

LAPORAN PRAKTIKUM MATA KULIAH

METODE NUMERIK

PRAKTIKUM 3 – PENGENALAN MATLAB/OCTAVE



DISUSUN OLEH:

M0521003 – ADI PRASETYA

PROGRAM STUDI INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS SEBELAS MARET

2022

BAB I

ANALISIS SOURCE CODE

1. Analisis Source Code “Looping”

```
1 % M0521003 - Adi Prasetya
2 for i = 1:7
3     p = i^3
4 end
```

Dalam source code tersebut, terdapat 1 looping yang menggunakan for loop dan 2 variabel, yaitu **i** dan **p**. Perulangan akan dilakukan dari **i = 1** hingga **i = 7** sehingga dituliskan dengan **for i = 1 : 7**. Kemudian, pada setiap perulangan nilai dari variabel **i** akan dieksposen 3 yang kemudian nilainya disimpan dan dicetak oleh variabel **p**.

```
1 % M0521003 - Adi Prasetya
2 for j = 1:0.5:5
3     q = j/2
4 end
```

Dalam source code tersebut terdapat 1 looping yang menggunakan for loop dan 2 variabel, yaitu **j** dan **q**. Pengulangan dilakukan dari **j = 1** dan setiap perulangan maka nilai dari **j** akan ditambah **0.5** hingga nilai akhir **j** menjadi **5**. Kemudian, pada setiap perulangan nilai dari variabel **j** akan dibagi **2** yang kemudian nilainya disimpan dan dicetak oleh variabel **q**.

```
1 % M0521003 - Adi Prasetya
2 p = 0;
3 while(p <= 10)
4     q = p^2 + p
5     p = p + 1;
6 end
```

Dalam source code tersebut terdapat 1 looping yang menggunakan while loop dan 2 variabel, yaitu **p** dan **q**. Awalnya variabel **p** dideklarasikan dengan **p = 0**, lalu terdapat while loop dengan kondisi jika **p <= 10** maka perulangan akan terus dilakukan. Kemudian, nilai dari variabel **p** dikuadratkan dan ditambah dengan nilai **p** sebelum dikuadratkan lalu disimpan dan dicetak dalam variabel **q**. Setelah itu, nilai dari **p** akan diincrement dengan menambahkan **1** setiap terjadi perulangan hingga **p = 10**.

2. Analisis Source Code “Differensial”

```
1 % M0521003 - Adi Prasetya
2 f = input('Masukkan bentuk persamaan f(x) = ');
3 f_real = sym (f)
4 f_differential = diff(f_real, 'x')
```

Dalam source code tersebut pada **Line-2** dimana user akan diminta memasukkan input berupa persamaan $f(x)$. Pada **line 3** terdapat nama objek berupa **f_real** yang di dalamnya terdapat fungsi **sym** dan memiliki value **f**. Lalu, pada **line 4** terdapat name objek berupa **f_differential** yang di dalamnya terdapat fungsi **diff** untuk menurunkan persamaan dari **f_real**.

3. Analisis Source Code “Integral”

```
1 % M0521003 - Adi Prasetya
2 f = input('Fungsi = ')
3 f_real = sym (f)
4 f_integral = int(f_real, 'x')
```

Dalam source code tersebut, pada **line 2** dimana user akan diminta memasukkan input berupa persamaan $f(x)$. Pada **line 3** terdapat nama objek berupa **f_real** yang di dalamnya terdapat fungsi **sym** dan memiliki value **f**. Lalu, pada **line 4** terdapat name objek berupa **f_integral** yang di dalamnya terdapat fungsi **int** untuk mengintegalkan persamaan dari **f_real**.

4. Analisis Source Code “Function”

```
1 % M0521003 - Adi Prasetya
2 function myNameIs(name)
3     printf ('My name is %s\n', name);
4 endfunction
5
```

Dalam source code tersebut, pada **line 2** terdapat fungsi yang bernama **myNameIs** dan memiliki parameter **name**. Kemudian pada **line 3** terdapat fungsi yang berfungsi untuk mencetak pernyataan dan di dalamnya terdapat **name** yang akan mencetak sesuai dengan masukan dari user.

```

1 % M0521003 - Adi Prasetya
2 function Activity(name)
3     Doing(name)
4     fprintf('His/her name is %s!\n', name)
5 endfunction
6
7 function Doing(name)
8     fprintf('%s is eating\n',name)
9 endfunction

```

Dalam source code tersebut, pada **line 2** terdapat fungsi yang bernama **Activity** yang bersifat global dan memiliki parameter **name**. Kemudian, pada **line 3** terdapat pemanggilan fungsi **Doing** yang bersifat lokal. Kemudian pada **line 4** terdapat fungsi yang berfungsi untuk mencetak pernyataannya dan di dalamnya terdapat **name** yang akan mencetak sesuai masukkan dari user. Pada **line 7** terdapat fungsi yang bernama **Doing** dan memiliki parameter **name**. Fungsi ini bersifat lokal, lalu pada **line 8** terdapat *fprintf* yang di dalamnya terdapat parameter **name**.

```

1 % M0521003 - Adi Prasetya
2 function hasil = nilai(value)
3     hasil = value/2
4 endfunction

```

Dalam source code tersebut, pada **line 2** terdapat fungsi yang bernama **hasil** yang kemudian diinisialisasikan dengan fungsi **nilai** yang memiliki parameter berupa **value**. Pada **line 3**, nilai dari **value** dibagi **2** yang kemudian hasilnya disimpan dan dicetak oleh variabel **hasil**.

```

1 % M0521003 - Adi Prasetya
2 function [div2, div5,div10] = devide(value1, value2, value3)
3     div2 = value1/2;
4     div5 = value2/5;
5     div10 = value3/10;
6 endfunction

```

Dalam source code tersebut, pada **line 2** terdapat fungsi dimana berisikan tiga variabel yang dapat diisi oleh user. Kemudian, terdapat fungsi **devide** yang memiliki tiga parameter. Pada **line 3** hingga **line 5** terdapat tiga variabel yang berbeda dimana masing-masing variabel memiliki operasi yang berbeda juga.

5. Analisis Source Code “Grafik”

```
1 % M0521003 - Adi Prasetya
2 x = 1:25:100;
3 y = 2*x.^3 + 4*x.^2-80*x;
4 x,y
5 plot(x,y)
```

Dalam source code tersebut, pada **line 2** terdapat variabel **x** yang memiliki nilai dari **1** hingga **100** dimana jarak untuk setiap **x** nya adalah **25**. Kemudian, pada **line 3** terdapat variabel **y** yang memiliki nilai berupa persamaan dengan eksponen 3. Pada **line 4**, terdapat variabel **x,y** dan pada **line 5** terdapat fungsi **plot** yang di dalamnya terdapat pendeklarasian **x** dan **y**. Fungsi **plot** yang berfungsi untuk merepresentasikan data dalam bentuk grafik sehingga lebih mudah untuk dilihat secara keseluruhan.

```
1 % M0521003 - Adi Prasetya
2 x = 1:1:100;
3 y = 2*x.^3 + 4*x.^2-80*x;
4 x,y
5 plot(x,y)
```

Dalam source code tersebut, pada **line 2** terdapat variabel **x** yang memiliki nilai dari **1** hingga **100** dimana jarak untuk setiap **x** yang berdekatan adalah **1**. Kemudian, pada **line 3** terdapat variabel **y** yang memiliki nilai berupa persamaan dengan eksponen 3. Pada **line 4** terdapat pendeklarasian **x** dan **y**. Pada **line 5**, terdapat fungsi **plot** yang berfungsi untuk merepresentasikan data dalam bentuk grafik sehingga lebih mudah untuk dilihat secara keseluruhan.

```
1 % M0521003 - Adi Prasetya
2 x = linspace(0,25)
3 y = exp(-2*x/5).*sin(2*x);
4 plot(x,y)
5 xlabel('Sumbu X');
6 ylabel('Sumbu Y');
7 title('Gambarkan grafik persamaan f(x) = exp(-2x/5).*sin(2x)');
```

Dalam source code tersebut, pada **line 2** terdapat variabel **x** yang di dalamnya terdapat fungsi **linspace** yang berfungsi untuk membuat vektor baris berisi **n** titik yang terpisah merata secara linear antara **0** dan **25**.. Pada **line 3**, terdapat variabel **y** yang memiliki nilai berupa fungsi persamaan. Kemudian pada **line 4** terdapat fungsi **plot** yang berfungsi untuk merepresentasikan data dalam bentuk grafik sehingga lebih mudah untuk dilihat secara keseluruhan.. Lalu, pada **line 5** dan **6** terdapat fungsi **xlabel**

dan **ylabel** yang berfungsi untuk memberi keterangan berupa label pada **sumbu x** dan **sumbu y**. Pada **line 7**, terdapat fungsi **title** yang berfungsi untuk memberi keterangan berupa judul pada grafik yang akan dibuat.

```
1 % M0521003 - Adi Prasetya
2 x = 0: 0.02 : 2*pi;
3 y = -5*cos(5*x) - 8*sin(3*x);
4 z = 8*sin(2*x) .* -6.*cos(10*x);
5 plot(x,y,x,z);
```

Dalam source code tersebut, pada **line 2** terdapat variabel **x** yang memiliki nilai dari **0** hingga **2phi** dimana jarak untuk **x** yang berdekatan adalah **0.02**. Pada **line 3** terdapat variabel **y** yang memiliki nilai berupa fungsi persamaan. Pada **line 4** terdapat variabel **z** yang memiliki nilai berupa fungsi persamaan juga. Pada **line 5**, terdapat fungsi **plot** berfungsi untuk merepresentasikan data dalam bentuk grafik sehingga lebih mudah untuk dilihat secara keseluruhan.

BAB II

ANALISIS PRAKTIKUM

1. Analisis “Looping”

```
Command Window
>> looping1

p = 1
p = 8
p = 27
p = 64
p = 125
p = 216
p = 343
>> |
```

Ketika source code dijalankan, looping akan langsung dilakukan dimana looping akan dilakukan sebanyak **7 kali**. Looping dimulai Ketika **i = 1** hingga **i = 7**, lalu pada setiap perulangan nilai **i** akan dieksposen **3** kemudian dimasukkan ke dalam variabel **p** dan dicetak ke layar. Proses ini terus berlangsung hingga **i = 7**.

Command Window

```
>> looping2
```

```
q = 0.5000  
q = 0.7500  
q = 1  
q = 1.2500  
q = 1.5000  
q = 1.7500  
q = 2  
q = 2.2500  
q = 2.5000  
>> |
```

Ketika source code dijalankan, looping akan langsung dilakukan dimana looping akan dilakukan sebanyak **10 kali**. Looping dimulai ketika **j = 1** hingga **j = 5** dimana setiap perulangannya nilai **j** akan ditambah **0.5**. Pada setiap perulangan, nilai **j** akan dibagi **2** kemudian nilai tersebut akan disimpan dan dicetak oleh variabel **q**. Proses ini berlangsung hingga **j = 5**.

Command Window

```
>> looping3
```

```
q = 0  
q = 2  
q = 6  
q = 12  
q = 20  
q = 30  
q = 42  
q = 56  
q = 72  
q = 90  
q = 110  
>> |
```

Ketika source code dijalankan, looping akan langsung dilakukan dimana looping akan dilakukan sebanyak **11 kali**. Pada setiap perulangannya, nilai **p** akan dikuadratkan dan ditambahkan dengan nilai **p** pada saat perulangan yang sama, kemudian disimpan dan dicetak oleh variabel **q**. Lalu, nilai **p** akan dilakukan *increment* pada setiap perulangan dan proses ini akan terus berlangsung hingga **p = 10**.

2. Analisis “Differensial”

```
Command Window
>> syms x
>> differensial

Masukkan bentuk persamaan f(x) = 3*x.^2 + 5*x + 10
f_real = (sym)

      2
    3*x + 5*x + 10

f_differential = (sym) 6*x + 5
>> |
```

Ketika source code dijalankan, user akan diminta memasukkan persamaan $f(x)$. Kemudian persamaan $f(x)$ asli akan dicetak ke layar. Setelah itu, persamaan $f(x)$ yang sudah diturunkan dicetak ke layar. Jadi, pada layar akan tercetak persamaan $f(x)$ yang sebelum diturunkan dan sesudah diturunkan.

3. Analisis “Integral”

```
Command Window
>> syms x
>> integral

Fungsi = 4*x.^4 - cos(5*x)
f = (sym)

      4
    4*x - cos(5*x)

f_real = (sym)

      4
    4*x - cos(5*x)

f_integral = (sym)

      5
    4*x  sin(5*x)
    ---  - ----
      5      5
```

Ketika source code dijalankan, user akan diminta memasukkan persamaan $f(x)$. Kemudian persamaan $f(x)$ asli akan dicetak ke layar. Setelah itu, persamaan $f(x)$ yang asli akan dicetak ke layar lagi. Lalu, persamaan $f(x)$ yang sudah diintegrasikan dicetak ke layar. Jadi, pada layar akan tercetak persamaan $f(x)$ yang sebelum diturunkan dan sesudah diintegrasikan.

4. Analisis “Function”

```
Command Window
>> myNameIs('Atya')
My name is Atya
>> myNameIs('Budi')
My name is Budi
>> |
```

User akan diminta untuk memanggil fungsi pada source code disertai dengan isi parameternya pada *command window*. Setelah source code dijalankan, maka pernyataan yang terdapat dalam fungsi akan dicetak ke layar. Pernyataan dalam fungsi tersebut disertai dengan nilai dari parameter yang dimasukkan oleh user.

```
Command Window
>> Activity('Budi')
Budi is eating
His/her name is Budi!
>> Activity('Jessica')
Jessica is eating
His/her name is Jessica!
>>
```

User akan diminta untuk memanggil fungsi yang sesuai pada source code yang disertai dengan isi parameternya pada *command window*. Setelah source code dijalankan, maka fungsi tersebut akan memanggil fungsi yang bersifat lokal kemudian mencetak pernyataan yang terdapat dalam fungsi lokal disertai dengan nilai parameternya. Setelah itu, barulah fungsi yang bersifat global mencetak pernyataan yang terdapat dalam fungsi global tersebut yang disertai dengan nilai parameternya.

```
Command Window
>> nilai(29)
hasil = 14.500
ans = 14.500
>> nilai([10 15 20])
hasil =

    5.0000    7.5000   10.0000

ans =

    5.0000    7.5000   10.0000

>> |
```

User akan diminta untuk memanggil fungsi yang sesuai dengan source code yang disertai dengan isi parameternya pada *command window*. Pada pemanggilan fungsi yang pertama, yaitu **nilai (29)**. Hasil akan dicetak pada layar, yaitu **14,500** yang merupakan hasil dari nilai parameter yang dimasukkan user (**29**) dibagi **2**. Lalu,

pada pemanggilan fungsi yang kedua, yaitu **nilai** ([10 15 20]). Pada pemanggilan kali ini, user memberikan 3 nilai parameter sehingga hasil dari operasi fungsi tersebut akan menghasilkan **3 nilai** juga, yaitu **5,0000 7,5000 10,0000**.

```
Command Window
>> [x,y,z] = devide(5,10,20)
x = 2.5000
y = 2
z = 2
>> |
```

User akan diminta untuk memanggil fungsi yang sesuai dengan source code yang disertai dengan isi parameternya pada *command window*. Pada pemanggilan fungsi ini, user memasukkan variabel **x**, **y**, dan **z** untuk menampung hasil dari operasi pada fungsi. Lalu, user memanggil fungsi **devide** dan berisikan parameter dari user yang berjumlah **3**. Kemudian, hasil dari operasi fungsi tersebut akan dicetak ke layar dalam variabel-variabel yang telah dimasukkan oleh user sebelumnya.

```
Command Window
>> squared = @(x) x.^3
squared =

@(x) x .^ 3

>> squared(4)
ans = 64
>> squared(5:8)
ans =

    125    216    343    512
```

User akan diminta untuk memberikan masukan pada *command window* dengan cara menuliskan fungsi dan mendeklarasi serta menginisialisasi operasi pada fungsi tersebut. Pada gambar di atas user menuliskan fungsi **squared** dan menuliskan operasi bahwa nilai **x** akan dieksposen **3**. Setelah membuat fungsi, user memanggil fungsi beserta nilai parameternya. Pada pemanggilan pertama, user memanggil **squared** dengan nilai parameter **4**. Kemudian, user memanggil **squared** dengan nilai parameter **5,6,7**, dan **8** atau jika dituliskan menjadi **(5:8)**.

```

Command Window
>> addition = @(a,b,c) a+b+c
addition =

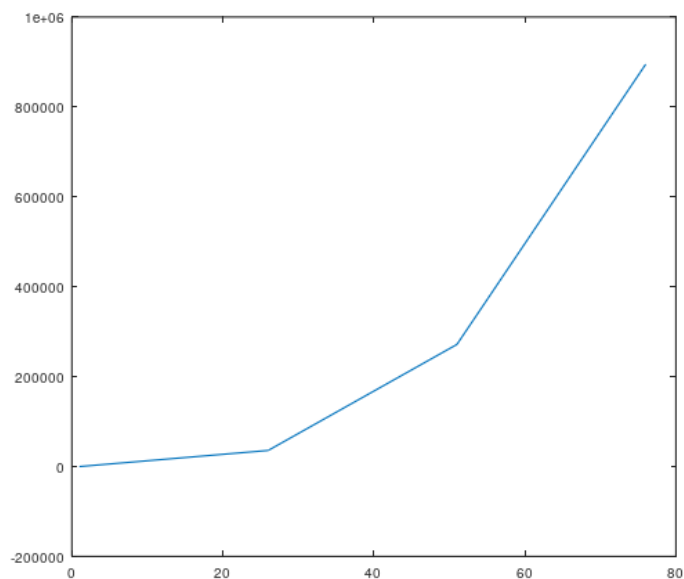
@(a, b, c) a + b + c

>> addition(32, 42, 121)
ans = 195
>> |

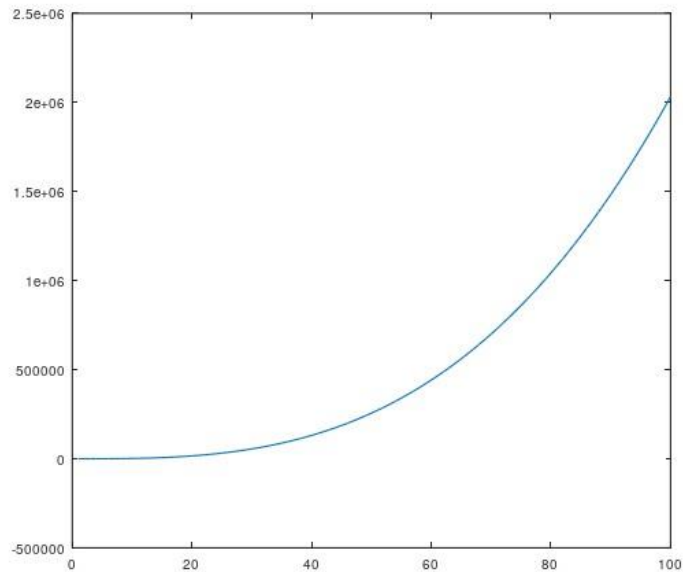
```

User akan diminta untuk memberikan masukan pada *command window* dengan cara menuliskan fungsi dan mendeklarasi serta menginisialisasi operasi pada fungsi tersebut. Pada gambar di atas user menuliskan fungsi **addition** dan menuliskan operasi bahwa nilai **a**, **b**, dan **c** akan dijumlahkan. Setelah membuat fungsi, user memanggil fungsi beserta nilai parameternya. Pada pemanggilan fungsi **addition**, user memanggil **addition** dengan nilai parameter **32**, **42**, dan **121** sehingga akan memberikan hasil akhir berupa **195**.

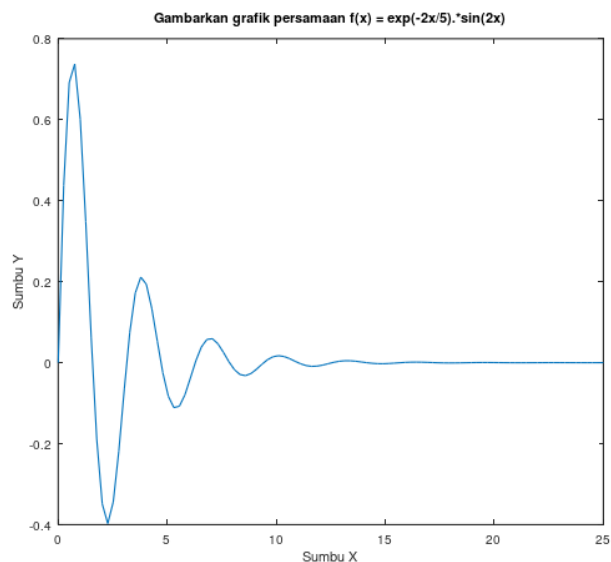
5. Analisis “Grafik”



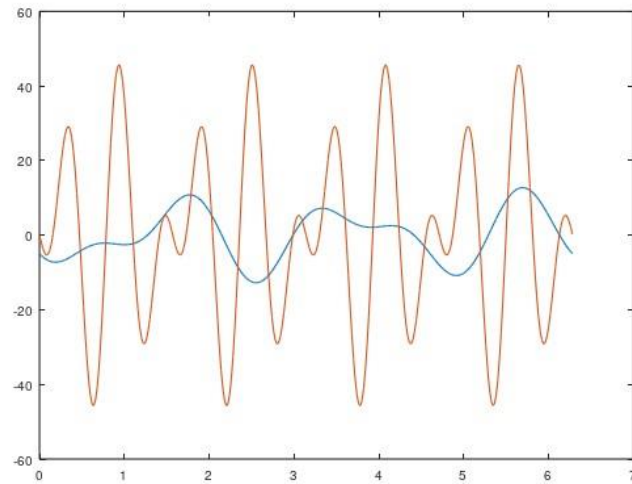
Pada grafik di atas merupakan grafik dua dimensi. Pendefinisian nilai **x** memberikan **4** buah titik **x** pada grafik. Lalu, nilai pada sumbu **y** merupakan hasil dari operasi **persamaan f(x)**. Kemudian, fungsi **plot** menghasilkan grafik dua dimensi seperti grafik di atas.



Pada grafik di atas merupakan grafik dua dimensi. Pendefinisian nilai **x** memberikan **100** buah titik **x** pada grafik. Lalu, nilai pada sumbu **y** merupakan hasil dari operasi **persamaan f(x)**. Kemudian, fungsi **plot** menghasilkan grafik dua dimensi seperti grafik di atas yang lebih landai daripada grafik yang pertama. Hal ini dikarenakan pembagi pada sumbu **x** lebih banyak, yaitu **100**.



Pada grafik di atas merupakan grafik dua dimensi. Pendefinisian nilai **x** memberikan **100** buah titik **x** pada grafik. Lalu, nilai pada sumbu **y** merupakan hasil dari operasi **persamaan f(x)**. Kemudian, fungsi **plot** menghasilkan grafik dua dimensi seperti grafik di atas yang lebih landai daripada grafik yang pertama. Hal ini dikarenakan pembagi pada sumbu **x** lebih banyak, yaitu **100**.



Pada grafik di atas merupakan grafik dua dimensi. Pendefinisian nilai x memberikan $2\pi/0,02$ buah titik x pada grafik. Lalu, nilai pada sumbu y merupakan hasil dari operasi **persamaan y** dan **z** . Kemudian, fungsi **plot** menghasilkan grafik dua dimensi seperti grafik di atas yang terdiri atas dua persamaan.