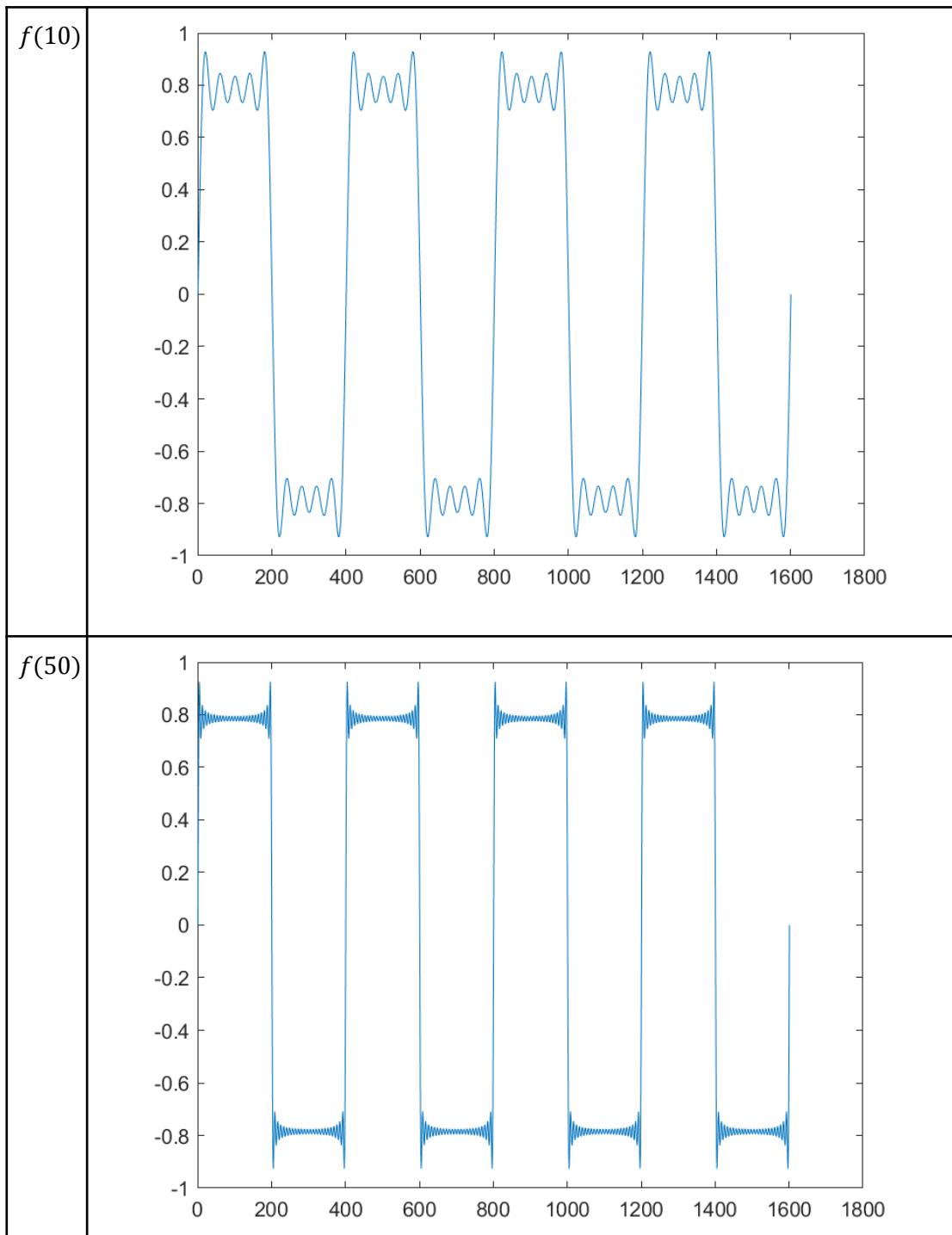


# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember



## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS

Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://www.instagram.com/@controlsystem.its) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)

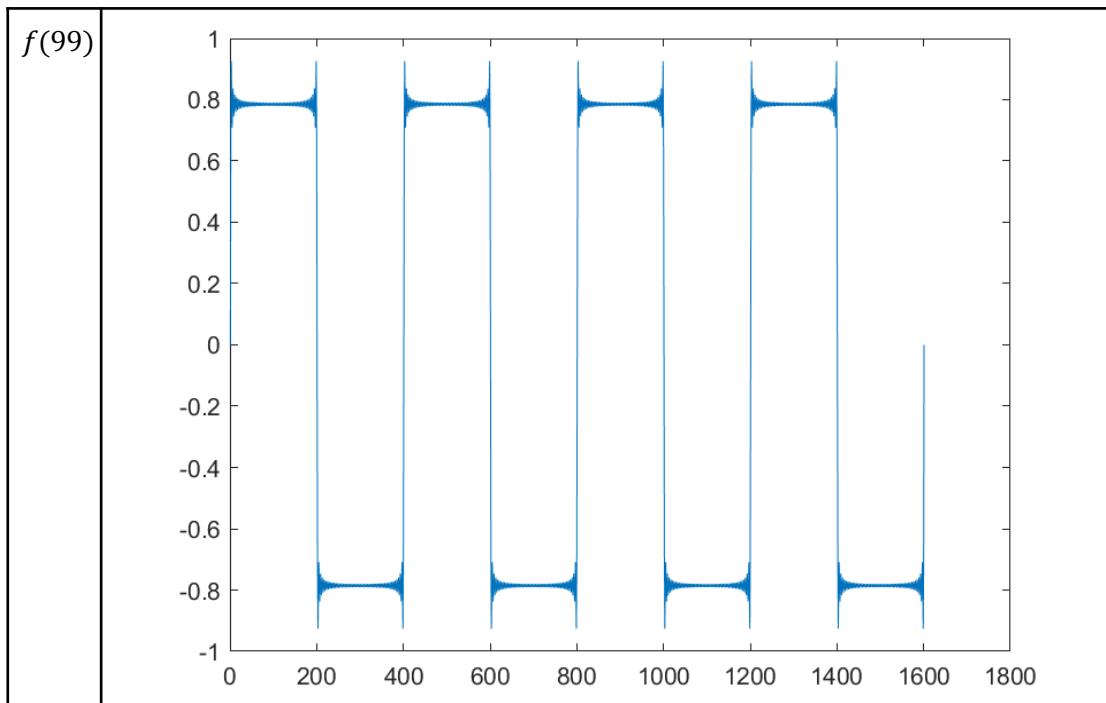


# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember



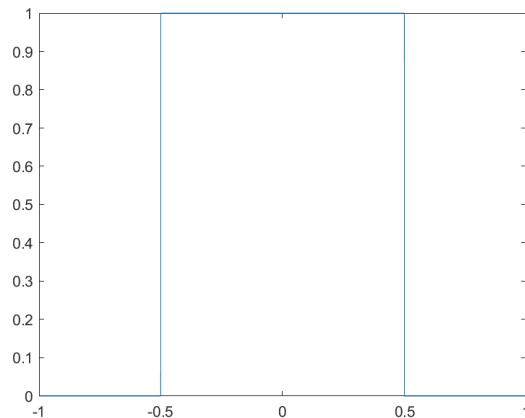
Fungsi diatas merupakan salah satu contoh dari deret fourier dan dari deret tersebutlah transformasi fourier diturunkan. Rumus dari transformasi fourier adalah sebagai berikut

$$X(\omega) = \mathcal{F}\{x(t)\} = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt$$

Catatan,  $e^{-j\omega t} = \cos(\omega t) - j \sin(\omega t)$

Sebagai contoh untuk transformasi fourier, suatu fungsi  $\text{rect}(t)$  yang merupakan fungsi rectangle dari  $-1/2 \leq t \leq 1/2$  akan dicari transformasi fouriernya

```
syms t %mendefinisikan variabel waktu t  
fplot(rectangularPulse(t), [-1 1])
```



## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS

Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

$$X(\omega) = \mathcal{F}\{x(t)\} = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt$$

$x(t)$  merupakan fungsi rectangle sehingga hanya memiliki nilai 1 ketika  $-1/2 \leq t \leq 1/2$ , maka

$$X(\omega) = \int_{-1/2}^{1/2} (1)e^{-j\omega t} dt$$

$$X(\omega) = -\frac{1}{j\omega} [e^{-j\omega t}]_{-1/2}^{1/2} = -\frac{1}{j\omega} (e^{\frac{-j\omega}{2}} - e^{\frac{j\omega}{2}})$$

$$X(\omega) = -\frac{1}{j\omega} (e^{\frac{-j\omega}{2}} - e^{\frac{j\omega}{2}}) \times \left(\frac{2}{2}\right) = \frac{2}{\omega} \left( \frac{e^{\frac{-j\omega}{2}} - e^{\frac{j\omega}{2}}}{2j} \right) = \left(\frac{2}{\omega}\right) (\sin(\frac{\omega}{2}))$$

Dari sini dapat dilihat bahwa metode penyelesaian transformasi fourier membutuhkan sedikit waktu untuk penyelesaiannya, dengan adanya matlab sebagai software komputasi akan mempermudah proses pengerjaan suatu masalah salah satunya yaitu transformasi fourier ini. Dengan soal yang sama akan diselesaikan menggunakan program pada matlab sebagai berikut

```
syms t
f=rectangularPulse(t)
fourier(f) %syntax untuk fourier transform
```

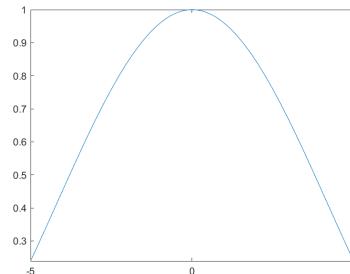
ans =

$$\frac{\sin(\frac{\omega}{2}) + \cos(\frac{\omega}{2})i}{\omega} - \frac{-\sin(\frac{\omega}{2}) + \cos(\frac{\omega}{2})i}{\omega}$$

Dari jawaban tersebut dapat kita sederhanakan menjadi

$$\frac{\sin(\frac{\omega}{2}) + \cos(\frac{\omega}{2})i}{\omega} - \frac{-\sin(\frac{\omega}{2}) + \cos(\frac{\omega}{2})i}{\omega} = \frac{\sin(\frac{\omega}{2}) + \cos(\frac{\omega}{2})i + \sin(\frac{\omega}{2}) - \cos(\frac{\omega}{2})i}{\omega} = \left(\frac{2\sin(\frac{\omega}{2})}{\omega}\right)$$

```
fplot(fourier(f))
```



## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Untuk contoh lainnya akan digunakan persamaan diferensial. Perlu diketahui bahwa persamaan diferensial pada transformasi fourier memiliki konsep yang sama hanya saja perlu diingat beberapa persamaan berikut

$$\mathcal{F}\{x'(t)\} = i\omega \mathcal{F}\{x(t)\} \text{ dan } \mathcal{F}\{x''(t)\} = -\omega^2 \mathcal{F}\{x(t)\}$$

Sebagai contoh akan digunakan persamaan orde 1 yaitu  $\frac{dy}{dt} + 5y = 1$ .

```
syms t w y(t) Y  
f = diff(y,t)+5*y-1
```

f(t) =

$$\frac{\partial}{\partial t}y(t) + 5y(t) - 1$$

```
F=fourier(f,t,w)
```

F =

$$5 \text{ fourier}(y(t), t, w) - 2\pi\partial(w) + w \text{ fourier}(y(t), t, w)i$$

```
F=subs(F,fourier(y(t), t, w),Y)
```

F =

$$5 Y - 2\pi\partial(w) + Y w i$$

```
ys=solve(F,Y) %untuk mencari nilai Y
```

ys =

$$\frac{2\pi\partial(w)}{5+w i}$$

Untuk contoh lainnya yaitu menggunakan bilangan euler. Perlu diingat pula bahwa dalam menggunakan bilangan euler sebagai fungsi dalam transformasi fourier perlu ditambahkan nilai absolut pada fungsi waktunya.

```
g=exp(-5*abs(t))
```

g =

$$e^{-5|t|}$$

```
gf=fourier(g)
```

gf =

$$\frac{10}{w^2+25}$$

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

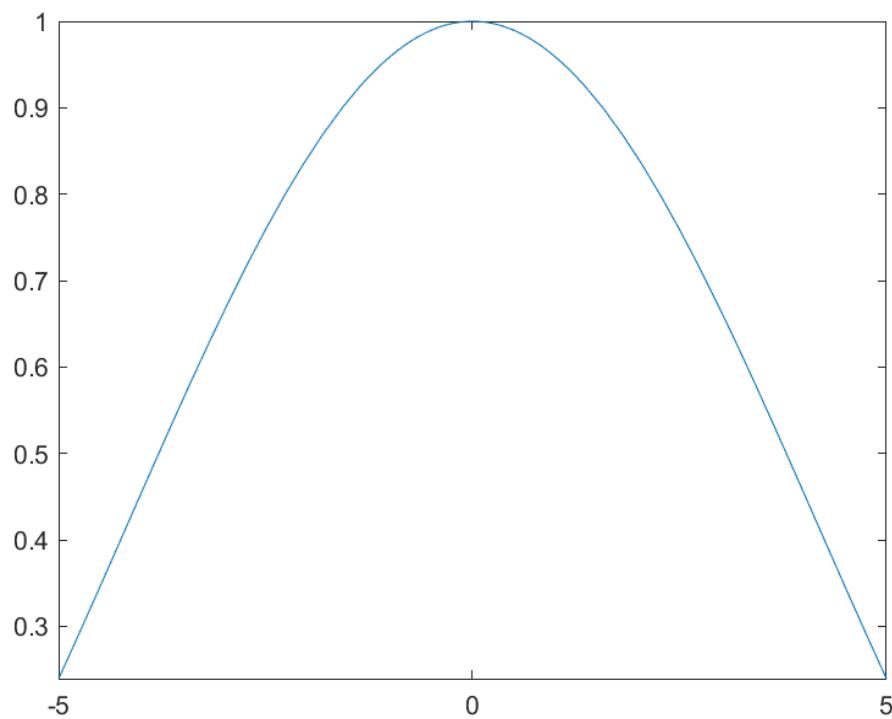
#### 4.3. Transformasi Fourier Balik (Inverse)

Setelah mempelajari bagaimana cara melakukan transformasi fourier, selanjutnya akan dipelajari juga invers dari transformasi fourier atau perubahan domain frekuensi ke domain waktu. Rumus dari transformasi fourier adalah sebagai berikut

$$x(t) = \mathcal{F}^{-1}\{X(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{-j\omega t} d\omega$$

Untuk contoh yang akan kita pakai adalah dengan menggunakan  $\frac{2\sin(\frac{\omega}{2})}{\omega}$  yang kita dapatkan pada contoh transformasi fourier sebelumnya. Untuk syntax yang digunakan adalah sebagai berikut

```
syms w %definisikan variabel w  
g=(2*sin(w/2))/w;  
fplot(g)
```



ifourier(g)

ans =



# Laboratorium Kontrol dan Otomasi

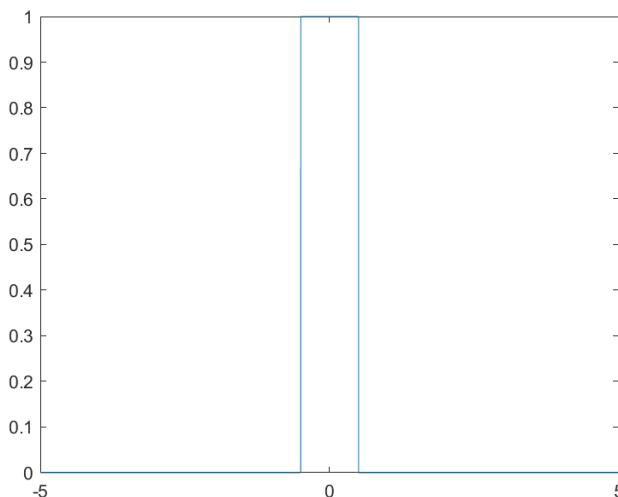
Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Note : heaviside merupakan fungsi yang keluarannya hampir sama seperti pada sinyal step.

```
fplot(ifourier(g))
```



## 4.4. Transformasi Laplace

Transformasi laplace adalah metode transformasi integral yang mengubah fungsi dalam domain real (biasanya dalam domain waktu) ke domain variabel kompleks (dalam domain frekuensi kompleks atau domain s). (MATLAB HANYA DAPAT MENYELESAIKAN TL UNILATERAL).

$$\mathcal{L}\{x(t)\} = X(s) = \int_0^{\infty} x(t)e^{-st} dt$$

Contoh:

### 1. Eksponensial

$$x(t) = e^{-2t}$$

Perhitungan Analitis:

$$X(s) = \int_0^{\infty} e^{-2t} e^{-st} dt = \int_0^{\infty} e^{-(s+2)t} dt$$

$$X(s) = -\frac{1}{s+2}(e^{-(s+2)t})_0^{\infty} = -\frac{1}{s+2}(0 - 1)$$

$$X(s) = \frac{1}{s+2}$$

Dengan Matlab:

Buat simbolik variabel t s

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

```
syms t s
```

Definisikan fungsi

```
x=exp(-2*t)
```

x =

$$e^{-2t}$$

Transformasi laplace dengan fungsi berikut

```
X=laplace(x,t,s)
```

X =

$$\frac{1}{s+2}$$

2. PD orde 1

$$2 \frac{dy}{dt} + 3y = 10$$

$$y(0) = 10$$

Deklarasikan variabel baru

```
syms t s y(t) Y
```

Definisikan persamaan diferensial yang akan ditransformasikan

```
f=2*(diff(y,t))+3*y-10
```

f(t) =

$$2 \frac{\partial}{\partial t} y(t) + 3y(t) - 10$$

Transformasi laplace F dengan fungsi berikut

```
F=laplace(f,t,s)
```

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://www.instagram.com/@controlsystem.its) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)

F =

$$2s \operatorname{laplace}(y(t), t, s) - 2y(0) - \frac{10}{s} + 3 \operatorname{laplace}(y(t), t, s)$$

Substitusikan  $\operatorname{laplace}(y(t), t, s)$  dengan variabel Y

```
F=subs(F,laplace(y(t), t, s),Y)
```

F =

$$3Y - 2y(0) + 2Ys - \frac{10}{s}$$

Substitusikan kondisi awal  $y(0)=0$

```
F=subs(F,y(0),0)
```

F =

$$3Y + 2Ys - \frac{10}{s}$$

Cari Y yang merupakan hasil transformasi dengan:

```
ys=solve(F,Y)
```

ys =

$$\frac{10}{s(2s+3)}$$

3. PD orde 2

$$\frac{d^2x}{dt^2} + 5\frac{dx}{dt} + 8x = 20$$

$$x(0) = 5 \text{ dan } x'(0) = 3$$

```
syms x(t) t s X
F=diff(diff(x,t),t)+7*diff(x,t)+10*x-20
```

$$F(t) =$$

$$\frac{\partial^2}{\partial t^2}x(t) + 7 \frac{\partial}{\partial t}x(t) + 10x(t) - 20$$

Transformasi laplace F dengan fungsi berikut

```
LF=laplace(F,t,s)
```

$$LF =$$

$$7s \text{ laplace}(x(t), t, s) - 7x(0) - s x(0) + s^2 \text{ laplace}(x(t), t, s) - ((\frac{\partial}{\partial t}x(t))_{t=0}) \\ - \frac{20}{s} + 10 \text{ laplace}(x(t), t, s)$$

Substitusikan laplace(y(t), t, s) dengan variabel X

```
LF=subs(LF,laplace(x(t), t, s),X)
```

$$LF =$$

$$10X - 7x(0) + 7Xs - s x(0) - ((\frac{\partial}{\partial t}x(t))_{t=0}) + X s^2 - \frac{20}{s}$$

Substitusikan kondisi awal x(0)=5

```
LF=subs(LF,x(0),5)
```

$$LF =$$

$$10X - 5s + 7Xs - ((\frac{\partial}{\partial t}x(t))_{t=0}) + X s^2 - \frac{20}{s} - 35$$

Substitusikan kondisi awal x'(0)=3

```
LF=subs(LF,subs(diff(x(t), t), t, 0),3)
```

$$LF =$$

$$10X - 5s + 7Xs + X s^2 - \frac{20}{s} - 38$$



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Cari X yang merupakan hasil transformasi dengan:

```
ys=solve(LF,X)
```

ys =

$$\frac{5s + \frac{20}{s} + 38}{s^2 + 7s + 10}$$

## 4.5. Transformasi Laplace Balik (Inverse)

Transformasi Laplace balik adalah metode integral yang mengubah fungsi domain s ke domain waktu

$$\mathcal{L}^{-1}\{X(s)\} = x(t) = \frac{1}{2\pi} \int_{\sigma-j\omega}^{\sigma+j\omega} X(s)e^{st} ds$$

Mencari Invers dengan metode ini melibatkan integral Kontur dalam bidang kompleks yang relatif sulit, karena itu hanya akan digunakan komputasi dengan Matlab

1.  $X(s) = \frac{1}{s+2}$

Buat simbolik variabel t

```
syms t s
```

Definisikan fungsi

```
X=1/(s+2)
```

X =

$$\frac{1}{s+2}$$

Transformasi laplace dengan fungsi berikut

```
x=ilaplace(X,s,t)
```

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)

$x =$

$$e^{-2t}$$

2.  $Y(s) = \frac{10}{s^2+3s}$

Buat simbolik variabel t

```
syms t s
```

Definisikan fungsi

```
Y=10/(s^2+3*s)
```

$y =$

$$\frac{10}{s^2+3s}$$

Transformasi laplace dengan fungsi berikut

```
y=ilaplace(Y,s,t)
```

$y =$

$$\frac{10}{3} - \frac{10e^{-3t}}{3}$$

3.  $X(s) = \frac{\frac{5s}{s} + 20 + 38}{s^2 + 7s + 10}$

Buat simbolik variabel t

```
syms t s
```

Definisikan fungsi

```
X=(5*s+(20/s)+38)/(s^2+7*s+10)
```

$x =$

$$\frac{\frac{5s}{s} + 20 + 38}{s^2 + 7s + 10}$$



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Transformasi laplace dengan fungsi berikut

```
x=ilaplace(X,s,t)
```

X =

$$6e^{-2t} - 3e^{-5t} + 2$$

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS

Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://www.instagram.com/@controlsystem.its) [linkedin.com/company/its-control-system/](https://linkedin.com/company/its-control-system/)



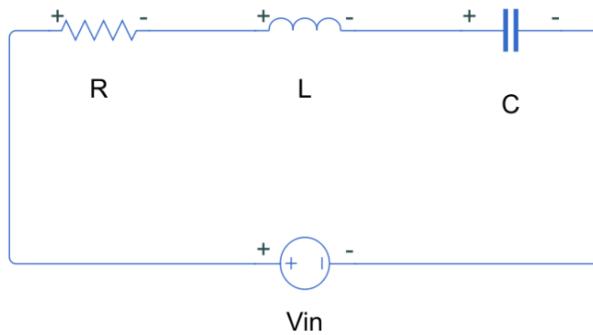
## BAB 5. Pemodelan Sistem Elektrik

### 5.1. Pendahuluan

Dalam melakukan analisis dan perancangan sistem pengaturan, pemodelan sistem adalah hal yang harus dilakukan. Analisis pada sistem pengaturan dilakukan untuk mengetahui performa dari sistem yang dimodelkan. Perancangan pada sistem pengaturan adalah proses merancang suatu sistem, berdasarkan model yang sudah ada, agar memiliki respon output yang perancang inginkan. Beberapa model matematika yang umum digunakan antara lain: model persamaan diferensial, model fungsi alih, dan model state space. Pada bab ini dibahas pemodelan sistem menggunakan persamaan diferensial yang akan diselesaikan dengan bantuan MATLAB. Fungsi MATLAB yang akan dimanfaatkan adalah *dsolve* dan *laplace* serta *ilaplace*.

### 5.2. Sistem Elektrik

Sistem elektrik yang digunakan pada proses pemodelan ini adalah suatu persamaan pada sistem rangkaian listrik yang terdiri dari komponen RLC yang disusun seri. Input berupa tegangan  $V_{in}$  yang diasumsikan dimulai pada  $t=0$ . Output adalah  $V_c$  yaitu tegangan pada kapasitor.



Diketahui bahwa,

$$v_R = i R; v_L = L \frac{di}{dt}; i = C \frac{dv_C}{dt}$$

Selanjutnya, dengan menggunakan *Kirchoff Voltage Law*, didapatkan

$$v_L + v_R + v_C = v_{in}$$

$$L \frac{di}{dt} + R i + v_C = V_{in}$$



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Substitusikan rumus  $i = C \frac{d v_c}{dt}$ :

$$LC \frac{d^2 v_c}{dt^2} + RC \frac{d v_c}{dt} + v_c = v_{in}$$

Mendefinisikan nilai-nilai parameter, kondisi awal, dan input

```
syms vc(t) Vc t s  
R = 1; % nilai R  
L = 1; % nilai L  
C = 1; % nilai C  
Vin = 10; % nilai V input  
cond1 = vc(0) == 0;% kondisi awal vc saat t = 0  
Dvc = diff(vc);  
cond2 = Dvc(0) == 0; % kondisi awal saat dvc/dt saat t = 0  
conds = [cond1 cond2];
```

Mendefinisikan persamaan diferensial yang sudah didapat

```
ODE = L*C*diff(vc,t,2)+R*C*diff(vc,t)+vc == Vin
```

ODE(t) =

$$\frac{\partial^2}{\partial t^2}vc(t) + \frac{\partial}{\partial t}vc(t) + vc(t) = 10$$

Untuk mendapatkan  $v_c(t)$  dapat dicari dengan dua metode yaitu menggunakan *dsolve* dan menggunakan *laplace*.

1. Fungsi *dsolve*

Menyelesaikan ODE dengan kondisi awal dengan fungsi berikut

```
vcSol(t)=dsolve(ODE,conds)
```

vcSol(t) =

$$10 - \frac{10\sqrt{3} e^{-t/2} \sin(\frac{\sqrt{3}t}{2})}{3} - 10 e^{-t/2} \cos(\frac{\sqrt{3}t}{2})$$

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://twitter.com/@controlsystem.its) [linkedin.com/company/its-control-system/](https://linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

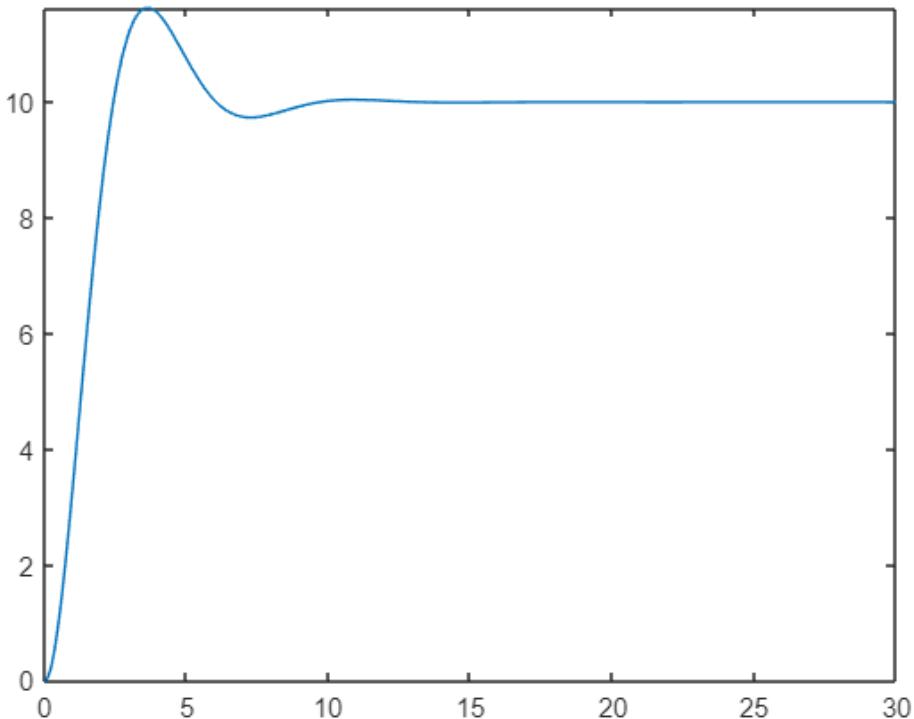
Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Plot fungsi solusi  $vc(t)$  dengan range [0 30]

```
fplot(vcSol, [0 30])
```



## 2. Laplace

Transformasi laplace ODE dengan fungsi berikut

```
LF=laplace(ODE,t,s)
```

LF =

$$s \text{ laplace}(vc(t), t, s) - vc(0) - s vc(0) + s^2 \text{ laplace}(vc(t), t, s) - \left( \left( \frac{\partial}{\partial t} vc(t) \right)_{t=0} \right) \\ + \text{ laplace}(vc(t), t, s) = \frac{10}{s}$$

Substitusikan  $\text{laplace}(vc(t), t, s)$  dengan variabel Vc

```
LF=subs(LF,laplace(vc(t),t,s),Vc)
```

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

LF =

$$Vc - vc(0) + Vc s - s vc(0) - \left( \left( \frac{\partial}{\partial t} \right)_{t=0} vc(t) \right) + Vc s^2 = \frac{10}{s}$$

Substitusikan kondisi awal  $vc(0)=0$

```
LF=subs(LF, vc(0), 0)
```

LF =

$$Vc + Vc s - \left( \left( \frac{\partial}{\partial t} \right)_{t=0} vc(t) \right) + Vc s^2 = \frac{10}{s}$$

Substitusikan kondisi awal  $vc'(0)=0$

```
LF=subs(LF,subs(diff(vc(t),t),t,0),0)
```

LF =

$$Vc s^2 + Vc s + Vc = \frac{10}{s}$$

Cari  $Vcs$  yang merupakan hasil transformasi laplace dengan fungsi berikut

```
Vcs=solve(LF,Vc)
```

$Vcs =$

$$\frac{10}{s^3 + s^2 + s}$$

Cari invers transformasi laplace  $I_s$  dengan fungsi berikut

```
vcInv=ilaplace(Vcs,s,t)
```

$vcInv =$

$$10 - 10 e^{-t/2} \left( \cos\left(\frac{\sqrt{3}}{2}t\right) + \frac{\sqrt{3} \sin\left(\frac{\sqrt{3}}{2}t\right)}{3} \right)$$

Plot fungsi  $vcInv(t)$  dengan range [0 30]

```
fplot(vcInv,[0 30])
```

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://@controlsystem.its) [linkedin.com/company/its-control-system/](https://linkedin.com/company/its-control-system/)

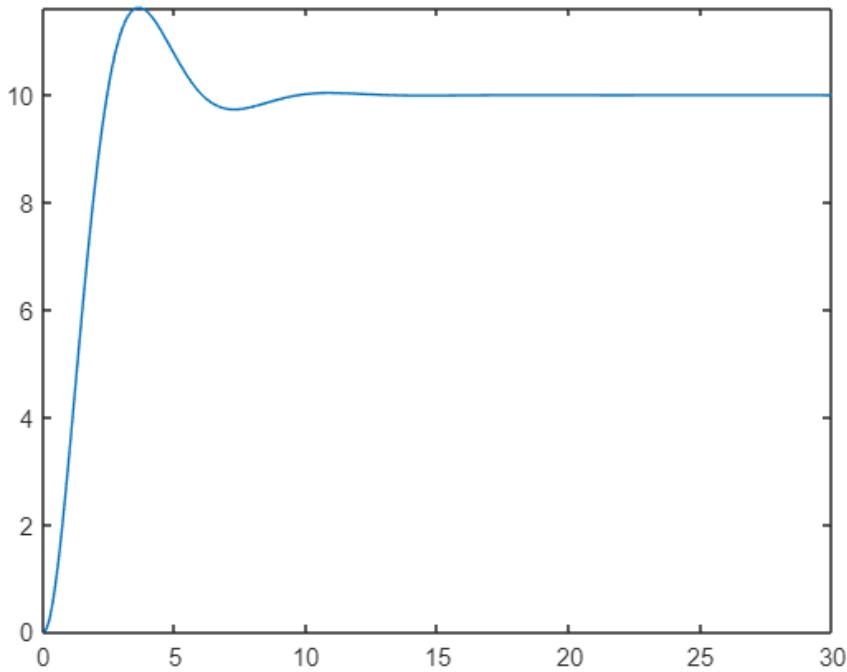


# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember



## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS

Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://www.instagram.com/@controlsystem.its) [linkedin.com/company/its-control-system/](https://linkedin.com/company/its-control-system/)

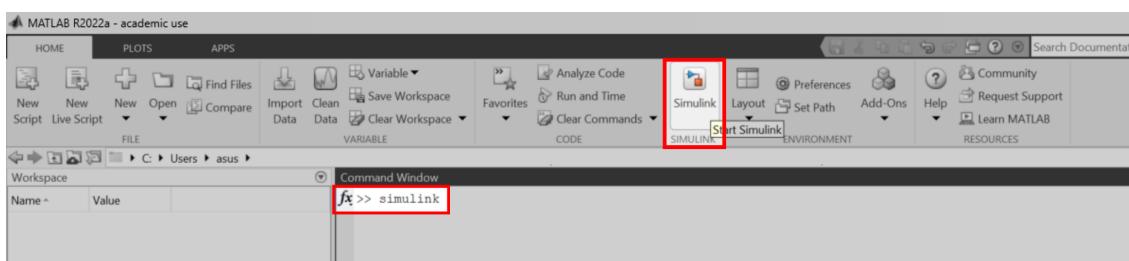


## BAB 6. Pengenalan SIMULINK

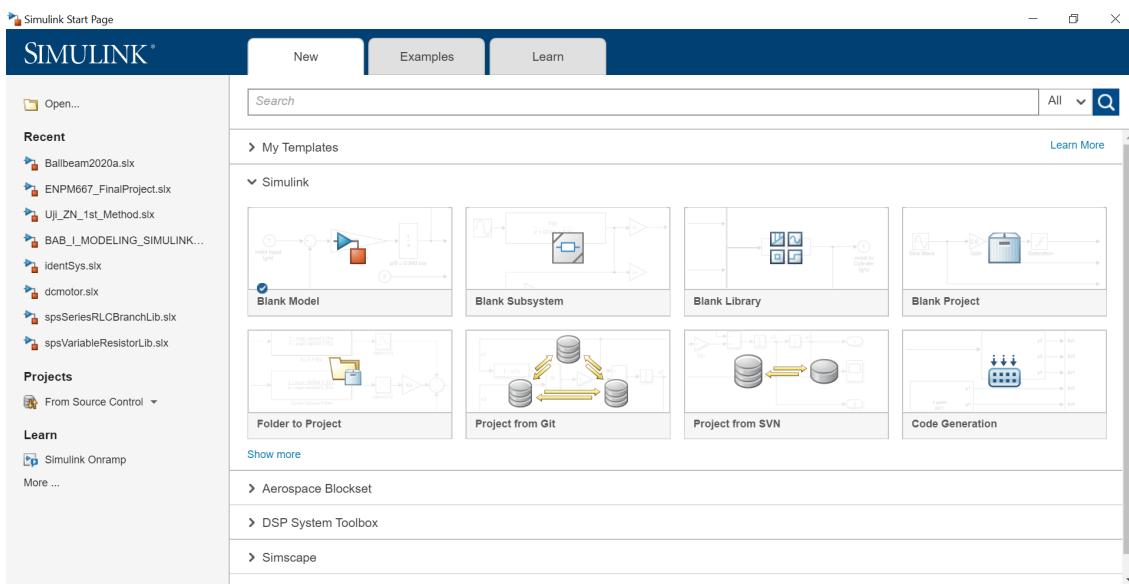
### 6.1. Pendahuluan

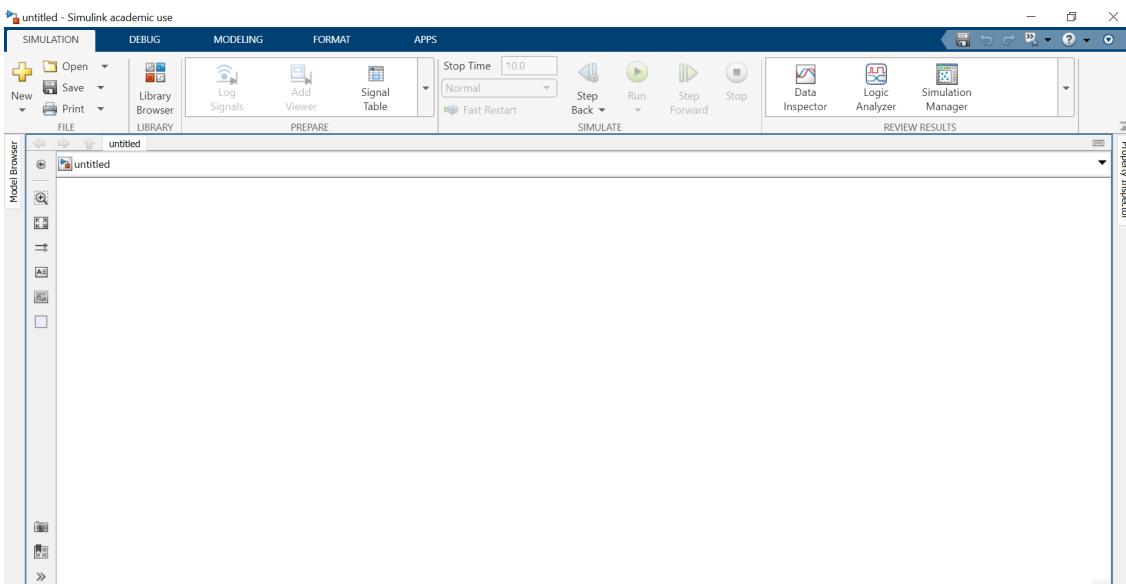
SIMULINK adalah lingkungan pemrograman grafis berbasis MATLAB. Umumnya digunakan untuk mendesain sistem multi domain dan melakukan simulasi tanpa perlu menulis kode. Tampilan utama SIMULINK berisi blok-blok grafis yang dapat dimodifikasi sedemikian rupa sesuai kebutuhan. Pada bagian ini, akan dipelajari mengenai fundamental dari SIMULINK diantaranya lingkungan grafis, cara melihat sinyal, algoritma dasar, serta pembuatan sistem kontinu dan diskrit.

### 6.2. Lingkungan Grafis

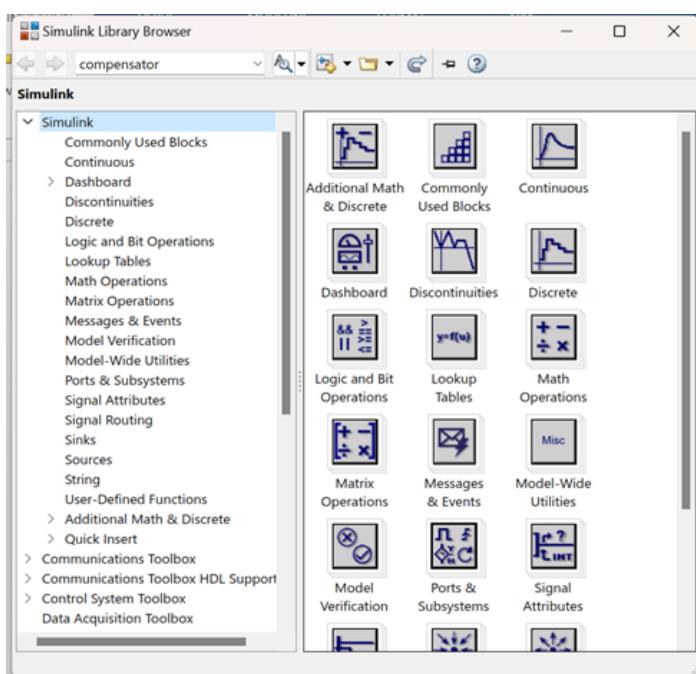
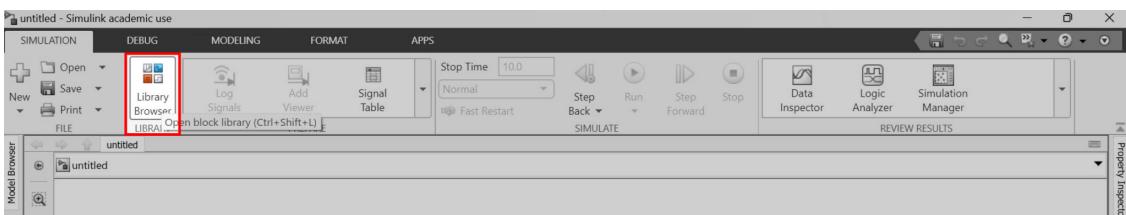


Langkah paling awal dari mempelajari SIMULINK adalah mengetahui cara membuka SIMULINK itu sendiri. Terdapat dua alternatif cara : 1. Dengan melakukan klik pada ikon Simulink 2. Dengan mengetikkan ‘simulink’ pada *command window*. Setelah melakukan langkah sebelumnya, maka akan diarahkan pada ‘SIMULINK Start Page’, selanjutnya klik ‘Blank Model’ untuk memulai SIMULINK dengan kanvas kosong.





Pada SIMULINK dikenal istilah ‘block’ yaitu elemen utama yang digunakan untuk membangun model. Kumpulan *block* ini dapat ditemukan pada ‘Library Browser’. Membuka *Library Browser* dilakukan dengan klik pada *icon* di *toolbar*.



## **Laboratorium Kontrol dan Otomasi**

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



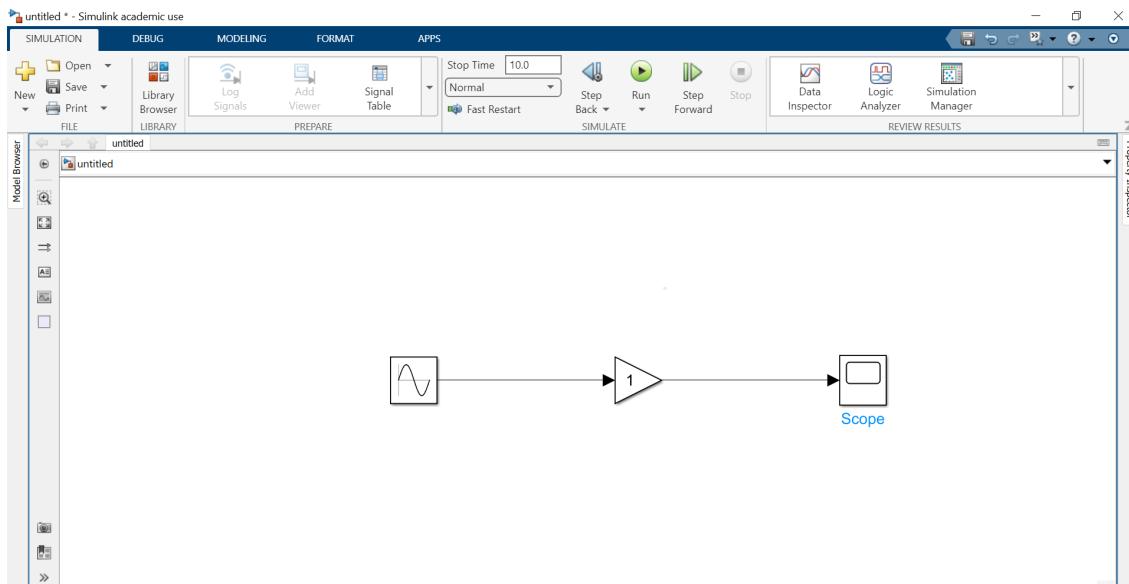
# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

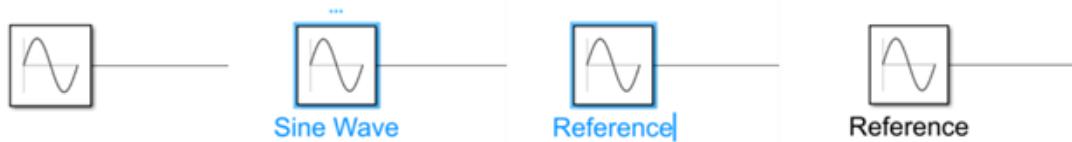
Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Berikutnya klik pada elemen blok yang dibutuhkan, maka blok tersebut akan muncul di kanvas SIMULINK.



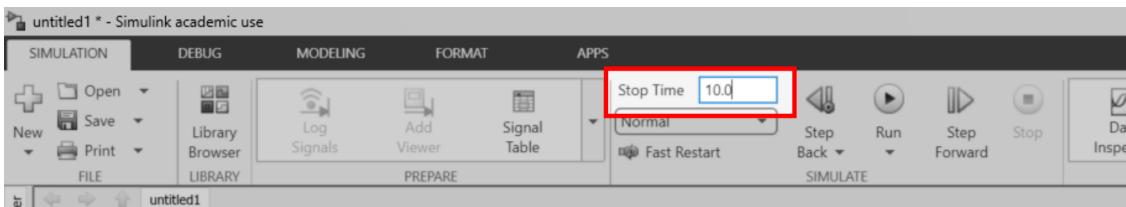
Selain melalui *Library Browser*, memanggil blok juga bisa dilakukan dengan cara *double-click* pada kanvas kosong kemudian menuliskan nama blok yang dimaksud. Blok-blok yang telah dimasukkan pada kanvas SIMULINK dapat dirubah namanya sesuai dengan yang diinginkan. Pertama lakukan klik pada blok yang inginkan, maka akan muncul nama blok berwarna biru. Kedua lakukan klik pada nama blok, maka akan membuka opsi untuk menulis ulang nama blok. Ketiga ubah nama sesuai dengan yang diinginkan kemudian klik ‘enter’ atau klik pada area kosong pada kanvas. Cara ini juga berlaku untuk elemen lain, misalnya untuk kabel penghubung.



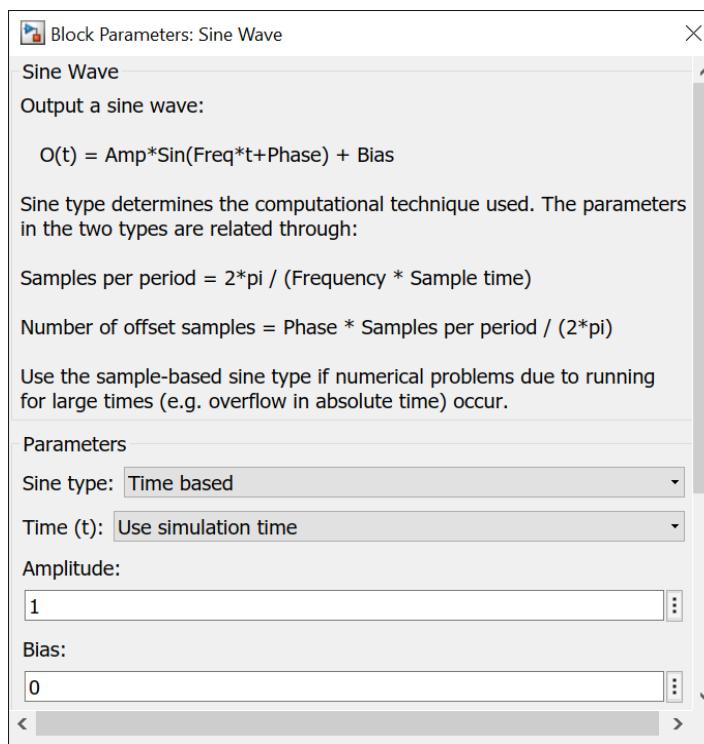
Sama seperti halnya MATLAB, SIMULINK juga menggunakan fungsi *run* untuk menjalankan simulasi, yang menjadi pembeda utama adalah pada SIMULINK terdapat waktu simulasi, pada *toolbar* diberi nama ‘*stop time*’. Seperti Namanya, *stop time* berfungsi untuk menentukan seberapa lama simulasi akan berlangsung. Nilai *default* dari *stop time* ini adalah 10 detik. Variasi nilai *stop time* dapat dilakukan langsung dengan mengganti nilainya sesuai dengan yang diinginkan.

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



Mayoritas elemen-elemen blok pada SIMULINK memiliki nilai parameter didalamnya yang dapat divariasikan sesuai dengan kebutuhan. Untuk melihat nilai parameter ini dapat dilakukan dengan melakukan *double-click* pada blok yang dituju. Sementara untuk merubah nilainya dapat langsung dilakukan dengan diketik ulang.



Nilai parameter tidak terbatas hanya angka saja, dapat juga memasukkan nilai parameter dalam bentuk variabel. Misalnya digunakan nilai r untuk amplitudo sinyal sinus

$r=2.5$

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

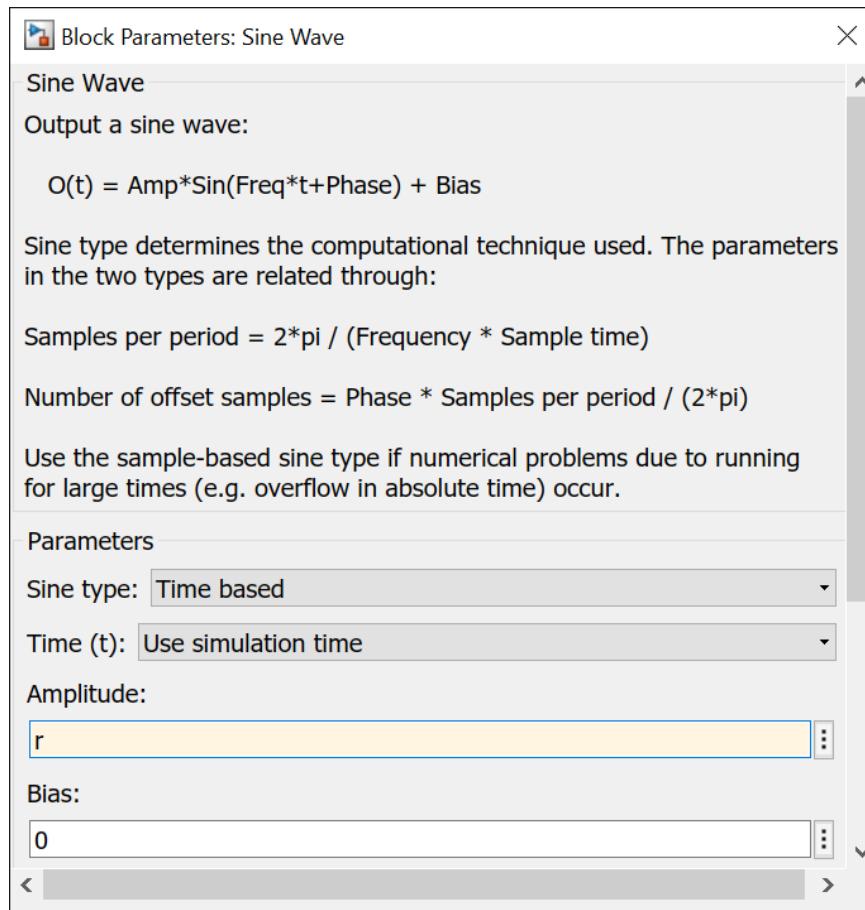


# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember



Hal yang harus diingat ketika menggunakan nilai variabel sebagai parameter adalah harus memasukkan nilai variabel tersebut kedalam *workspace* MATLAB terlebih dahulu sebelum melakukan *run* pada SIMULINK. Peringatan merah pada Gambar 10 muncul karena variabel ‘r’ belum ada pada *workspace*.

### 6.3. Melihat Sinyal

Setelah mengetahui cara untuk menambah blok dan merubah parameter, berikutnya adalah cara melihat hasil dari simulasi. Blok yang paling sering digunakan untuk melihat *output* dari simulasi adalah blok ‘scope’. Blok ini memungkinkan untuk melihat sinyal keluaran sekaligus melakukan inspeksi tentang nilai-nilai keluaran per waktu. Dimisalkan sebuah sinyal sinus dengan amplitudo 2.5 dan diberi *gain* 2, maka hasil yang terlihat pada scope adalah sebagai berikut

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

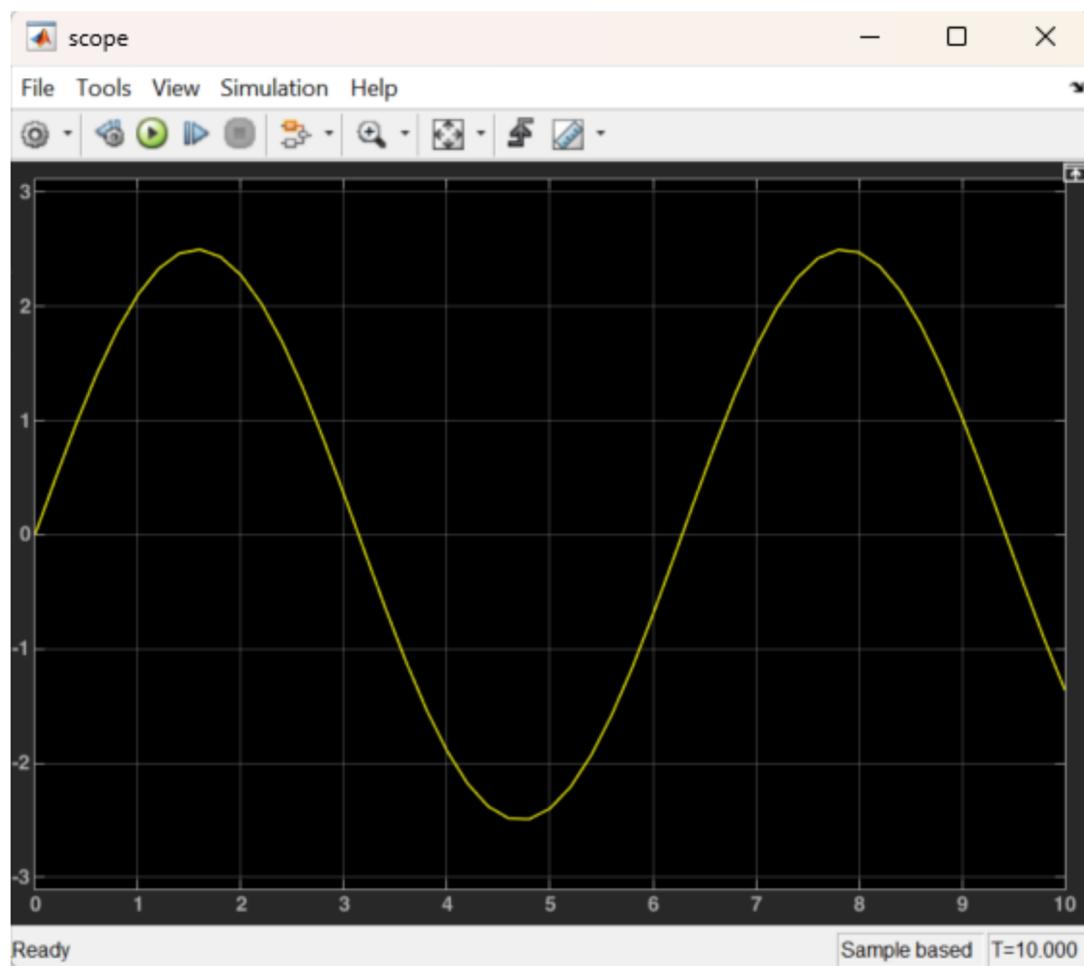
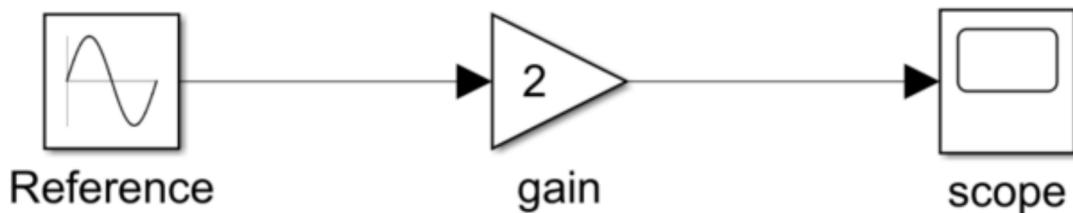


# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember



Blok scope memiliki beberapa ikon yang cukup sering digunakan pada toolbarnya sebagai berikut



## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



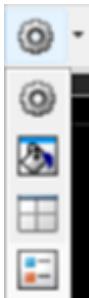
# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

## 1. Configuration Properties



Menu ini digunakan untuk memodifikasi *setting scope* dan visual dari sinyal. *Configuration properties* dapat digunakan untuk menambah jumlah *port*, mengatur *sample time*, dan membatasi data. *Style* digunakan untuk memodifikasi visual sinyal, mulai dari warna, ketebalan, hingga menambah *marker*. *Layout* digunakan untuk mengatur tata letak sinyal ketika diberi 2 *input* atau lebih. Terakhir, *legend*, digunakan untuk menampilkan nama/keterangan pada grafik yang muncul.

## 2. Zoom



*Zoom* digunakan untuk memperbesar/memperkecil bagian tertentu dari sinyal. Terdapat pula fitur *pan* untuk menggeser sinyal.

## 3. Scale Limits



*Scale limits* digunakan untuk menampilkan sinyal secara utuh pada batasan sumbu tertentu. Misalkan *Scale X & Y Axes Limits* menampilkan sinyal secara utuh pada seluruh batasan nilai sumbu x dan sumbu y.

Fitur lain yang sering digunakan juga terdapat pada **Tools > Measurements**. Terdapat berbagai macam opsi pengukuran, misalnya *Cursor Measurement* dapat mengukur nilai pada waktu tertentu (keakuratan 3 angka dibelakang koma).

# Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://@controlsystem.its) [linkedin.com/company/its-control-system/](https://linkedin.com/company/its-control-system/)

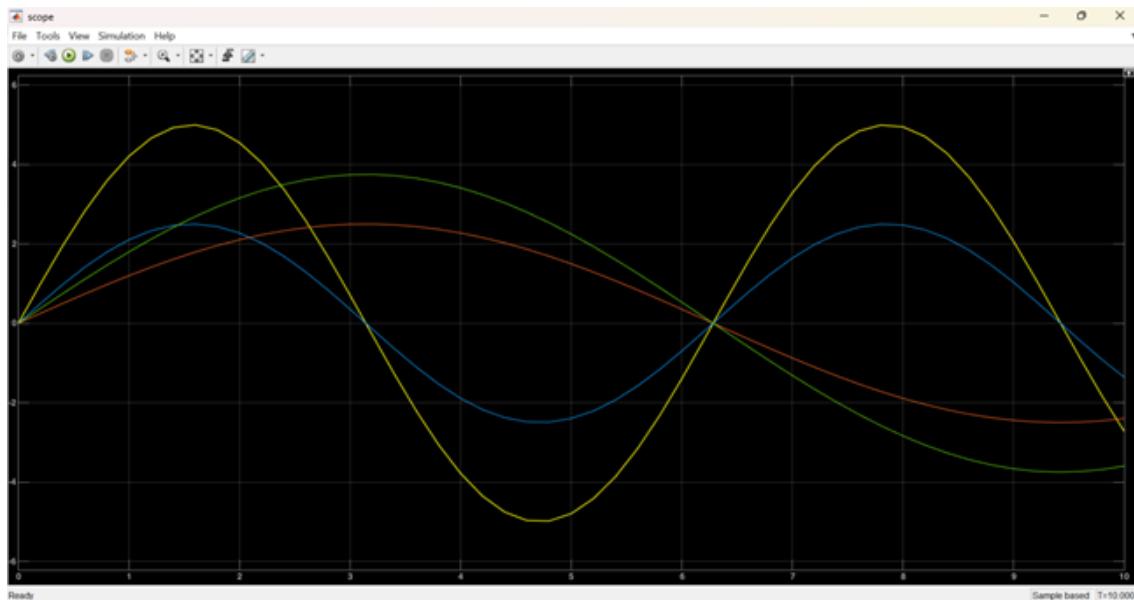


# Laboratorium Kontrol dan Otomasi

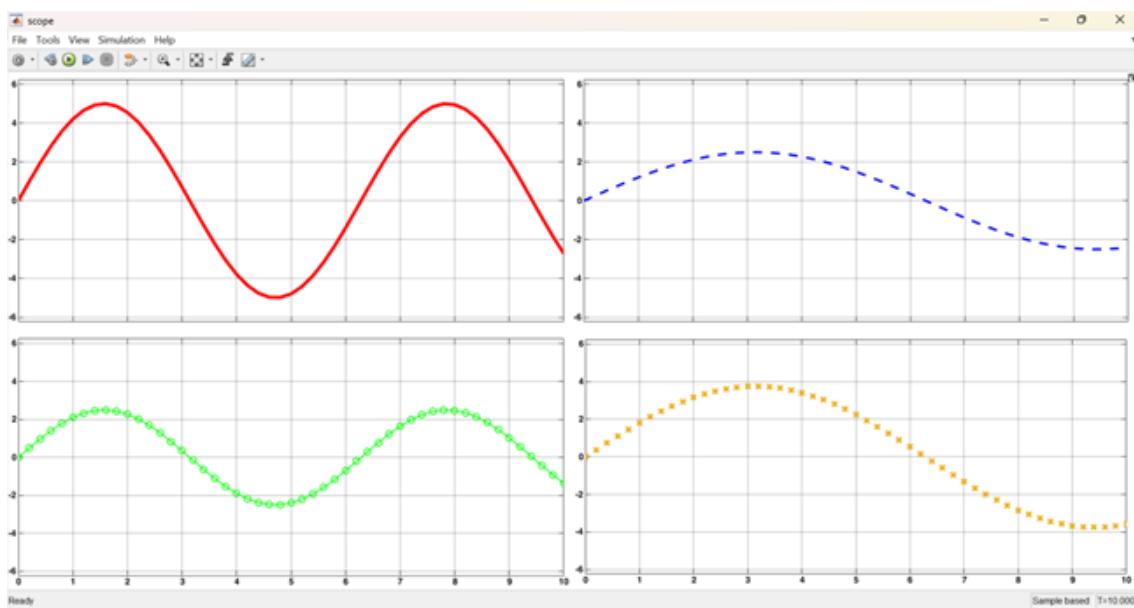
Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember



Penggunaan Layout dan Style



Penggunaan cursor measurement

## Laboratorium Kontrol dan Otomasi

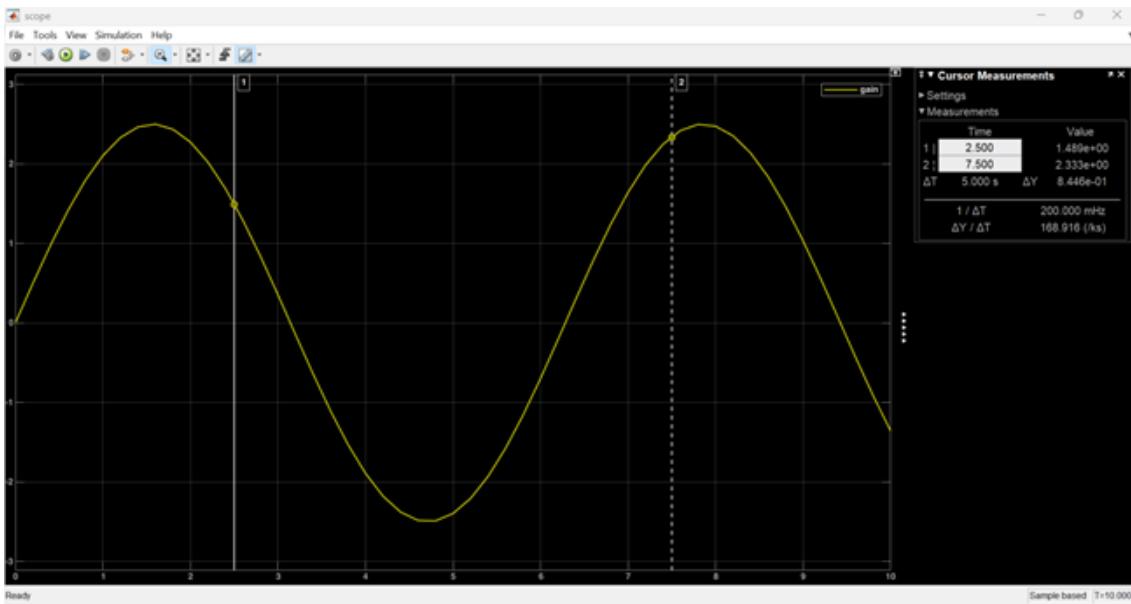
Ruang AJ104 & B105, Departemen Teknik Elektro ITS

Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id)

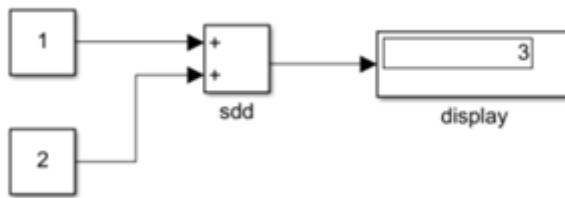
[@controlsystem.its](http://@controlsystem.its)

[linkedin.com/company/its-control-system/](http://linkedin.com/company/its-control-system/)

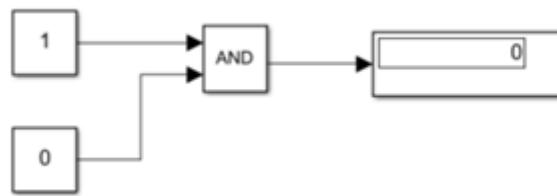


## 6.4. Algoritma Dasar

SIMULINK juga memungkinkan untuk melakukan simulasi operasi matematika, gerbang logika, maupun pernyataan bersyarat. Misalnya untuk operasi matematika, gunakan blok ‘constant’, kemudian gunakan blok ‘add’, blok ini berfungsi untuk menjumlahkan sinyal. Apabila melihat parameter pada blok, maka dapat juga difungsikan pengurangan dengan merubah tanda operasi dalam parameternya. Terakhir gunakan blok ‘display’ untuk menampilkan hasil perhitungan.



Jenis operasi matematika lain dapat ditemui pada **Library Browser > Math Operations**. Selanjutnya untuk membuat gerbang logika dapat menggunakan blok pada **Library Browser > Logic and Bit Operations**. Sebagai contoh kita gunakan logika AND seperti berikut





# Laboratorium Kontrol dan Otomasi

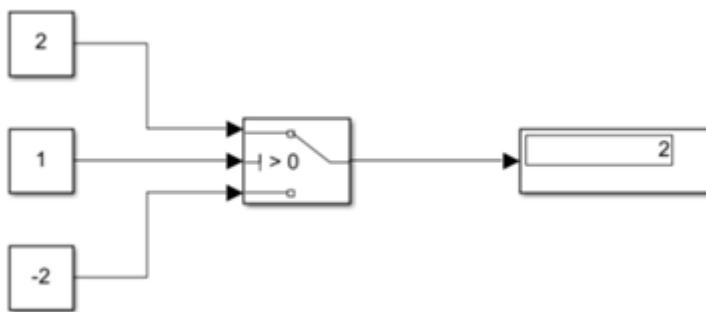
Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Terakhir untuk pernyataan bersyarat, pada SIMULINK dapat dibuat rangkaian yang memiliki arti yang sama dengan *pseudocode* tertentu. Misalnya terdapat kode berikut

```
if (constant > 0)
    output 2
else
    output -2
end
```



## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS

Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

## LAMPIRAN

```
% EULER METHOD TABLE COMPARISON

clear;clc

dy = @(x,y)x+y;           %Persamaan Diferensial
f = @(x)-x+exp(x)-1;      %solusi analitis dari PD

x0 = 0;        %interval awal
xn = 1;        %interval akhir
y = 0;         %initial condition y(0)
h = 0.2;       %step size

fprintf ('x \t\t\t y (euler) \t y (analitis)\t error \n')    %print
header tabel data

fprintf('%f \t %f \t %f\t\t %f\n',x0,y,f(x0),abs(y-f(x0)));   %print
baris pertama tabel yang merupakan kondisi awal

for x = x0 : h : xn-h      %menghitung solusi dari x0 sampai xn
dengan step size h
    y = y + dy(x,y)*h;      %formula metode euler
    x = x + h;              %increment x dengan step size h
    fprintf('%f \t %f \t %f\t\t %f\n',x,y,f(x),abs(y-f(x)));   %print
solusi tiap baris dan errornya

End
```

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

```
% RK 4 METHOD TABLE COMPARISON
```

```
clear;clc
```

```
dy = @(x,y)x+y; %Persamaan Diferensial
```

```
f = @(x)-x+exp(x)-1; %solusi analitis dari PD
```

```
x0 = 0; %interval awal
```

```
xn = 1; %interval akhir
```

```
y = 0; %initial condition y(0)
```

```
h = 0.2; %step size
```

```
fprintf ('x \t\t\ty (RK4) \t y (analitis)\t error \n') %print header tabel data
```

```
fprintf('%f \t %f \t %f\t\t %f\n',x0,y,f(x0),abs(y-f(x0))); %print baris pertama tabel yang merupakan kondisi awal
```

```
for x = x0 : h : xn-h %menghitung solusi dari x0 sampai xn dengan step size h
```

```
%formula RK orde 4
```

```
k1 = h*dy(x,y);
```

```
k2 = h*dy(x+0.5*h,y+0.5*k1);
```

```
k3 = h*dy(x+0.5*h,y+0.5*k2);
```

```
k4 = h*dy(x+h,y+k3);
```

```
y = y + (1/6)*(k1 + 2*k2 + 2*k3 + k4);
```

```
x = x + h; %increment x dengan step size h
```

```
fprintf('%f \t %f \t %f\t\t %f\n',x,y,f(x),abs(y-f(x))); %print solusi tiap baris dan errornya
```

```
end
```

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [@controlsystem.its](https://www.instagram.com/@controlsystem.its) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

```
% ODE23 TABLE COMPARISON
```

```
clear;clc

dy = @(x,y)(x+y); %Persamaan Diferensial

f = @(x)-x+exp(x)-1; %solusi analitis dari PD

x0 = 0; %interval awal

xn = 1; %interval akhir

y0 = 0; %initial condition y(0)

h = 0.2; %step size

fprintf ('x \t\t\ty (ode23) \t y (analitis)\t error \n');

[x,y] = ode23(dy,[x0:h:xn],y0);

for i = 1 : length(x0:h:xn)

    fprintf('%f \t %f \t %f\t\t
%f\n',x(i),y(i),f(x(i)),abs(y(i)-f(x(i)))); %print solusi tiap baris
% dan errornya

end
```

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)



# Laboratorium Kontrol dan Otomasi

Departemen Teknik Elektro ITS

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

```
% ODE45 TABLE COMPARISON
```

```
clear;clc

dy = @(x,y)(x+y); %Persamaan Diferensial

f = @(x)-x+exp(x)-1; %solusi analitis dari PD

x0 = 0; %interval awal

xn = 1; %interval akhir

y0 = 0; %initial condition y(0)

h = 0.2; %step size

fprintf ('x \t\t\ty (ode23) \t y (analitis)\t error \n');

%print header tabel data

[x,y] = ode45(dy,[x0:h:xn],y0);

for i = 1 : length(x0:h:xn)

    fprintf('%f \t %f \t %f\t\t
%f\n',x(i),y(i),f(x(i)),abs(y(i)-f(x(i)))); %print solusi tiap baris
% dan errornya

end
```

## Laboratorium Kontrol dan Otomasi

Ruang AJ104 & B105, Departemen Teknik Elektro ITS  
Keputih, Sukolilo, Surabaya

[cse.ee.its.ac.id](http://cse.ee.its.ac.id) [linkedin.com/company/its-control-system/](https://www.linkedin.com/company/its-control-system/)