**MAKALAH GUI APLIKASI KALKULATOR MATLAB**

****

Oleh :

Faiz Daffa Kusuma Laksana (1706700054)

**SEKOLAH TINGGI MANAJEMEN DAN INFORMATIKA KOMPUTER**

**SUBANG**

**2020**

BAB I

## Pendahuluan

Apa itu Matlab? Apa yang dapat dilakukan oleh Matlab? Kemampuan apa yang dimilikinya? Bagaimana kita menggunakan Matlab untuk menyelesaikan permasalahan-permasalahan komputasi? Mungkin kalimat tersebut adalah beberapa pertanyaan awal bagi mahasiswa yang baru pertama mengenal Matlab. Namun bagi mahasiswa yang sudah mengenal Matlab pertanyaan di atas bukanlah hal yang baru dan tentu saja seorang mahasiswa dapat menjawabnya dengan mudah.

Matlab dapat dipandang sebagai sebuah kalkulator dengan fitur yang lengkap.Kita pernah menggunakan kalkulator dengan fasilitas minimal, misalnya hanya terdapat fasilitas penambahan, pengurangan perkalian dan pembagian. Kalkulator yang lebih lengkap lagi adalah kalkulator *scientific* dimana fasilitas yang diberikan tidak hanya yang disebutkan di atas, melainkan sudah ada fungsi-fungsi trigonometri, bilangan kompleks, akar kuadrat dan logaritma.Nah, Matlab mirip dengan kalkulator tersebut, tetapi dengan fitur-fitur yang lengkap diantaranya dapat digunakan untuk memprogram aplikasi berbasis GUI (Graphical User Interface) dan lengkap dengan *toolbox* yang dapat dimanfaatkan untuk memecahkan masalah sains dan teknik.

Matlab (Matrix Laboratory) adalah sebuah program untuk analisis dan komputasi numerik dan merupakan suatu bahasa pemrograman matematika lanjutan yang dibentuk dengan dasar pemikiran menggunkan sifat dan bentuk matriks.Pada awalnya, program ini merupakan *interface*untuk koleksi rutin numerik dari proyek LINPACK dan EISPACK.

Dan dikembangkan menggunakan bahasa FORTRAN, namun sekarang merupakan produk komersial dari perusahaan Mathworks, Inc.yang dalam perkembangan selanjutnya dikembangkan menggunakan bahasa C++ dan assembler (utamanya untuk fungsi-fungsi dasar Matlab).

Matlab telah berkembang menjadi sebuah *environment*pemrograman yang canggih yang berisi fungsi-fungsi *built-in*untuk melakukan tugas pengolahan sinyal, aljabar linier, dan kalkulasi matematis lainnya. Matlab juga berisi *toolbox*yang berisi fungsi-fungsi tambahan untuk aplikasi khusus. Matlab bersifat *extensible*, dalam arti bahwa seorang pengguna dapat menulis fungsi baru untuk ditambahkan pada *library*ketika fungsi-fungsi *built-in*yang tersedia tidak dapat melakukan tugas tertentu. Kemampuan pemrograman yang dibutuhkan tidak terlalu sulit bila kita telah memiliki pengalaman dalam pemrograman bahasa lain seperti C, PASCAL, atau FORTRAN. Pada percobaan ini akan dibahas mengenai Pemrograman Matlab dan beberapa hal yang harus dipahami untuk menjalankan Matlab

## Tujuan

1. Untuk mengetahui pengertian dari Matlab
2. Untuk mengetahui GUI Matlab
3. Membantu menghitung dengan cepat

Bab II

Tinjauan Pustaka

## Dasar Teori

Matlab merupakan merek *software* yang dikembangkan oleh Mathworks.Yang merupakan *software* paling efisien untuk perhitungan numerik berbasis matriks.Dengan demikian jika di dalam perhitungan kita dapat menformulasikan masalah ke dalam format matriks maka matlab merupakan *software* terbaik untuk penyelesaian numeriknya.Matlab (Matrix Laboratory) yang merupakan bahasa pemrograman tingkat tinggi berbasis pada matriks sering digunakan untuk teknik komputasi numerik, yang digunakan untuk menyelesaikan masalah-masalah yang melibatkan operasi matematika elemen, matrik, optimasi, aproksimasi dll. Sehingga matlab banyak digunakan pada :

·         Matematika dan Komputansi

·         Pengembangan dan Algoritma

·         Pemrograman modeling, simulasi, dan pembuatan prototipe

·         Analisa data, eksplorasi dan visualisasi

·         Analisis numerik dan statistik

·         Pengembangan aplikasi teknik

Matlab adalah sistem interaktif dengan elemen dasar array yang merupakan basis datanya.Array tersebut tidak perlu dinyatakan khusus seperti di bahasa pemograman yang ada sekarang. Hal ini memungkinkan kita untuk memecahkan banyak masalah perhitungan teknik, khususnya melibatkan matriks dan vektor dengan waktu yang lebih singkat dari waktu yang dibutuhkan untuk menulis program dalam bahasa C atau FORTRAN.

Untuk memahami Matlab, terlebih dahulu kita harus paham mengenai matematika terutama operasi vektor dan matriks, karena operasi matriks merupakan inti utama dari matlab.Pada intinya matlab merupakan sekumpulan fungsi-fungsi yang dapat dipanggil dan dieksekusi. Fungsi-fungsi tersebut dibagi-bagi berdasarkan kegunaannya yang dikelompokan di dalam *toolbox* yang ada pada matlab (Crowell, 2005).

Nama matlab merupakan singkatan dari *matrix laboratory*.Dalam lingkungan perguruan tinggi teknik, Matlab merupakan perangkat standar untuk memperkenalkan dan mengembangkan penyajian materi matematika, rekayasa dan kelimuan.Di industri, matlab merupakan perangkat pilihan untuk penelitian dengan produktifitas yang tingi, pengembangan dan analisanya. Fitur-fitur matlab sudah banyak dikembangkan, dan lebih kita kenal dengan nama *toolbox*. Sangat penting bagi seorang pengguna matlab, *toolbox*mana yang mandukung untuk *learn*dan *apply technology* yang sedang dipelajarinya. *Toolbox*ini merupakan kumpulan dari fungsi-fungsi matlab (M-files) yang telah dikembangkan ke suatu lingkungan kerja matlab untuk memecahkan masalah dalam kelas partikular. Area-area yang sudah bisa dipecahkan dengan *toolbox* saat ini meliputi pengolahan sinyal, sistem kontrol, *neuralnetworks, fuzzy logic, wavelets*, dan lain-lain. Sebagai sebuah sistem, matlab tersusun dari 5 bagian utama:

**1.**      **Development Environment,**merupakan sekumpulan perangkat dan fasilitas yang membantu kita untuk menggunakan fungsi-fungsi dan file-file matlab. Beberapa perangkat ini merupakan sebuah Graphical User Interfaces (GUI). Termasuk didalamnya adalah matlab desktop dan *Command Window*, *Command History*, sebuah editor dan *debugger*, dan *browsers* untuk melihat *help*, *workspace, files*, dan *search path.*

**2.**      **Matlab Mathematical Function Library,**merupakan sekumpulan algoritma komputasi mulai dari fungsi-fungsi dasar sepertri sum, sin, cos, dan*complex arithmetic,*sampai dengan fungsi-fungsi yang lebih komplek seperti *Matrix Inverse*, *Matrix Eigenvalues, Bessel Functions,*dan *Fast Fourier Transforms.*

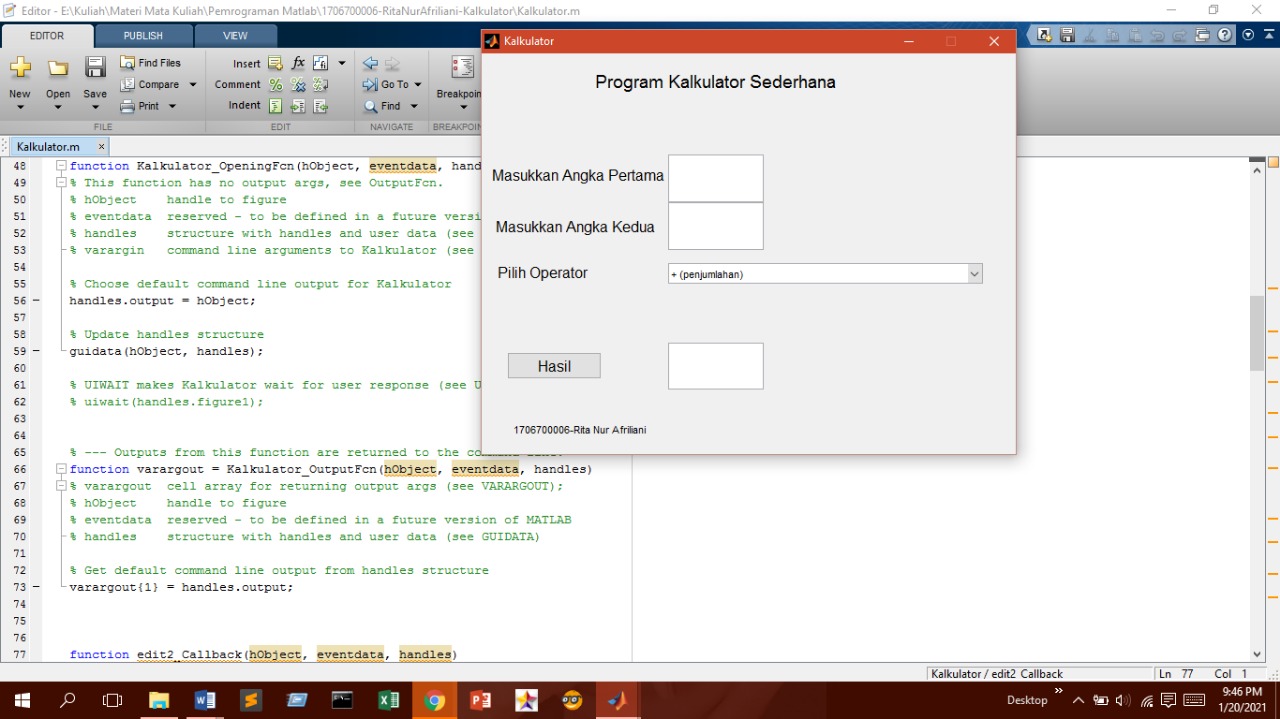
**3.**      **Matlab Language**, merupakan suatu level tinggibahasa matrix atau array dengan *control flow statements, functions,*data*structures, input/output,*danfitur-fitur*object-oriented programming*. Ini memungkinkan bagi kita untuk melakukan kedua hal baikpemrograman dalam lingkup sederhanauntuk mendapatkan hasil yang cepat, dan pemrograman dalam lingkup yang lebih besar untuk memperoleh hasil-hasil dan aplikasi yang komplek.

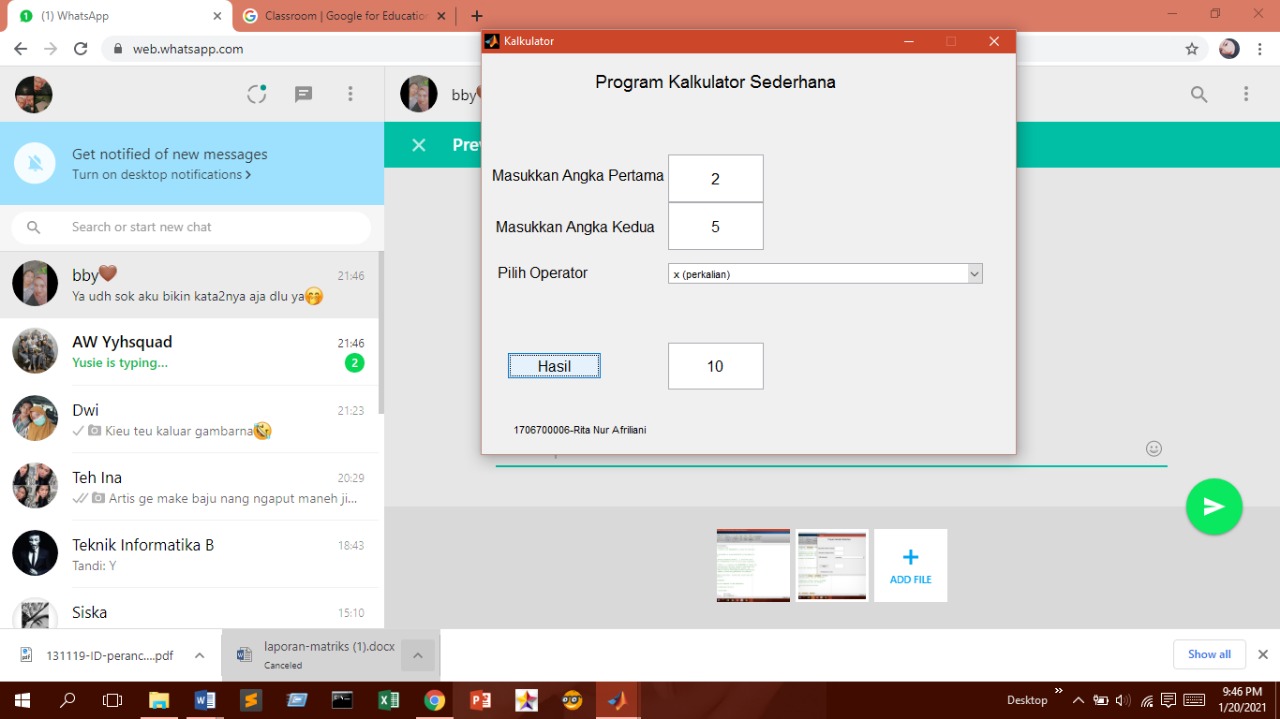
**4.**      **Graphics**, matlab memiliki fasilitas untuk menampilkan vektor dan matrik sebagai suatu grafik. Didalamnya melibatkan *high-level functions*(fungsi-fungsi level tinggi) untuk visualisasi data dua dimensi dan tiga dimensi, *image processing, animation*, dan*presentation graphics*. Ini juga melibatkan fungsi level rendah yang memungkinkan bagi kita untuk membiasakan diri untuk memunculkan grafik mulai dari bentuk yang sederhana sampai dengan tingkatan *graphical user interfaces* pada aplikasi matlab.

**5.**      **Matlab Application Program Interface (API),**merupakan suatu *library*yang memungkinkan program yang telah kita tulis dalam bahasa C dan Fortranmampu berinterakasi dengan matlab. Ini melibatkan fasilitas untuk pemanggilan *routines*dari matlab (dynamic linking), pemanggilan matlab sebagai sebuah *computational engine*, dan untuk membaca dan menuliskanMat*-files*(Miller, F.J.R., 1989).

## Tampilan Aplikasi dan Source Code

1. Tampilan Memasukan Angka Kalkulator



1. Tampilan Hasil
2. Source Code

function varargout = Kalkulator(varargin)

% KALKULATOR MATLAB code for Kalkulator.fig

% KALKULATOR, by itself, creates a new KALKULATOR or raises the existing

% singleton\*.

%

% H = KALKULATOR returns the handle to a new KALKULATOR or the handle to

% the existing singleton\*.

%

% KALKULATOR('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in KALKULATOR.M with the given input arguments.

%

% KALKULATOR('Property','Value',...) creates a new KALKULATOR or raises the

% existing singleton\*. Starting from the left, property value pairs are

% applied to the GUI before Kalkulator\_OpeningFcn gets called. An

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to Kalkulator\_OpeningFcn via varargin.

%

% \*See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one

% instance to run (singleton)".

%

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Kalkulator

% Last Modified by GUIDE v2.5 03-Mar-2020 16:02:21

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @Kalkulator\_OpeningFcn, ...

'gui\_OutputFcn', @Kalkulator\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% End initialization code - DO NOT EDIT

% --- Executes just before Kalkulator is made visible.

function Kalkulator\_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% varargin command line arguments to Kalkulator (see VARARGIN)

% Choose default command line output for Kalkulator

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes Kalkulator wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = Kalkulator\_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

varargout{1} = handles.output;

function edit2\_Callback(hObject, eventdata, handles)

% hObject handle to edit2 (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 edit2 as text

% str2double(get(hObject,'String')) returns contents of edit2 as a double

% --- Executes during object creation, after setting all properties.

function edit2\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function edit3\_Callback(hObject, eventdata, handles)

% hObject handle to edit3 (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 edit3 as text

% str2double(get(hObject,'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties.

function edit3\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

% --- Executes on button press in pushbutton1.

function pushbutton1\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

angka\_pertama = str2double(get(handles.edit1,'String'));

angka\_kedua = str2double(get(handles.edit2,'String'));

operator = get(handles.popupmenu1,'Value');

if operator == 1

hasil = angka\_pertama + angka\_kedua;

elseif operator == 2

hasil = angka\_pertama - angka\_kedua;

elseif operator == 3

hasil = angka\_pertama \* angka\_kedua;

elseif operator == 4

hasil = angka\_pertama / angka\_kedua;

end

set(handles.edit3,'String',num2str(hasil))

% --- Executes on selection change in popupmenu1.

function popupmenu1\_Callback(hObject, eventdata, handles)

% hObject handle to popupmenu1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as cell array

% contents{get(hObject,'Value')} returns selected item from popupmenu1

% --- Executes during object creation, after setting all properties.

function popupmenu1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to popupmenu1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function edit1\_Callback(hObject, eventdata, handles)

% hObject handle to edit1 (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 edit1 as text

% str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.

function edit1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to edit1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

BAB III

Kesimpulan

Matlab merupakan bahasa pemrograman yang hadir dengan fungsi dan karakteristik yang berbeda dengan bahasa pemrograman lain yang sudah ada lebih dahulu seperti Delphi, Basic maupun C++. Matlab merupakan bahasa pemrograman level tinggi yang dikhususkan untuk kebutuhan komputasi teknis, visualisasi dan pemrograman seperti komputasi matematik, analisis data, pengembangan     algoritma, simulasi dan pemodelan dan grafik-grafik perhitungan. Matlabadalah sebuah bahasa dengan (high-performance) kinerja tinggi untuk komputasi masalah teknik. Matlab mengintegrasikan komputasi, visualisasi, dan pemrograman dalam suatu model yang sangat mudah untuk pakai dimana masalah-masalah dan penyelesaiannya diekspresikan dalam notasi matematika yang familiar.