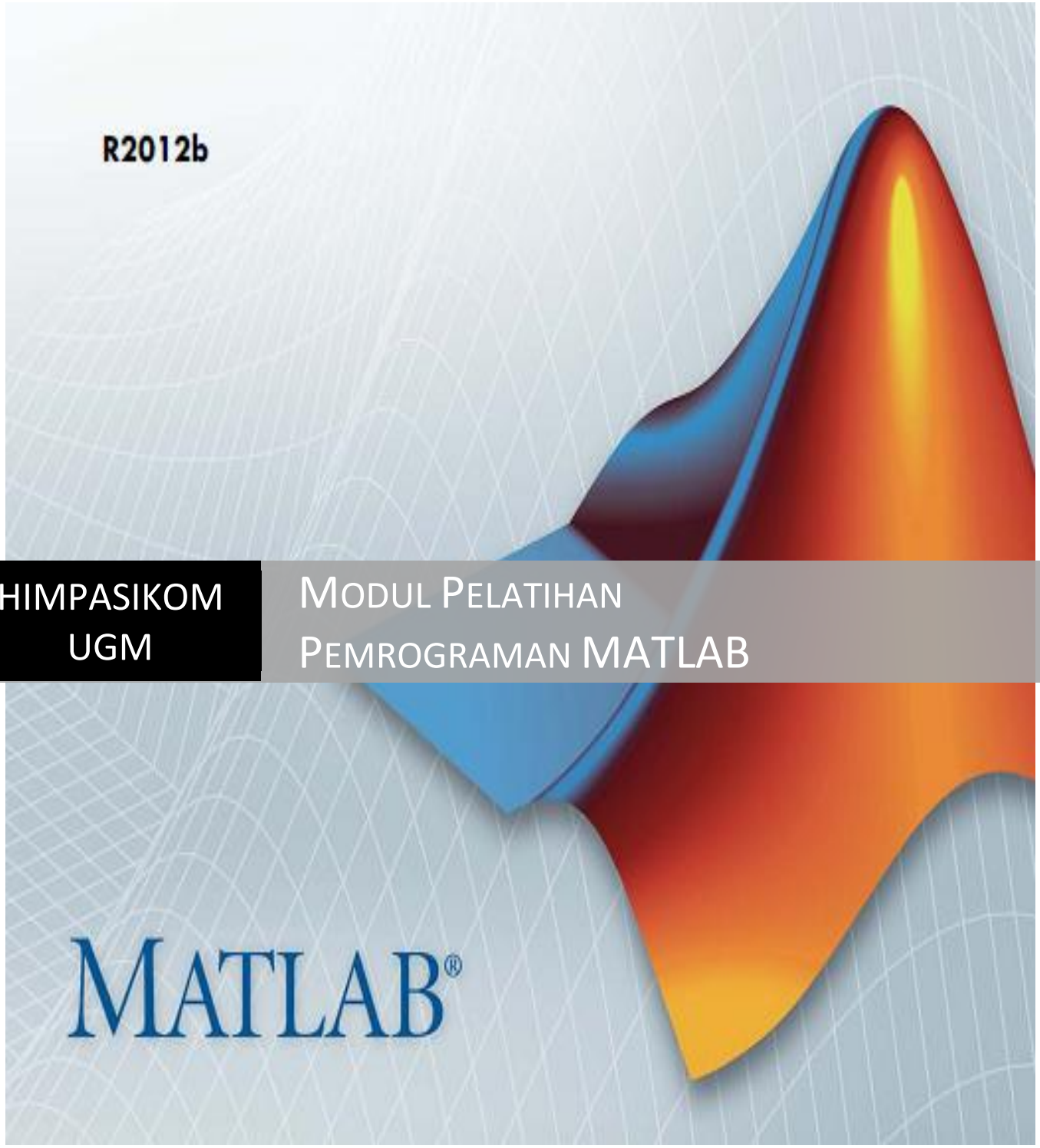


HIMPASIKOM UGM

MODUL PELATIHAN PEMROGRAMAN MATLAB

Yogyakarta, 2 Maret – 6 April 2013 | Pemateri : Septia Rani

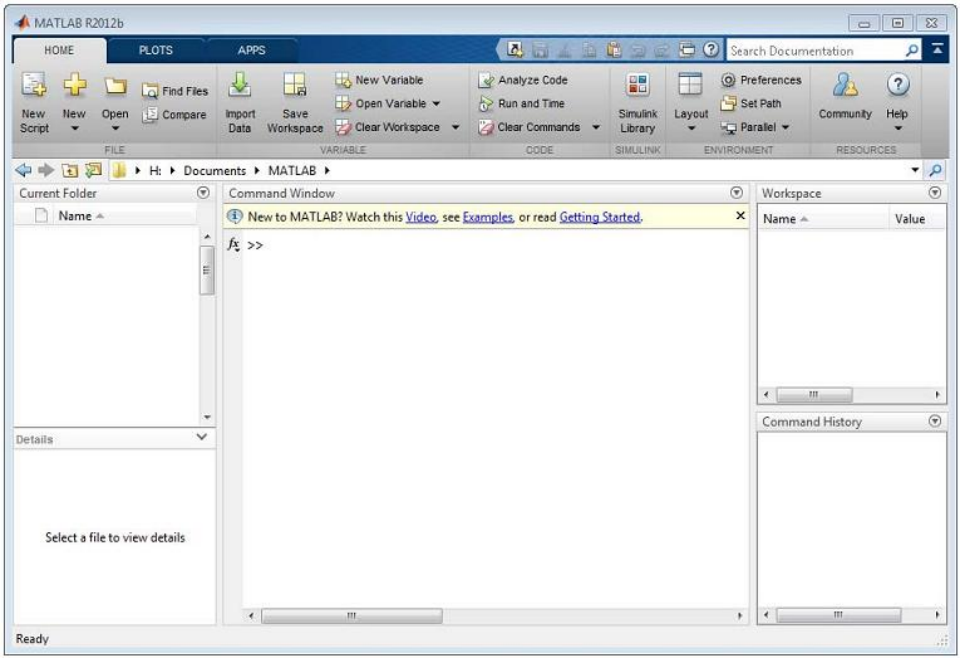
Modul 1
Dasar-Dasar MATLAB

1.1Sekilas MATLAB

MATLAB ® merupakan bahasa pemrograman tingkat tinggi yang dikembangkan oleh MathWorks dan dikhususkan untuk komputasi numerik, visualisasi, dan pemrograman. Dengan memanfaatkan MATLAB, pengguna dapat melakukan analisis data, mengembangkan algoritma, dan membuat model maupun aplikasi. Bahasa, *tools*, dan fungsi-fungsi *built-in* akan memudahkan pengguna untuk mengeksplorasi berbagai pendekatan dan memperoleh solusi dengan lebih cepat dibandingkan apabila menggunakan *spreadsheets* atau bahasa pemrograman tradisional, seperti C/C++ atau Java™. MATLAB menggunakan konsep array/matrik sebagai standar variabel elemennya tanpa memerlukan pendeklarasian array seperti pada bahasa lainnya. Selain itu juga dapat diintegrasikan dengan aplikasi dan bahasa pemrograman eksternal seperti C, Java, .NET, dan Microsoft® Excel®.

1.2Lingkungan Kerja MATLAB

Ketika memulai menjalankan MATLAB, window utama akan terlihat seperti gambar berikut.



Keterangan :

- Current Folder – untuk mengakses file-file pada direktori saat ini.
- Command Window – untuk menuliskan perintah (sintak program).
- Workspace – untuk mengeksplorasi data yang dibuat atau diimport dari file lain.
- Command History – untuk melihat atau menjalankan kembali perintah yang pernah dimasukkan sebelumnya pada *command line*.

Dalam melakukan pemrograman menggunakan bahasa MATLAB, Anda dapat menggunakan beberapa cara sebagai berikut :

Cara 1 : Menggunakan Command Window

Cara ini adalah yang paling sering dilakukan oleh pemula. Perintah akan dieksekusi secara langsung baris perbaris. Untuk membuat program, Anda hanya perlu mengetikkan perintah pada *prompt* MATLAB dalam Command Window, misalnya :

```
>> p = 10;
```

Tekan tombol enter kemudian ketikkan :

```
>> l = 7;
```

Tekan tombol enter kemudian ketikkan :

```
>> luas = p * l
```

Hasil akhir yaitu :

```
>> luas =
       70
```

Jika Anda mengakhiri perintah dengan tanda (;) titik koma, MATLAB akan melakukan komputasi namun tidak menampilkan display output pada Command Window. Anda dapat memanggil kembali perintah sebelumnya dengan menekan tombol *up* dan *down-arrow* (↑ dan ↓). Jika perhitungan menggunakan langkah-langkah yang cukup panjang, maka menggunakan Command Window menjadi kurang efisien.

Cara 2 : Menggunakan M-File

Cara ini biasanya akan dipilih oleh pengguna yang telah terbiasa menggunakan MATLAB. Cara ini memberikan kemudahan untuk mengevaluasi perintah secara keseluruhan. Langkah-langkah menggunakan M-File sebagai berikut :

Pada Command Window ketikkan :

```
>> edit
```

Tekan enter, selanjutnya akan muncul MATLAB editor. Disini Anda dapat mengetikkan program yang akan dijalankan.

```
% MATLAB programming
% by : (fill your name here)

clear all;
clc;

disp('hello world! this is my first program using MATLAB');

p = 10;
l = 7;
area = p * l;
disp(['area : ' num2str(area)]);
```

Langkah selanjutnya Anda harus menyimpan file tersebut, misalnya dengan nama myprogram.m. Untuk menjalankannya, buka kembali Command Window, kemudian ketikkan nama file (tanpa ekstensi) sebagai berikut :

```
>> myprogram
```

Tekan enter. Program Anda selesai dijalankan. ☺

1.3Sintak Dasar pada MATLAB

Seperti bahasa pemrograman lainnya, MATLAB juga memiliki sintak tersendiri. Pada MATLAB hanya terdapat dua tipe data, yaitu numerik dan string. Tidak dibutuhkan pendeklarasian secara eksplisit karena tipe data akan dikenali oleh MATLAB secara otomatis. Namun demikian terdapat beberapa hal penting yang harus diperhatikan dalam penulisan sintak :

- Penamaan variabel bersifat case sensitive.
- Penamaan variabel harus selalu diawali dengan huruf, tidak boleh dengan simbol atau angka.
- Penamaan variabel dan M-File tidak boleh sama dengan nama-nama default yang dikenal MATLAB.

Cara Penulisan Variabel

- Data Numerik Tunggal

```
x = 20;
```

- Data Numerik Berdimensi Banyak (Array/Matrik)

```
x = [ 20 11; 20 13];
```

- Data String

```
x = 'bonjour';
```

Operasi dan Fungsi-Fungsi Matematika

Berikut ini adalah tabel operator matematika yang digunakan dalam pemrograman MATLAB :

Operasi	Simbol	Contoh
Penjumlahan	+	X + Y
Pengurangan	-	X – Y
Perkalian	*	X * Y
Pembagian	/ atau \	X / Y atau X \ Y
Perpangkatan	^	X ^ Y

Selain itu MATLAB juga menyediakan fungsi-fungsi matematika, di antaranya :

Fungsi	Deskripsi
exp	Eksponensial
log	Logaritma natural
log10	Logaritma basis 10

log2	Logaritma basis 2
sqrt	Akar pangkat
cos	Kosinus
sin	Sinus
tan	Tangen

Input dan Output

Untuk meminta input dari user, MATLAB menyediakan fungsi input. Sintak penulisannya sebagai berikut :

```
reply = input('string ditampilkan');
reply = input('string ditamplikan','s');
```

Baris pertama digunakan jika input yang diharapkan berupa angka sedangkan baris kedua digunakan jika input yang diharapkan berupa string.

Sedangkan untuk menampilkan output program ke layar, MATLAB menyediakan fungsi disp. Sintak penulisannya sebagai berikut :

```
disp('string ditampilkan');
```

Control Flow

- **Perulangan dengan (for … end)**

  Sintak ini digunakan untuk melakukan pengulangan proses yang telah diketahui banyaknya perulangan yang harus dijalankan. Cara penulisannya sebagai berikut :

  ```
  for variabel = mulai:interval:akhir
        perintah-perintah
  end
  ```

  Sebagai latihan, buatlah sebuah program yang dapat menampilkan bilangan 1 sampai 100!

- **Perulangan dengan (while … end)**

  Sintak ini digunakan untuk melakukan pengulangan proses yang tidak diketahui banyaknya perulangan yang harus dijalankan. Yang diketahui hanyalah syarat atau kondisi kapan program akan tetap dijalankan. Cara penulisannya sebagai berikut :

  ```
  while syarat
        perintah-perintah
  end
  ```

  Cara penulisan syarat, menggunakan ekspresi matematika yang meliputi :

  == sama dengan
  ~= tidak sama dengan
  > lebih besar dari
  >= lebih besar atau sama dengan
  < lebih kecil dari
  <= lebih kecil atau sama dengan

  Sedangkan untuk mengkombinasikan menggunakan operator logika berikut ini :

  & and
  | or

Sebagai latihan, buatlah program sebagai berikut dengan memanfaatkan perulangan while :
Tulislah program untuk mencari jumlah N bilangan asli pertama. Masukan dari program ini adalah bilangan bulat N. Keluaran dari program ini adalah jumlah N bilangan asli pertama.

- **Kondisional dengan (if … elseif … else … end)**

  Kondisional digunakan untuk mengontrol alur suatu program. Kondisional if-else sering digunakan di dalam program. Cara penulisannya sebagai berikut :

```
if syarat1
       perintah-perintah
elseif syarat2
       perintah-perintah
else
       perintah-perintah
end
```

  Sebagai latihan, buatlah program untuk mengecek apakah suatu bilangan yang diinputkan oleh user merupakan bilangan ganjil atau bilangan genap!

- **Kondisional dengan (switch … case … otherwise … end)**

  Penulisan sintak switch-case sebagai berikut :

```
switch variabel
       case nilai1
              perintah-perintah
       case nilai2
              perintah-perintah
       .
       .
       .
       otherwise
              perintah-perintah
end
```

  Sebagai latihan, tuliskan program yang mengoutputkan nama bulan tertentu berdasarkan angka inputan user.

Modul 2
Array dan Matrik

2.1 Membuat Array dan Matrik

Untuk memasukkan matrik pada MATLAB, ada beberapa cara yang dapat dilakukan :

- Memasukkan secara langsung dengan menuliskan semua elemennya.
- Meng-*load* matrik dari file eksternal.
- Meng-*generate* matrik menggunakan fungsi-fungsi *built-in*.
- Membuat matrik dengan fungsi yang Anda buat sendiri dan disimpan dalam file.

Untuk membuat array dengan empat elemen pada satu baris, setiap elemen harus dipisahkan dengan koma (,) atau spasi. Contohnya sebagai berikut :

```
A = [1 2 3 4]
```

Untuk membuat matrik yang memiliki beberapa baris, maka setiap barisnya harus dipisahkan dengan tanda titik koma (;). Contohnya sebagai berikut :

```
A = [1 2 3; 4 5 6; 7 8 10]
```

Selain itu MATLAB juga menyediakan empat buah fungsi untuk membuat matrik :

- zeros : semua elemennya bernilai nol.
- ones : semua elemennya bernilai satu.
- rand : elemen-elemennya bernilai random dengan distribusi uniform.
- randn : elemen-elemennya bernilai random dengan distribusi normal.

Sebagai contoh dibuat matrik berukuran 5x1 dengan elemennya semua bernilai nol :

```
Z = zeros(5,1)
Z =
    0
    0
    0
    0
    0
```

Sedangkan contoh berikut ini adalah matrik berukuran 3x3 dengan elemen-elemennya dibangkitkan secara random :

```
R = rand(3,3)
R =
    0.8147    0.9134    0.2785
    0.9058    0.6324    0.5469
    0.1270    0.0975    0.9575
```

2.2 Operasi-Operasi pada Array dan Matrik

MATLAB memungkinkan pemrosesan semua nilai pada sebuah matrik menggunakan sebuah operator matematika atau fungsi tunggal.

```
A + 10
ans =
   11  12  13
   14  15  16
   17  18  20
```

Untuk melakukan **transpose** pada sebuah matrik, gunakan tanda petik atas (‘) :

```
A'
ans =
    1   4   7
    2   5   8
    3   6   10
```

Untuk melakukan **perkalian** pada matrik, gunakan operator (*). Sebagai contoh, sebuah matrik apabila dikalikan dengan hasil inversnya maka akan menghasilkan matrik identitas :

```
P = A * inv(A)
P =
    1.0000 0       -0.0000
    0       1.0000 0
    0       0       1.0000
```

Untuk melakukan perkalian elemen *by* elemen pada matrik (bukan perkalian matrik), gunakan operator (.*) :

```
P = A .* A
P =
     1      4      9
    16     25     36
    49     64    100
```

Berikut ini daftar operator pada array :

Operator	Deskripsi
+	Penambahan
-	Pengurangan
.*	Perkalian per elemen
./	Pembagian per elemen
.\	Pembagian kiri per elemen
.^	Perpangkatan per elemen

Operasi pada array sangat bermanfaat untuk pembentukan tabel. Misal diberikan n sebuah vector kolom sebagai berikut :

```
n = (0:9)';
```

kemudian fungsi berikut ini akan membuat sebuah tabel yang berisi bilangan kuadrat dan perpangkatan dari 2 :

```
pows = [n  n.^2  2.^n]
pows =
     0      0      1
     1      1      2
     2      4      4
     3      9      8
     4     16     16
     5     25     32
     6     36     64
     7     49    128
     8     64    256
     9     81    512
```

Selain operasi-operasi yang telah dijelaskan di atas, MATLAB memiliki beberapa fungsi yang sering digunakan untuk operasi pada matrik, di antaranya :

Fungsi	Deskripsi
diag	Untuk mengambil elemen diagonal pada sebuah matrik
inv	Untuk melakukan operasi inverse pada matrik
size	Untuk mengetahui dimensi sebuah matrik
sort	Untuk melakukan pengurutan pada matrik

Untuk mengetahui lebih detail cara penggunaan dari masing-masing fungsi tersebut, pada Command Window ketikkan :

```
help nama_fungsi
```

kemudian tekan enter.

2.3 Konkatenasi

Konkatenasi merupakan proses penggabungan dua buah array sehingga diperoleh sebuah array dengan ukuran yang lebih besar. Tanda [] merupakan operator untuk konkatenasi.

```
B = [A,A]
B =
     1     2     3     1     2     3
     4     5     6     4     5     6
     7     8    10     7     8    10
```

Operasi di atas disebut dengan konkatenasi horizontal dan dapat dilakukan apabila banyaknya baris kedua array sama. Selain itu, apabila banyaknya kolom pada dua buah array sama maka dapat dilakukan operasi konkatenasi vertikal sebagai berikut :

```
B = [A;A]
B =
     1     2     3
     4     5     6
     7     8    10
     1     2     3
     4     5     6
     7     8    10
```

2.4 Indeks pada Array

Setiap variabel pada MATLAB merupakan sebuah array yang dapat terdiri atas beberapa bilangan. Jika Anda ingin mengakses sebuah elemen pada array, maka harus menggunakan indeks.

Sebagai contoh, diberikan matrik *magic square 4x4* sebagai berikut :

```
A = magic(4)
A =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

Terdapat dua buah cara untuk mengakses suatu elemen tertentu pada sebuah array. Cara paling umum adalah dengan menyebutkan baris dan juga kolomnya seperti contoh berikut :

```
A(3,4)
ans =
    12
```

Cara kedua adalah dengan melakukan iterasi ke bawah pada setiap kolom secara berurutan :

```
A(15)
```

```
ans =
    12
```

Anda juga dapat menambahkan satu elemen baru pada matrik di luar ukuran dimensi saat ini. Secara otomatis ukuran matrik akan bertambah sehingga dapat menampung elemen tersebut.

```
A(4,5) = 17
A =

    16     2     3    13     0
     5    11    10     8     0
     9     7     6    12     0
     4    14    15     1    17
```

Untuk mengakses beberapa elemen pada array, gunakan operator (:) yang memungkinkan Anda untuk menspesifikasikan rentang dalam format *start:end*. Sebagai contoh, berikut ini akan ditampilkan tiga elemen pertama pada kolom ke-2 matrik A :

```
A(1:3,2)
ans =
     2
    11
     7
```

Jika tidak dituliskan nilai awal maupun nilai akhir, maka penggunaan tanda (:) akan menspesifikasikan semua elemen pada baris atau kolom yang dimaksud. Sebagai contoh, berikut ini akan ditampilkan semua elemen pada baris ke-3 matrik A :

```
A(3,:)
ans =
     9     7     6    12     0
```

2.5 Menghapus Baris dan Kolom

Anda dapat menghapus baris atau kolom dari sebuah matrik dengan menggunakan sepasang tanda kurung siku ([]). Misalkan terdapat matrik X sebagai berikut :

```
X = magic(4);
```

Kemudian akan dihapus kolom kedua dari matrik X, menggunakan :

```
X(:,2) = []
X =
    16     3    13
     5    10     8
     9     6    12
     4    15     1
```

Untuk menghapus baris pertama dari matrik X, menggunakan :

```
X(1,:) = []
X =
     5    10     8
     9     6    12
     4    15     1
```

Jika yang ingin dihapus hanya satu buah elemen saja, maka hasilnya bukan lagi sebuah matrik, sehingga ekspresi seperti berikut ini :

```
X(1,2) = []
```

akan menghasilkan error.

Latihan

Agar lebih memahami penggunaan MATLAB untuk mengolah data array/matrik, kerjakan latihan berikut ini :

1. Buat program yang meminta inputan angka sebanyak N, dimana N adalah inputan user. Output program adalah bilangan maksimum dan minimum dari angka-angka tersebut. (Gunakan array untuk menyimpan data angka-angka tersebut).
   Contoh:
   input N: 4
   angka[1]: 3
   angka[2]: 11
   angka[3]: 8
   angka[4]: -5
   maks= 11
   min=-5
2. Ron sangat suka bermain dengan angka-angka. Suatu hari dia iseng belajar pemrograman MATLAB. Ron ingin membuat program yang dapat menyimpan inputan berupa bilangan-bilangan integer ke dalam array, kemudian dari bilangan-bilangan yang disimpan tersebut akan ditampilkan bilangan-bilangan yang habis dibagi 4. Karena masih asing dengan MATLAB, Ron mengalami kesulitan untuk membuat program tersebut. Hermione yang biasa membantunya juga sedang sibuk mengerjakan tugas sekolah. Jadi, bantulah Ron untuk membuat programnya!
   Contoh:
   input: 1 1 2 2 8 6 4 2 18 16 4
   output: 8 4 16 4

Modul 3
Fungsi & Interaksi dengan File Eksternal

Bagian 1 : Fungsi

3.1 Fungsi dalam M-File

Pada dasarnya, semua tools yang disediakan oleh MATLAB dibuat dalam format fungsi dan dikelompokkan ke dalam folder-folder toolbox. Selain menggunakan fungsi-fungsi yang telah ada tersebut, kita juga dapat membuat fungsi-fungsi sendiri sesuai kebutuhan. Keuntungan membuat program dalam format fungsi adalah kemudahannya untuk digunakan lagi pada program lainnya. Berikut ini pola untuk menuliskan fungsi pada MATLAB :

```
Function [out1,out2,…] = Nama (in1,in2,…)   → bagian deklarasi fungsi
% penjelasan                                → bagian penjelasan fungsi (opsional)
- perintah -                                → bagian program utama
- perintah -
```

Agar lebih memahami cara penulisan fungsi, perhatikan contoh program berikut :

```
function y = pangkat(a,b)
%---------------------------------------
%fungsi untuk menghitung perpangkatan a^b
%cara menggunakan :
%y = pangkat(2,10)
%---------------------------------------
hasil = 1;
for i=1:b
    hasil = hasil * a;
end
y = hasil;
```

Simpan program di atas dengan nama sama dengan nama fungsinya yaitu pangkat.m. Untuk menggunakan fungsi tersebut, pada Command Window ketikkan perintah sebagai berikut :

```
>> y = pangkat(2,6)
```

sehingga akan muncul hasil perhitungan pangkatnya. Kemudian apabila ingin melihat penjelasan fungsi, ketikkan perintah sebagai berikut :

```
>> help pangkat
```

3.2 Pemanggilan Fungsi

MATLAB menyediakan banyak sekali fungsi-fungsi yang dapat menyelesaikan berbagai permasalahan komputasi. Misalkan diberikan variabel A dan B sebagai berikut :

```
A = [1 3 5];
B = [10 6 4];
```

Untuk memanggil fungsi, parameter input dituliskan di dalam tanda kurung :

```
max(A);
```

Jika terdapat beberapa parameter input, maka penulisannya dipisahkan dengan tanda koma :

```
max(A,B);
```

Hasil output dari suatu fungsi dapat disimpan ke dalam sebuah variabel :

```
maxA = max(A);
```

Jika terdapat beberapa parameter output, maka penulisannya menggunakan tanda kurung siku :

```
[maxA,location] = max(A);
```

Untuk memanggil sebuah fungsi yang tidak memerlukan input dan tidak mengembalikan suatu output maka tuliskan nama fungsinya saja :

```
clc
```

Fungsi clc akan melakukan *clear* pada Command Window.

Latihan

Agar lebih memahami penulisan dan penggunaan fungsi dalam MATLAB perhatikan program berikut ini :

```
function [luas,volume] = balok (p,l,t)
% -------------------------------------------------
%fungsi : untuk menghitung luas dan volume pada balok
%cara menggunakan :
%[luas,volume] = balok (12,10,5)
%-------------------------------------------------
luas = 2 * (p*l + p*t + l*t);
volume = p*l*t;
```

Simpan program tersebut dengan nama balok.m. Selanjutnya, tugas Anda, buatlah fungsi untuk menghitung luas permukaan dan volume pada tabung dan pada kerucut. Simpan fungsi tersebut dengan nama tabung.m dan kerucut.m.

Setelah Anda selesai membuat fungsi-fungsi tersebut, buatlah sebuah script yang meminta inputan dari user, bangun ruang apakah yang akan dihitung luas dan volumenya.

Contoh tampilan program :

>> Program Hitung Luas dan Volume <<
Pilihan bangun ruang :
1. Balok
2. Tabung
3. Kerucut
Masukkan pilihan Anda : 1
>> Hitung Luas dan Volume Balok <<
Masukkan panjang balok : 10
Masukkan lebar balok : 8
Masukkan tinggi balok : 5
Luas permukaan balok = 340
Volume balok = 400

Bagian 2 : Interaksi dengan File Eksternal

3.3 Metode Save dan Load

Pada MATLAB, metode termudah untuk berinteraksi dengan file adalah menggunakan fungsi save dan load. Fungsi save digunakan untuk menyimpan data dari memori MATLAB ke file, sedangkan fungsi load digunakan untuk mengambil data file ke memori MATLAB.

Interaksi dengan File MAT

File dengan ekstensi MAT adalah file biner yang hanya dapat dibuka oleh MATLAB. Jenis file MAT dapat menyimpan informasi data berupa variabel beserta nilai-nilainya tanpa merepotkan Anda untuk mengatur format datanya. Satu file MAT dapat menyimpan beberapa variabel sekaligus beserta nilainya. Berikut ini adalah sintak untuk menggunakan fungsi save :

```
save [nama_file] var1 var2;
```

Nama_file diisi dengan nama file yang akan kita simpan, diketikkan tanpa ekstensi. File tersebut akan otomatis berekstensi MAT dan menyimpan data nilai-nilai var1 dan var2. Agar lebih memahami mengenai fungsi save, coba jalankan program berikut ini :

```
%Program save file MAT
clear all;
clc;
disp('Program save file MAT');
X = rand(20,20);
Y = rand(10,10);
save datarandom X Y;
disp('Data telah disimpan!');
```

Setelah script tersebut dijalankan maka pada directori aktif saat ini akan ada file baru dengan nama datarandom.mat. Untuk mengambil kembali data yang telah disimpan tadi, gunakan fungsi load dengan sintak sebagai berikut :

```
load nama_file
```

Interaksi dengan File Teks

Untuk menyimpan dalam format teks ascii menggunakan sintak sebagai berikut :

```
save nama_file variabel –ascii
```

Nama file boleh ditulis lengkap dengan ekstensinya. Ketika menggunakan fungsi save untuk tipe ascii Anda dapat menyimpan beberapa nilai dari variabel, namun direkomendasikan untuk hanya menyimpan data dari satu variabel saja agar nantinya tidak kesulitan saat harus mengaksesnya kembali. Agar lebih memahami penggunaan fungsi save pada file teks, coba jalankan program berikut ini :

```
%Program save file txt
clear all;
clc;
disp('Program save file txt');
n_depan = input('masukkan nama depan : ','s');
n_belakang = input('masukkan nama belakang : ','s');
save namadepan.txt n_depan –ascii;
save namabelakang.txt n_belakang –ascii;
disp('Data telah disimpan!');
```

Kemudian data file teks yang telah Anda simpan dapat diambil kembali untuk diolah lebih lanjut dengan sintak sebagai berikut :

```
var = load('nama_file.txt');
```

Apabila data yang dibaca aslinya adalah string, maka harus diconvert lagi ke dalam format string karena data yang terbaca masih dalam format ascii. Gunakan sintak berikut ini :

```
var = char(var);
```

3.4 Interaksi dengan Ms. Excel

Membaca File Microsoft Excel

Anda dapat membaca sebuah file Ms. Excel dari MATLAB dengan cara sebagai berikut :

```
num = xlsread(filename)
```

Sintak tersebut bertujuan untuk membaca data dari worksheet pertama pada file Ms. Excel dengan nama sesuai dengan filename dan mengembalikan data numerik yang akan disimpan pada array num. Apabila yang akan dibaca adalah worksheet tertentu maka sintaknya sebagai berikut :

```
num = xlsread(filename,sheet)
```

Setelah Anda mendapatkan data dari dokumen Ms. Excel, selanjutnya dengan menggunakan script-script pada MATLAB Anda dapat mengolah data dengan lebih mudah dan sesuai dengan kebutuhan.

Menuliskan Data pada File Microsoft Excel

Untuk menuliskan data pada sebuah file Ms. Excel menggunakan cara sebagai berikut :

xlswrite(filename,A)

Sintak tersebut bertujuan untuk menuliskan array A ke dalam worksheet pertama pada file Excel, dimulai pada cell A1. Apabila akan dituliskan bukan pada worksheet pertama maka menggunakan sintak sebagai berikut :

xlswrite(filename,A,sheet)

3.5 Interaksi dengan File Gambar

Membaca File Gambar

Berikut ini adalah sintak umum untuk membaca sebuah file gambar :

```
image1 = imread('namafile');
```

Setelah file gambar dibaca menggunakan sintak di atas, selanjutnya data gambar akan tersimpan dalam variabel "image1". Variabel tersebut tidak ada bedanya dengan variabel MATLAB lainnya. Semua fungsi penanganan matrik MATLAB dapat diterapkan terhadap variabel tersebut.

Menampilkan File Fambar

Untuk menampilkan file gambar yang telah dibaca, menggunakan sintak sebagai berikut :

```
imshow(nama_variabel_image);
```

Jika perintah tersebut dijalankan maka akan muncul window baru yang menampilkan gambar image.

Menuliskan File Fambar

Selain membaca dan menampilkan file gambar, Anda juga dapat membuat file gambar baru dengan menggunakan sintak sebagai berikut :

```
imwrite(nama_variabel_image,'namafile');
```

Agar lebih memperjelas pemahaman terhadap interaksi dengan file gambar, jalankan program berikut ini :

```
%Program interaksi dengan file gambar
clear all;
clc;
disp('Program interaksi dengan file gambar');
disp('step 1 : baca data');
image1 = imread('gambar_rgb.jpg');
disp('data sudah tersimpan di variabel image1');
disp('step 2 : tampilkan data');
imshow(image1);
disp('step 3 : manipulasi data');
disp('konversi citra rgb ke citra grayscale');
gray(:,:,1) = uint8( round( (double(image1(:,:,1)) + double(image1(:,:,2)) +
double(image1(:,:,3))) / 3 ) ) ;
gray(:,:,2) = gray(:,:,1);
gray(:,:,3) = gray(:,:,1);
disp('step 4 : simpan data');
imwrite(gray,'gambar_grayscale.jpg');
disp('Data telah disimpan!');
```

Setelah program di atas dijalankan, pada directori aktif saat ini akan ada file baru yang merupakan citra versi grayscale dari citra ‘gambar_rgb.jpg’.

Selain dapat berinteraksi dengan file-file yang dijelaskan di atas, MATLAB juga dapat berinteraksi dengan file-file jenis lainnya, seperti file suara dan juga program exe. Untuk mengetahui cara interaksi MATLAB dengan file-file tersebut, Anda bisa mempelajarinya sendiri melalui fasilitas help maupun dari sumber lainnya seperti internet ☺.

Modul 4
Visualisasi Data

4.1 Membuat Plot Data

Terdapat beberapa bentuk dari fungsi plot, tergantung pada argumen inputnya.

- Jika y adalah sebuah vector, maka sintak plot(y) akan menghasilkan sebuah grafik linear dengan sumbu x merupakan nilai index dari elemen y, sedangkan sumbu y merupakan nilai y itu sendiri.
- Jika terdapat dua vector sebagai argumennya, maka sintak plot(x,y) akan menghasilkan grafik y versus x.

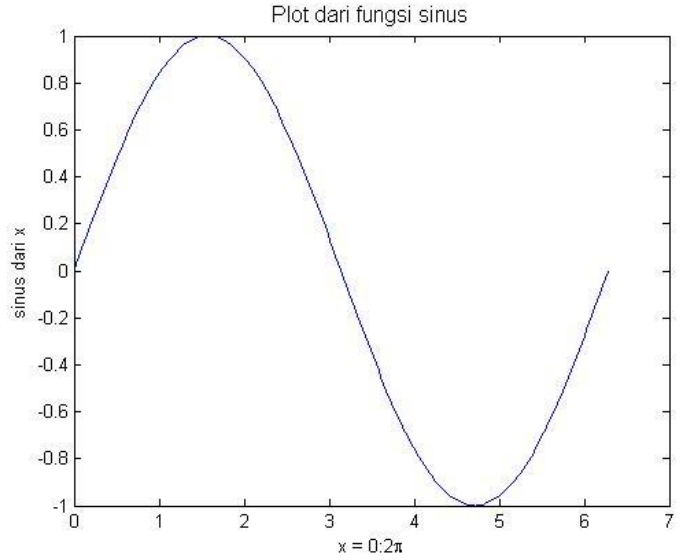
Sebagai contoh, sebuah vektor x bernilai antara 0 sampai dengan 2π, hitung nilai sinusnya, kemudian plot hasilnya :

```
x = 0:pi/100:2*pi;
y = sin(x);
plot(x,y)
```

Tambahkan label axis dan judul grafik :

```
xlabel('x = 0:2\pi')
ylabel('sinus dari x')
title('Plot dari fungsi sinus','FontSize',12)
```

Karakter \pi digunakan untuk membuat simbol π dan properti FontSize digunakan untuk memperbesar ukuran text yang digunakan untuk judul. Hasilnya sebagai berikut :



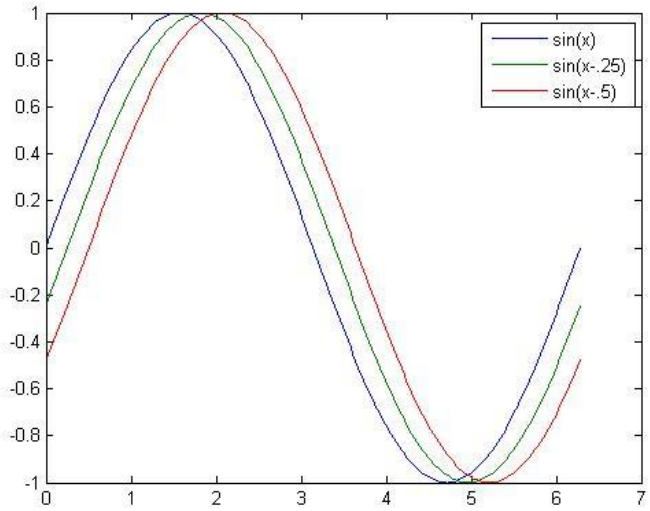
4.2 Membuat Plot Multiple Dataset pada Satu Gambar

Beberapa pasangan argumen x-y akan menghasilkan beberapa grafik dengan menggunakan sekali pemanggilan fungsi plot. MATLAB akan menggunakan warna yang berbeda untuk masing-masing garis. Sebagai contoh, perintah berikut ini akan melakukan plot terhadap tiga buah fungsi yang berbeda :

```
x = 0:pi/100:2*pi;
```

```
y = sin(x);
y2 = sin(x-.25);
y3 = sin(x-.5);
plot(x,y,x,y2,x,y3)
legend('sin(x)','sin(x-.25)','sin(x-.5)')
```

Berikut ini hasilnya :



4.3 Mengubah Bentuk dan Warna Garis

Anda dapat mengubah warna, bentuk garis, dan juga marker (seperti tanda plus dan lingkaran) ketika Anda melakukan plot data menggunakan perintah berikut ini :

```
plot(x,y,'color_style_marker')
```

color_style_marker merupakan sebuah string yang terdiri atas satu sampai empat buah karakter yang merepresentasikan warna, bentuk garis, dan tipe marker. Sebagai contoh :

```
plot(x,y,'r:+')
```

Perintah di atas akan melakukan plot data menggunakan dotted line berwarna merah dan menempatkan sebuah marker + pada setiap titik data.

Berikut ini tabel simbol yang dapat digunakan :

Type	Values	Meanings
Color	'c' 'm' 'y' 'r' 'g' 'b' 'w' 'k'	cyan magenta yellow red green blue white black
Line style	'-' '--' ':' '.-' no character	solid dashed dotted dash-dot no line

Marker type	'+'	plus mark
	'o'	unfilled circle
	'*'	asterisk
	'x'	letter x
	's'	filled square
	'd'	filled diamond
	'^'	filled upward triangle
	'v'	filled downward triangle
	'>'	filled right-pointing triangle
	'<'	filled left-pointing triangle
	'p'	filled pentagram
	'h'	filled hexagram
	no character	no marker

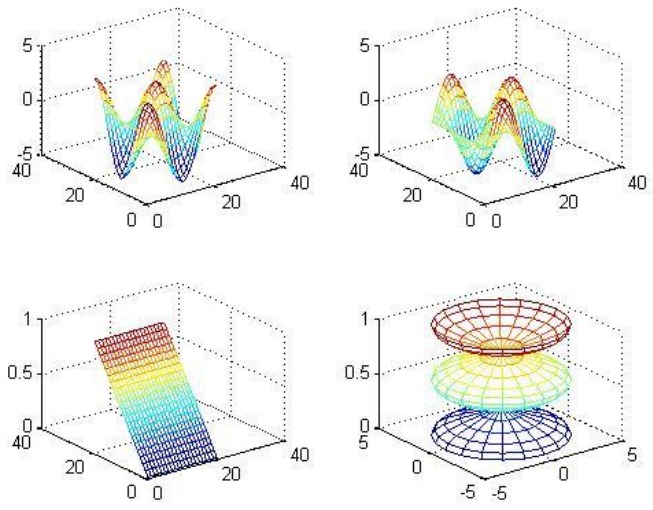
4.4 Menampilkan Beberapa Plot dalam Satu Figure

Perintah subplot memungkinkan Anda untuk menampilkan beberapa plot pada sebuah window yang sama. Perintah berikut ini,

```
subplot(m,n,p)
```

akan membagi figure window ke dalam matrik subplot berukuran mxn dan memilih subplot ke-p untuk plot saat ini. Penomoran plot dimulai dari baris pertama pada figure window, kemudian baris kedua, dst. Sebagai contoh, perintah berikut ini akan melakukan plot data ke dalam empat buah bagian subregion pada figure window :

```
t = 0:pi/10:2*pi;
[X,Y,Z] = cylinder(4*cos(t));
subplot(2,2,1); mesh(X)
subplot(2,2,2); mesh(Y)
subplot(2,2,3); mesh(Z)
subplot(2,2,4); mesh(X,Y,Z)
```



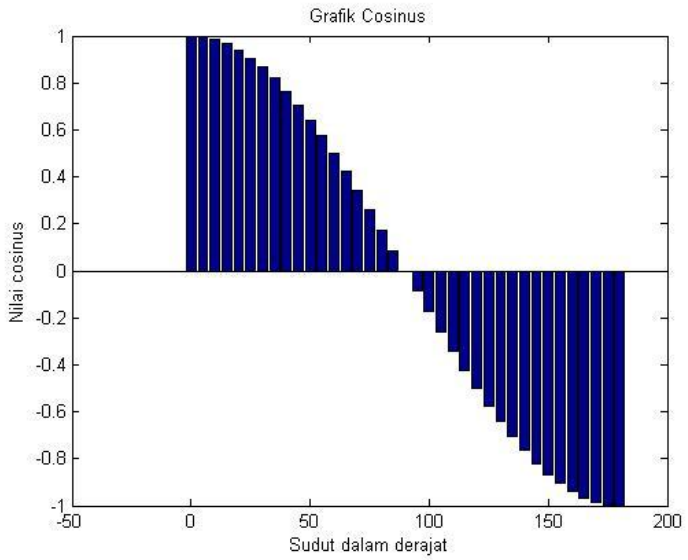
4.5 Visualisasi Data 2D dengan Fungsi Bar

Fungsi bar digunakan untuk menampilkan data dalam grafik batang. Sintak penulisannya sebagai berikut :

bar(x,y);

Contoh pemakaian dalam program sebagai berikut :

```
sudut = [0:5:180];
y = cos(sudut*pi/180);
bar(sudut,y)
title('Grafik Cosinus');
xlabel('Sudut dalam derajat');
ylabel('Nilai cosinus');
```



Selain fungsi plot dan juga fungsi bar, MATLAB juga masih menyediakan beberapa fungsi visualisasi 2D lainnya antara lain fungsi stem dan fungsi stair. Lebih lanjut mengenai fungsi-fungsi lainnya dapat Anda pelajari sendiri ☺.

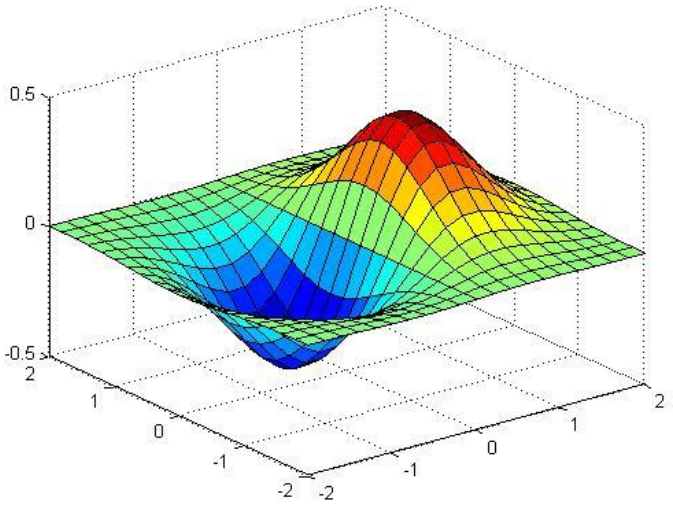
4.6 Visualisasi Data 3D

Plot tiga dimensi biasanya digunakan untuk menampilkan suatu permukaaan yang didefinisikan oleh fungsi dengan dua buah variabel, z = f(x,y). Untuk memperoleh z, pertama-tama buatlah sekumpulan titik (x,y) pada domain fungsi menggunakan meshgrid.

```
[X,Y] = meshgrid(-2:.2:2);
Z = X .* exp(-X.^2 - Y.^2);
```

Kemudian buat plot permukaannya.

```
surf(X,Y,Z)
```

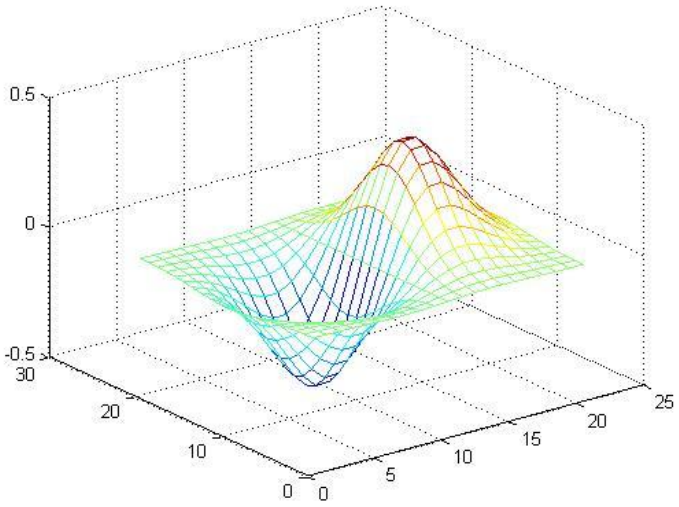


Fungsi mesh

Fungsi ini digunakan untuk menampilkan data dalam bentuk permukaan 3D. Cirinya berupa grid-grid yang menghubungkan 4 titik terdekat dalam ruang 3D. Sintak penulisannya sebagai berikut :

```
mesh(Z)
```

Jika fungsi Z di atas digambarkan menggunakan fungsi mesh, hasilnya sebagai berikut :



Fungsi surf

Fungsi ini digunakan untuk menampilkan data dalam bentuk permukaan 3D seperti pada mesh. Bedanya adalah setiap grid diisi dengan warna tertentu sesuai bobot nilai. Sintak penulisannya sebagai berikut :

```
surf(Z)
```

atau

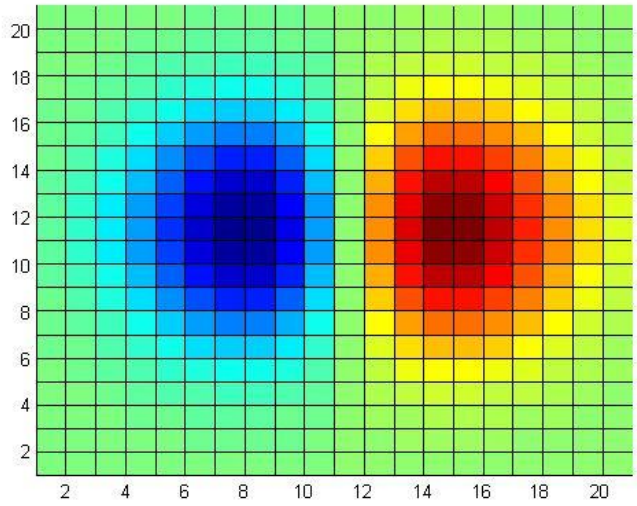
```
surf(X,Y,Z)
```

Fungsi pcolor

Fungsi ini digunakan untuk menampilkan data 3D dalam bentuk permukaan 2D (tampak atas). Cirinya berupa grid berwarna yang menunjukkan bobot nilai tertentu. Sintak penulisannya sebagai berikut :

```
pcolor(Z)
```

Jika fungsi Z di atas digambarkan menggunakan fungsi pcolor, hasilnya sebagai berikut :

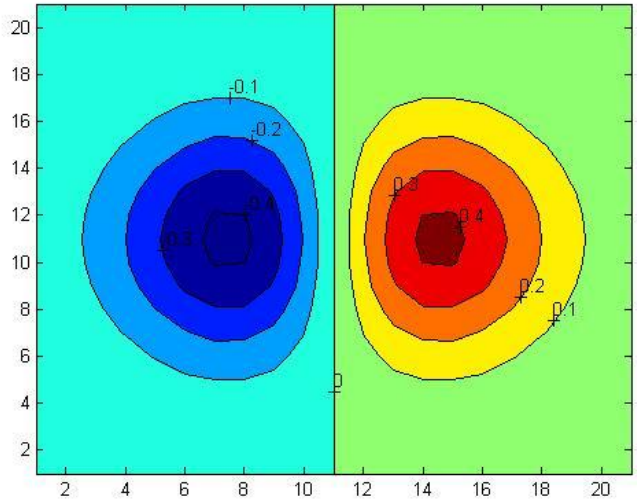


Fungsi contourf

Fungsi ini digunakan untuk membuat garis kontur dari data 3D. Garis-garis kontur ini dibuat dengan teknik interpolasi dari titik-titik terdekat dan pada setiap level kontur diberikan warna sesuai bobot garis konturnya. Sintak penulisannya sebagai berikut :

```
obj_handle = contourf(Z)
clabel(obj_handle)
```

obj_handle adalah object handle yang dibutuhkan untuk menampilkan label pada garis kontur. Hasilnya sebagai berikut :



Latihan

Berikut ini adalah data nilai matakuliah pemrograman dari sekelompok mahasiswa :

No Mhs	Kuis	UTS	UAS
1	81	65	86
2	90	75	78
3	21	48	76
4	91	93	79
5	63	67	54
6	50	75	48
7	27	74	44
8	54	39	64
9	95	65	70
10	96	71	75
11	15	70	27
12	97	75	67
13	95	72	65
14	48	46	32
15	80	97	88
16	14	28	49
17	42	69	59
18	91	81	68
19	79	95	58
20	95	68	66

Salinlah data tersebut ke dalam file excel, kemudian bacalah file excel tersebut menggunakan MATLAB. Langkah selanjutnya, hitunglah nilai akhir tiap mahasiswa dengan rumus sebagai berikut :

Nilai akhir = 40% x UTS + 40% x UAS + 20% x Kuis

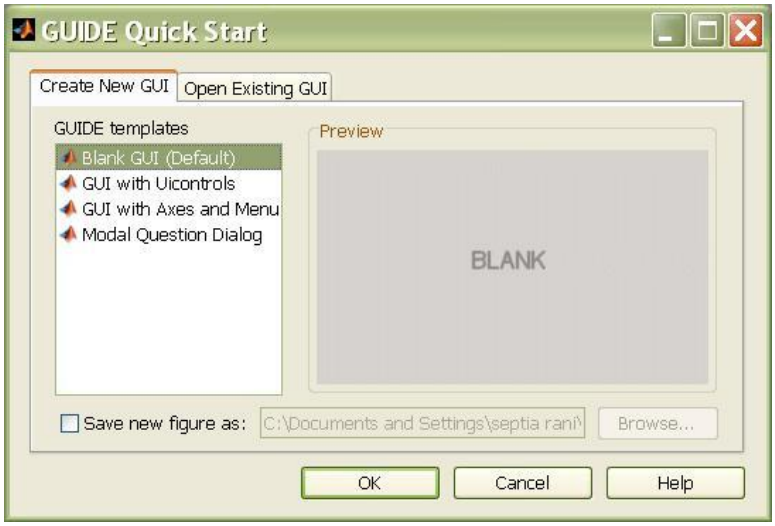
Setelah diperoleh nilai akhir dari tiap mahasiswa, laporkan hasilnya dalam bentuk grafik batang, sumbu x menyatakan tiap mahasiswa dan sumbu y menyatakan nilai akhir dari mahasiswa yang bersangkutan.

Modul 5
Graphic User Interface (GUI)

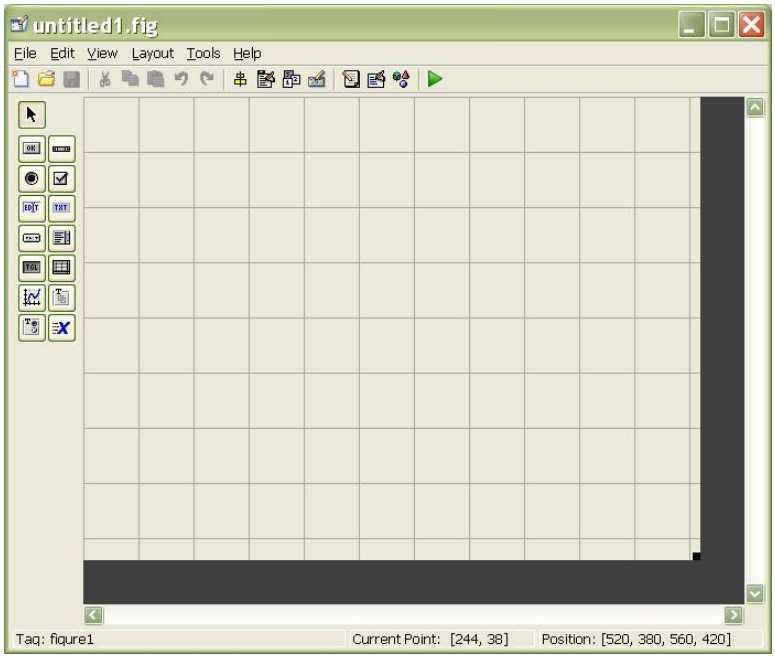
Mengapa menggunakan GUI di MATLAB? Alasan utama menggunakan GUI karena dapat memudahkan *end-user* untuk mengoperasikan program yang telah dibuat. Jika tidak ada GUI, maka *user* harus bekerja melalui interface command line yang tentu saja lebih sulit.

5.1 Menggunakan GUIDE (GUI Creator)

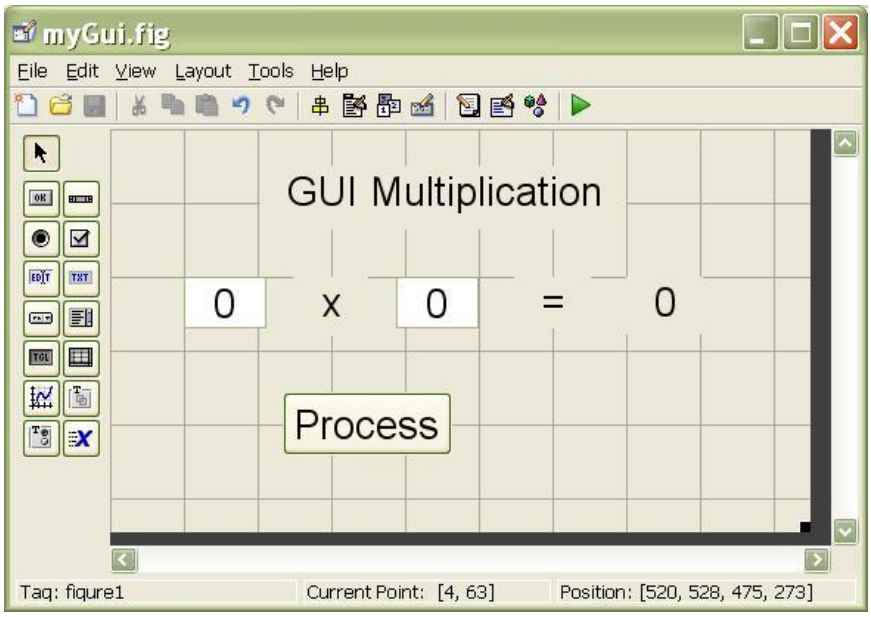
1. Langkah pertama, pada Command Window ketikkan perintah berikut ini :
   ```
   >> guide
   ```
2. Selanjutnya akan muncul window seperti gambar di bawah ini. Pilih option pertama : Blank GUI.



3. Selanjutnya akan muncul window seperti gambar di bawah ini.

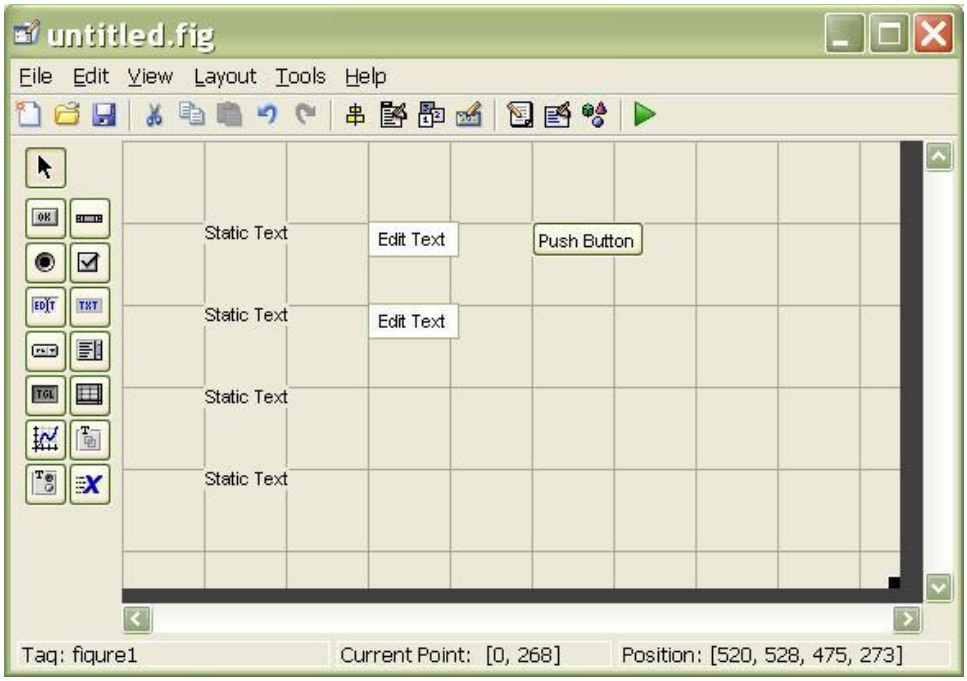


4. Sebelum menambahkan komponen pada figure, sebaiknya Anda sudah membuat desain / gambaran seperti apakah tampilan GUI nantinya. Berikut ini contoh dari tampilan GUI yang sudah jadi.

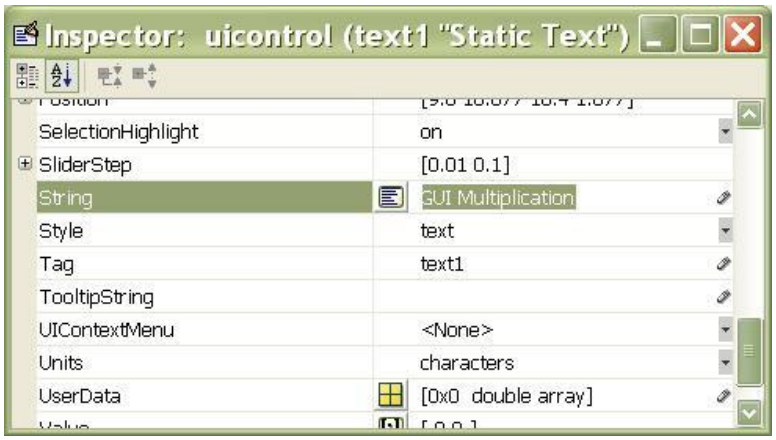


5.2 Membuat Aspek Visual GUI

1. Untuk GUI Multiplication (perkalian), kita akan membutuhkan komponen berikut ini :
   - Empat buah komponen Static Text
   - Dua buah komponen Edit Text
   - Satu buah komponen Push Button

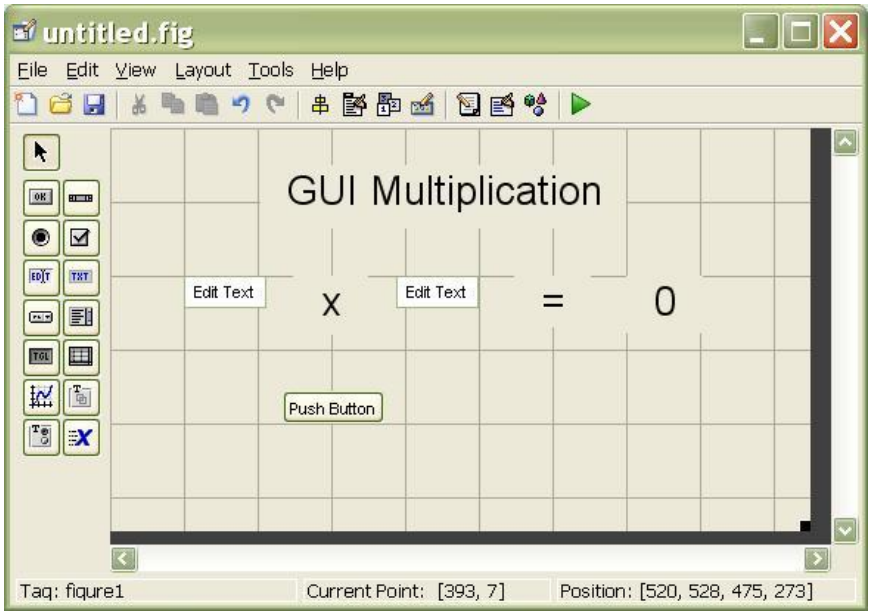


2. Selanjutnya, kita perlu mengedit properties dari komponen-komponen tadi. Dimulai dari Static Text. Double click pada Static Text, maka akan muncul tabel seperti gambar di bawah ini (disebut Property Inspector).

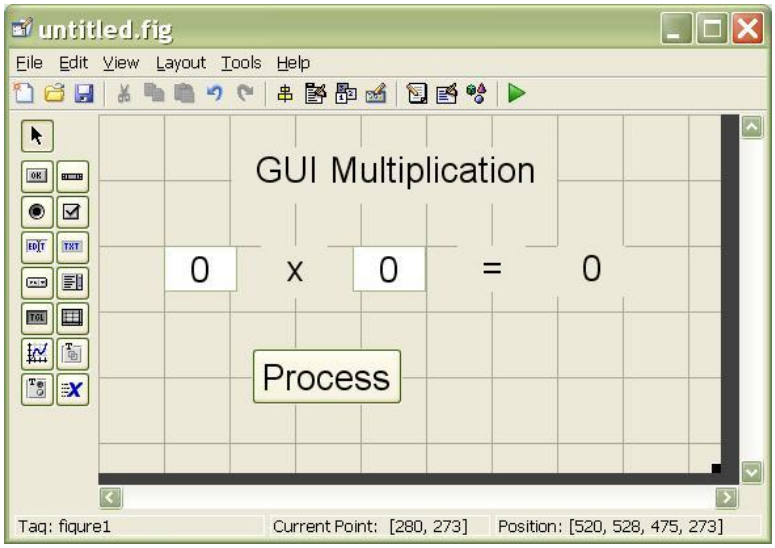


Ganti nilai String dengan "GUI Multiplication", kemudian ubah FontSize dari 8 menjadi 20. Lakukan langkah yang sama untuk Static Text kedua dan ketiga. Ganti nilai String pada Static Text kedua dengan "x" dan nilai String pada Static Text ketiga dengan "=".

Untuk Static Text keempat, ganti nilai String dengan "0", kemudian ganti nilai parameter Tag dengan "answer", komponen ini nanti akan digunakan untuk menampilkan hasil perkalian. Setelah dimodifikasi, hasilnya sebagai berikut :



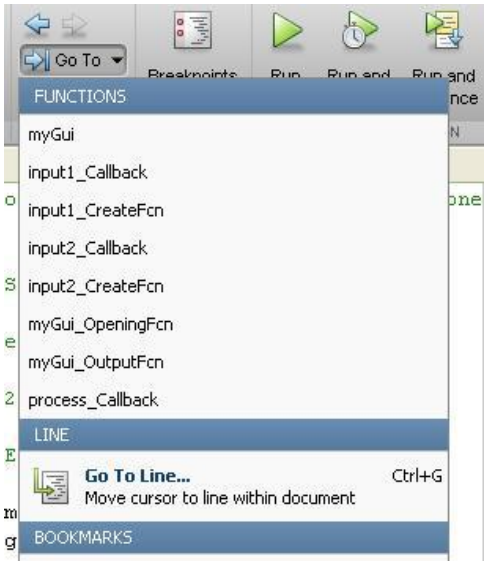
3. Langkah selanjutnya, kita perlu memodifikasi Edit Text dan juga Push Button. Double click pada Edit Text pertama, kemudian ganti nilai parameter FontSize menjadi 20, nilai String menjadi "0", dan nilai Tag diganti "input1". Lakukan hal yang sama untuk Edit Text yang kedua, yang berbeda hanyalah nilai Tag yaitu diganti "input2". Kemudian untuk Push Button, ganti nilai parameter FontSize menjadi 20, nilai String menjadi "Process", dan nilai Tag diganti "process". Jika nanti user menekan/mengklik tombol ini maka akan dilakukan proses perkalian. Setelah modifikasi selesai, hasilnya sebagai berikut :



4. Sekarang simpan GUI Anda dengan nama yang diinginkan, misalnya myGui. Ketika Anda menyimpan file ini, secara otomatis MATLAB akan menggenerate dua buah file : myGui.fig dan myGui.m. File .fig berisi grafik dari interface yang tadi telah dibuat, sedangkan file .m berisi semua code untuk bagian GUI.

5.3 Menuliskan Kode untuk GUI Callbacks

1. Buka file .m yang tergenerate secara otomatis ketika Anda menyimpan GUI. Pada MATLAB editor, klik pada icon "Go To", akan ada list dari fungsi-fungsi yang ada pada file .m. Pilih *input1_Callback*.



2. Cursor secara otomatis akan membawa Anda pada blok kode berikut ini :

```
function input1_Callback(hObject, eventdata, handles)
% hObject    handle to input1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of input1 as text
%        str2double(get(hObject,'String')) returns contents of input1 as a double
```

Tambahkan kode berikut ini di bawahnya :

```
%store the contents of input1 as a string. if the string
%is not a number then input will be empty
input = str2num(get(hObject,'String'));
%checks to see if input is empty. if so, default input1 to zero
if (isempty(input))
     set(hObject,'String','0')
end
guidata(hObject, handles);
```

Kode tersebut bertujuan untuk memastikan inputnya valid, yaitu berupa bilangan. Baris terakhir dari kode bertujuan untuk memberitahu GUI agar mengupdate *handles structure* setelah callback selesai.

3. Tambahkan kode yang sama untuk fungsi `input2_Callback`.
4. Sekarang kita perlu mengubah fungsi `process_Callback`. Anda akan melihat blok kode berikut ini pada file .m.

```
function process_Callback(hObject, eventdata, handles)
% hObject    handle to process (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

Tambahkan kode berikut ini di bawahnya :

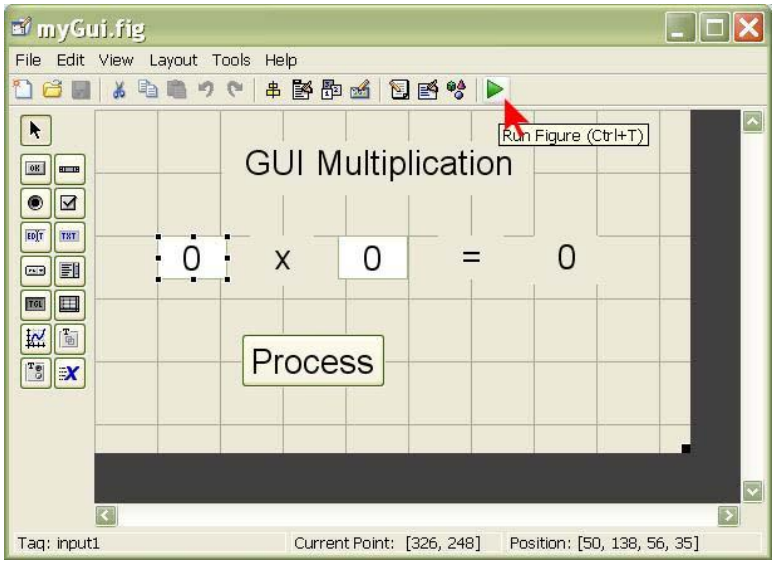
```
a = get(handles.input1,'String');
b = get(handles.input2,'String');
% a and b are variables of Strings type, and need to be converted
% to variables of Number type before they can be multiplied together
result = str2num(a) * str2num(b);
c = num2str(result);
% need to convert the answer back into String type to display it
set(handles.answer,'String',c);
guidata(hObject, handles);
```

5. Congratulations, we're finished coding the GUI!

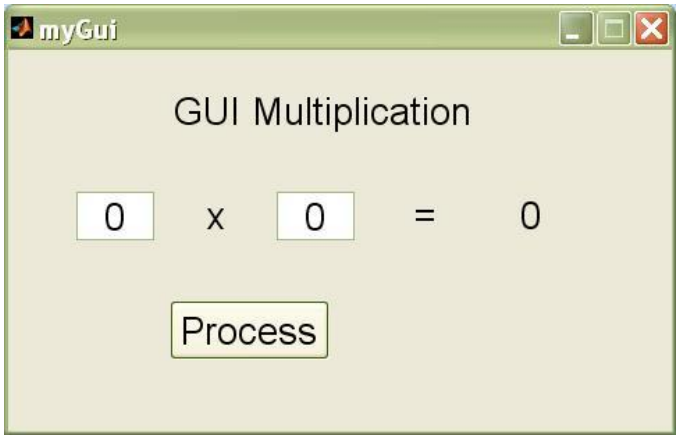
5.4 Menjalankan GUI

Terdapat dua buah cara untuk menjalankan GUI :

1. Menjalankan GUI dari Command Window MATLAB. Ketikkan perintah berikut ini :
   `>>myGui`
2. Melalui GUIDE editor atau M-File editor. Pilih icon  pada GUIDE seperti diperlihatkan gambar berikut ini :



Selanjutnya akan muncul window baru yang merupakan GUI yang sudah Anda buat.



Cobalah untuk memasukkan beberapa bilangan untuk mengetes GUI yang sudah Anda buat! ☺

Modul 6
Topik Pilihan : Membuat Mini-Photoshop

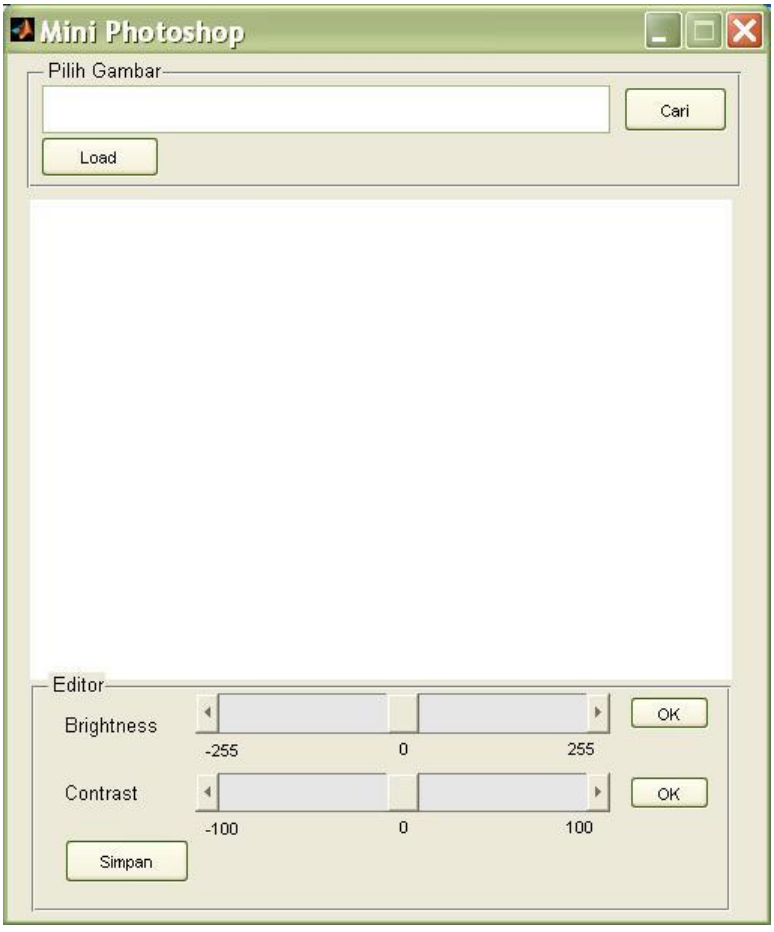
6.1 Desain Fungsional Aplikasi

Sebelum membuat sebuah aplikasi, ada baiknya kita mendefinisikan terlebih dahulu fungsi-fungsi yang dapat dilakukan oleh aplikasi. Pada aplikasi Mini Photoshop kali ini, kita akan mencoba untuk meniru software Adobe Photoshop yang sering digunakan untuk editing gambar, tentunya fungsi-fungsi yang dapat dijalankan masih sangat terbatas yaitu sebagai berikut:

1. Melakukan browsing data gambar yang akan diedit (gambar berekstensi .jpg).
2. Menampilkan data gambar yang dipilih.
3. Mengubah brightness gambar.
4. Mengubah contrast gambar.
5. Menyimpan gambar yang telah diedit.

6.2 Membuat Layout Aplikasi

Berikut ini contoh layout dari aplikasi yang akan dibuat :



Pada aplikasi terdapat 3 bagian bagian utama, yaitu :

- Panel Pilih Gambar, digunakan untuk memilih gambar.
- Canvas, digunakan untuk menampilkan gambar.
- Panel Editor, digunakan untuk melakukan editing brightness dan contrast gambar. Untuk mengatur nilai brightness dan contrast digunakan Slider.

Buatlah tampilan/layout aplikasi sesuai dengan contoh di atas!

6.3 Coding

1. Browsing Data Gambar

Untuk melakukan browsing data, kita akan menggunakan tombol "Cari". Jika tombol tersebut diklik maka akan muncul window baru untuk melakukan browsing data (berekstensi .jpg), contohnya sebagai berikut :

Untuk lebih memudahkan dalam pengkodean selanjutnya, ada baiknya kita menyamakan setting pada bagian GUI terlebih dahulu.

- Ganti nilai Tag pada Edit Text menjadi "namafile".
- Ganti nilai Tag pada Push Button Cari menjadi "cari".

Selanjutnya, pada bagian Inspector untuk Push Button Cari, klik simbol link yang ada di bagian Callback, pointer akan mengarah pada bagian kode program berikut ini :

```
function cari_Callback(hObject, eventdata, handles)
% hObject    handle to cari (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

Tambahkan kode berikut ini di bawahnya :

```
%browse file
[filename,pathname] = uigetfile('*.jpg','Pilih File');
%cek file name
if isequal([filename,pathname],[0,0])
    return
```

```
else
    fullpath = fullfile(pathname,filename);
    handles.gui.fullpath = fullpath;
    handles.gui.filename = filename;
    handles.gui.pathname = pathname;
    %tuliskan nama file pada bagian edit text
    set(handles.namafile,'String',handles.gui.fullpath);
    guidata(hObject, handles);
end
```

Setelah selesai menambahkan kode di atas, coba jalankan program GUI Anda untuk mengecek apakah fungsionalitas browsing data gambar berjalan dengan benar.

2. Menampilkan Data Gambar

Jika user mengklik tombol "Load", maka gambar yang tadi telah dibrowse akan ditampilkan di bagian Canvas. Berikut ini setting yang harus diubah untuk komponen GUI :

- Ganti nilai Tag pada Push Button Load menjadi "load".

Selanjutnya, pada bagian Inspector untuk Push Button Load, klik simbol link yang ada di bagian Callback, pointer akan mengarah pada bagian kode program berikut ini :

```
function load_Callback(hObject, eventdata, handles)
% hObject    handle to load (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

Tambahkan kode berikut ini di bawahnya :

```
% tampilkan citra
addpath(handles.gui.pathname);
I = imread(handles.gui.filename);
handles.gui.I = I;
imshow(I);
guidata(hObject, handles);
```

3. Mengubah Brightness

Editing brightness pada gambar dilakukan dengan melakukan penggeseran pada komponen Slider kemudian mengklik tombol OK untuk melakukan konfirmasi pengubahan. Berikut ini setting yang harus diubah untuk komponen GUI :

- Ganti nilai Tag pada Slider brightness menjadi "brightness". Kemudian nilai Max diganti menjadi 255, nilai Min diganti menjadi -255.
- Ganti nilai Tag pada Push Button OK (bagian brightness) menjadi "bright_ok".

Perlu diperhatikan disini bahwa jika kita melakukan perubahan nilai pixel pada sebuah gambar RGB, kita harus memastikan bahwa nilai pixelnya tetap berada di range 0-255. Maka dari itu, kita perlu membuat sebuah fungsi untuk membatasi nilai pixel yang sudah diedit.

```
function [y] = bound(x)
%++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
%fungsi : membatasi nilai dari pixel rgb
%++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
if x > 255
    y = 255;
elseif x < 0
    y = 0;
else
    y = x;
end;
```

Simpan dengan nama bound.m.

Langkah berikutnya, pada bagian Inspector untuk Slider brightness, klik simbol link yang ada di bagian Callback, pointer akan mengarah pada bagian kode program berikut ini :

```
function brightness_Callback(hObject, eventdata, handles)
% hObject    handle to brightness (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider
```

Tambahkan kode berikut ini di bawahnya :

```
% get slider value
brightness_value = get(hObject,'Value');
% process
I_edit = handles.gui.I;
bright_I(:,:,1) = uint8( bound( double(I_edit(:,:,1)) + brightness_value ) ) ;
bright_I(:,:,2) = uint8( bound( double(I_edit(:,:,2)) + brightness_value ) ) ;
bright_I(:,:,3) = uint8( bound( double(I_edit(:,:,3)) + brightness_value ) ) ;
handles.gui.bright_I = bright_I;
imshow(bright_I);
guidata(hObject, handles);
```

Selanjutnya, pada bagian Inspector untuk Push Button OK, klik simbol link yang ada di bagian Callback, pointer akan mengarah pada bagian kode program berikut ini :

```
function bright_ok_Callback(hObject, eventdata, handles)
% hObject    handle to bright_ok (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

Tambahkan kode berikut ini di bawahnya :

```
set(handles.brightness,'Value',0);
handles.gui.I = handles.gui.bright_I;
guidata(hObject, handles);
```

4. Mengubah Contrast

Editing contrast pada gambar hampir sama dengan editing brightness, dilakukan dengan melakukan penggeseran pada komponen Slider kemudian mengklik tombol OK untuk melakukan konfirmasi pengubahan. Berikut ini setting yang harus diubah untuk komponen GUI :

- Ganti nilai Tag pada Slider contrast menjadi “contrast”. Kemudian nilai Max diganti menjadi 100, nilai Min diganti menjadi -100.
- Ganti nilai Tag pada Push Button OK (bagian contrast) menjadi “contrast_ok”.

Langkah berikutnya, pada bagian Inspector untuk Slider contrast, klik simbol link yang ada di bagian Callback, pointer akan mengarah pada bagian kode program berikut ini :

```
function contrast_Callback(hObject, eventdata, handles)
% hObject    handle to contrast (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider
```

Tambahkan kode berikut ini di bawahnya :

```
% get slider value
contrast_value = get(hObject,'Value');
% process
if contrast_value ~= 0
    cv = contrast_value / 2;
    contrast_value = 127 ^ (cv / 100);
    I_edit = handles.gui.I;
    contrast_I(:,:,1) = uint8( bound( round( (double(I_edit(:,:,1))-
                        128)*contrast_value + 128 ) ) );
    contrast_I(:,:,2) = uint8( bound( round( (double(I_edit(:,:,2))-
                        128)*contrast_value + 128 ) ) );
    contrast_I(:,:,3) = uint8( bound( round( (double(I_edit(:,:,3))-
                        128)*contrast_value + 128 ) ) );
end;
handles.gui.contrast_I = contrast_I;
imshow(contrast_I);
guidata(hObject, handles);
```

Selanjutnya, pada bagian Inspector untuk Push Button OK, klik simbol link yang ada di bagian Callback, pointer akan mengarah pada bagian kode program berikut ini :

```
function contrast_ok_Callback(hObject, eventdata, handles)
% hObject    handle to contrast_ok (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

Tambahkan kode berikut ini di bawahnya :

```
set(handles.contrast,'Value',0);
handles.gui.I = handles.gui.contrast_I;
guidata(hObject, handles);
```

5. Menyimpan Gambar

Langkah terakhir adalah menyimpan gambar hasil editing. Kita dapat memilih di directory mana kita akan menyimpan gambar dan memberi nama gambar yang baru. Berikut ini setting yang harus diubah untuk komponen GUI :

- Ganti nilai Tag pada Push Button Simpan menjadi “simpan”.

Selanjutnya, pada bagian Inspector untuk Push Button Simpan, klik simbol link yang ada di bagian Callback, pointer akan mengarah pada bagian kode program berikut ini :

```
function simpan_Callback(hObject, eventdata, handles)
% hObject    handle to simpan (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

Tambahkan kode berikut ini di bawahnya :

```
% command simpan file
[FileName, PathName] = uiputfile('*.jpg', 'Save As');
if PathName==0
    return;
end
Name = fullfile(PathName, FileName);
imwrite(handles.gui.I, Name, 'jpg');
guidata(hObject, handles);
```

Selamat, Anda sudah selesai melakukan coding untuk aplikasi Mini-Photoshop! ☺

6.4 Menjalankan Aplikasi

Berikut ini hasil akhir dari aplikasi yang sudah kita buat :

(Gambar pemandangan sebelum diedit)

(Gambar pemandangan setelah diedit)

Untuk mengasah kreativitas / mengisi waktu luang, Anda bisa menambahkan sendiri fungsionalitas pada aplikasi Mini-Photoshop yang sudah dibuat. Misalnya dengan menambahkan fungsi zoom in, zoom out, rotasi gambar, atau edge detection. Selamat mencoba!

Daftar Pustaka

Gunaidi Abdia Away. 2010. “The Shortcut of MATLAB Programming”. Informatika Bandung.
MATLAB Guide R2012b. The MathWorks, Inc.
http://stackoverflow.com