Aplikasi GUI Matlab untuk Menaksir Koefisien Parameter Model Regresi Non Linier Menggunakan Algoritma Levenberg-Marquardt

REKAYASA DATA

PENGUSUL

Atika Ratna Dewi, S.Si, M.Sc 0615128703

INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2020

1. Pendahuluan

Regresi non linier merupakan suatu analisis regresi dimana data penelitian digambarkan oleh suatu fungsi yang merupakan kombinasi non linier dari parameter-parameter atau dapat diartikan sebagai hubungan antara variabel independen (X) terhadap variabel dependen (Y) [1]. Penaksiran parameter pada regresi non linier dapat dilakukan dengan beberapa macam cara salah satunya menggunkan algoritma *Levenberg-Marquardt.*

Algoritma *Levenberg-Marquardt* dikembangkan pertama kali pada tahun 1963 untuk menyelesaikan masalah *nonlinear least square* [4]. Algoritma *Levenberg-Marquardt* merupakan gabungan antara algoritma *Gauss-Newton* dan algoritma *Steepest Descent*. Berdasarkan penelitian terdahulu menunjukkan bahwa algoritma Levenberg-Marquardt adalah algoritma yang cepat dan memiliki konvergensi yang stabil [3]. Metode *Levenberg-Marquardt* menggunakan metode algoritma seperti algoritma *Gauss-Newton* yaitu menggunakan *first order condition* (FOC) dari *sum of least square error* [4]. Perbedaannya adalah ada penambahan perkalian skalar dan matriks identitas μI_K pada algoritma *Levenberg-Marquardt*. Selain itu, penentuan panjang langkah atau *step length* (t_n) dalam algoritma *Levenberg-Marquardt* dapat bervariasi.Riset lanjutan tentang estimasi parameter juga telah banyak diteliti, salah satunya penelitian [5].

Penaksiran parameter β yang dilakukan secara manual akan cukup rumit dan memakan waktu yang lama. Untuk memudahkan penaksiran parameter β pada algoritma *Levenberg-Marquardt* dikembangkan aplikasi berbasis Matlab. Matlab merupakan salah satu software matematika yang dapat digunakan menyelesaikan berbagai persoalan matematika. Pengembangan aplikasi GUI (*Graphical User Interface*) pada Matlab hadir sebagai alat yang dapat memudahkan dan mempercepat pengguna matlab dalam menyelesaikan masalah matematika. Oleh karena itu akan dikembangkan aplikasi GUI Matlab guna menaksir parameter pada model regresi non linier menggunakan algoritma *Levenberg Marquardt*.

2. Deskripsi

2.1. Sistem Estimasi Parameter Model Regresi Non Linear menggunakan Algoritma Levenberg-Marquardt

Bentuk umum dari model regresi non linier adalah

$$y = f(X, \beta) + e, \qquad (1)$$

dengan fungsi non linier dalam parameter β dan $e \sim N(0, \sigma^2 I_T)$. Ada 2 cara untuk menaksir β pada model regresi non linier yaitu dengan metode *nonlinear least square* dan *maximum likelihood*. Kedua metode tersebut menghasilkan penaksiran β yaitu:

$$\hat{\beta} = f(X, \beta) + e. \qquad (2)$$

Penaksiran β dengan metode *nonlinear least square* bertujuan untuk mendapatkan nilai β yang meminimumkan residual *sum of squares* $S(\beta)$ [2].

$$\min_{\beta} S(\beta) = e'e, \qquad (3)$$

$$= \left(y - f(x, \beta)\right)' \left(y - f(x, \beta)\right). \qquad (4)$$

Syarat perlu untuk minimisasi adalah

$$\frac{\partial S}{\partial \beta} = -2\,[Z(\beta)]'\big(y - f(X,\beta)\big), \qquad (5)$$

$$[Z(\beta)]'(y - f(X,\beta)) = 0. \qquad (6)$$

Fungsi $f(X,\beta)$ dalam persamaan (6) adalah fungsi non linier sehingga penaksiran nilai β memerlukan proses iterasi yang memberikan global minimum. Secara umum, iterasi untuk mendapatkan taksiran β dengan *nonlinear least square* adalah

$$\beta^{(n+1)} = \beta^{n} + t_n P_n \gamma_n. \qquad (7)$$

Jenis iterasi yang dapat digunakan untuk mendapatkan taksiran β dengan *nonlinear least square* antara lain:

1. Algoritma *Gauss-Newton*
2. Algoritma *Steepest Descent*
3. Algoritma *Levenberg-Marquardt*

Aproksimasi $y = f(X,\beta)$ di sekitar *initial value* $\beta^{(1)}$ dilakukan dengan deret *Taylor* orde 1, yaitu

$$f(X,\beta) = f\big(X,\beta^{(1)}\big) + \frac{\partial f(X,\beta)}{\partial \beta'}\Big|_{\beta^{(1)}}\big(\beta - \beta^{(1)}\big), \qquad (3)$$

Misalkan $\frac{\partial f(X,\beta)}{\partial \beta'}\Big|_{\beta^{(1)}} = Z\big(\beta^{(1)}\big)$, maka:

$$y = f\big(X,\beta^{(1)}\big) + Z\big(\beta^{(1)}\big)\beta - Z\big(\beta^{(1)}\big)\beta^{(1)} + e, \qquad (4)$$

atau

$$y - f\big(X,\beta^{(1)}\big) + Z\big(\beta^{(1)}\big)\beta^{(1)} = Z\big(\beta^{(1)}\big)\beta + e,$$

$$\bar{y}\big(\beta^{(1)}\big) = Z\big(\beta^{(1)}\big)\beta + e. \qquad (5)$$

Nilai parameter β dapat ditaksir dengan menggunakan metode *least square*, diperoleh

$$\beta^{(2)} = \Big(Z\big(\beta^{(1)}\big)' Z\big(\beta^{(1)}\big)\Big)^{-1} Z\big(\beta^{(1)}\big)'^{\bar{y}}\big(\beta^{(1)}\big) \qquad (6)$$

Algoritma secara umum yang diperoleh adalah

$$\beta^{(n+1)} = \beta^{(n)} + \Big(Z\big(\beta^{(n)}\big)' Z\big(\beta^{(n)}\big)\Big)^{-1} Z(\beta^{(n)})' \Big(y - f\big(X,\beta^{(n)}\big)\Big). \qquad (7)$$

Persamaan (7) disebut sebagai persamaan algoritma *Gauss-Newton*, sedangkan persamaan algoritma *Levenberg-Marquardt* adalah persamaan yang diperoleh dengan cara memodifikasi persamaan (7) menjadi :

$$\beta^{(n+1)} = \beta^{(n)} + \Big(Z\big(\beta^{(n)}\big)' Z\big(\beta^{(n)}\big) + \mu I\Big)^{-1} Z(\beta^{(n)})' \Big(y - f\big(X,\beta^{(n)}\big)\Big). \qquad (8)$$

Persamaan (8) disebut sebagai algoritma *Levenberg-Marquardt*, dimana μ adalah *damping parameter* yang nilainya tidak boleh negatif dan biasanya nilai μ merupakan faktor dari 10. Sedangkan *I* adalah matriks identitas. algoritma *Levenberg-Marquardt* akan berhenti pada saat nilai algoritma tersebut konvergen yaitu jika memenuhi :

$$\|\beta^{n+1} - \beta^{n}\| \le \varepsilon. \qquad (9)$$

Koding dari sistem yang dibangun disajikan pada uraian di bawah ini.

2.2 Source Code

a. Main Source Code

```
function varargout = gui_levmar(varargin)
```

```
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @gui_levmar_OpeningFcn, ...
                   'gui_OutputFcn',  @gui_levmar_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 gui_levmar is made visible.
function gui_levmar_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 gui_levmar (see VARARGIN)

% Choose default command line output for gui_levmar
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes gui_levmar wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = gui_levmar_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 data_Callback(hObject, eventdata, handles)
% hObject    handle to data (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 data as text
%        str2double(get(hObject,'String')) returns contents of data as a
double


% --- Executes during object creation, after setting all properties.
function data_CreateFcn(hObject, eventdata, handles)
% hObject    handle to data (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 sheet_Callback(hObject, eventdata, handles)
% hObject    handle to sheet (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 sheet as text
%        str2double(get(hObject,'String')) returns contents of sheet as a
double


% --- Executes during object creation, after setting all properties.
function sheet_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sheet (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 rangey_Callback(hObject, eventdata, handles)
% hObject    handle to rangey (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 rangey as text
%        str2double(get(hObject,'String')) returns contents of rangey as a
double
```

```
% --- Executes during object creation, after setting all properties.
function rangey_CreateFcn(hObject, eventdata, handles)
% hObject    handle to rangey (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 rangex_Callback(hObject, eventdata, handles)
% hObject    handle to rangex (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 rangex as text
%        str2double(get(hObject,'String')) returns contents of rangex as a
double


% --- Executes during object creation, after setting all properties.
function rangex_CreateFcn(hObject, eventdata, handles)
% hObject    handle to rangex (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 nilaiawal1_Callback(hObject, eventdata, handles)
% hObject    handle to nilaiawal1 (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 nilaiawal1 as text
%        str2double(get(hObject,'String')) returns contents of nilaiawal1 as
a double


% --- Executes during object creation, after setting all properties.
function nilaiawal1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nilaiawal1 (see GCBO)
```

```matlab
% 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 nilaiawal2_Callback(hObject, eventdata, handles)
% hObject    handle to nilaiawal2 (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 nilaiawal2 as text
%        str2double(get(hObject,'String')) returns contents of nilaiawal2 as
a double


% --- Executes during object creation, after setting all properties.
function nilaiawal2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nilaiawal2 (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 nilaiawal3_Callback(hObject, eventdata, handles)
% hObject    handle to nilaiawal3 (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 nilaiawal3 as text
%        str2double(get(hObject,'String')) returns contents of nilaiawal3 as
a double


% --- Executes during object creation, after setting all properties.
function nilaiawal3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nilaiawal3 (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 hasil1_Callback(hObject, eventdata, handles)
% hObject    handle to hasil1 (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 hasil1 as text
%        str2double(get(hObject,'String')) returns contents of hasil1 as a
double


% --- Executes during object creation, after setting all properties.
function hasil1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hasil1 (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 hasil2_Callback(hObject, eventdata, handles)
% hObject    handle to hasil2 (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 hasil2 as text
%        str2double(get(hObject,'String')) returns contents of hasil2 as a
double


% --- Executes during object creation, after setting all properties.
function hasil2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hasil2 (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 hasil3_Callback(hObject, eventdata, handles)
% hObject    handle to hasil3 (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 hasil3 as text
%        str2double(get(hObject,'String')) returns contents of hasil3 as a
double


% --- Executes during object creation, after setting all properties.
function hasil3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hasil3 (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 sse_Callback(hObject, eventdata, handles)
% hObject    handle to sse (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 sse as text
%        str2double(get(hObject,'String')) returns contents of sse as a
double


% --- Executes during object creation, after setting all properties.
function sse_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sse (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 aic_Callback(hObject, eventdata, handles)
% hObject    handle to aic (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 aic as text
%        str2double(get(hObject,'String')) returns contents of aic as a
double


% --- Executes during object creation, after setting all properties.
function aic_CreateFcn(hObject, eventdata, handles)
% hObject    handle to aic (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 sc_Callback(hObject, eventdata, handles)
% hObject    handle to sc (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 sc as text
%        str2double(get(hObject,'String')) returns contents of sc as a double


% --- Executes during object creation, after setting all properties.
function sc_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sc (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 browsedata.
function browsedata_Callback(hObject, eventdata, handles)
% hObject    handle to browsedata (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
filename=get(handles.data,'string');
[data,pathname]=uigetfile({'*.xls','Microsoft Excel (.xls)';...
    '*.*','All Files (*.*)'},'Pick a file');
if data == 0
    set(handles.file_name,'string',filename);
else
    set(handles.data,'string',data);
    set(handles.sheet,'string','sheet1');
    set(handles.rangey,'string','C2:C71');
    set(handles.rangex,'string','A2:A71');
end


% --- Executes on button press in getdata.
function getdata_Callback(hObject, eventdata, handles)
% hObject    handle to getdata (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global y x
data=get(handles.data,'string');
sheet=get(handles.sheet,'string');
rangey=get(handles.rangey,'string');
rangex=get(handles.rangex,'string');
y=xlsread(data,sheet,rangey);
x=xlsread(data,sheet,rangex);

set(handles.nilaiawal1,'string',0.03451);
set(handles.nilaiawal2,'string',0.00021);
set(handles.nilaiawal3,'string',-0.01542);


% --- Executes on button press in estimasi.
function estimasi_Callback(hObject, eventdata, handles)
% hObject    handle to estimasi (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global y x
a1=str2double(get(handles.nilaiawal1,'string'));
a2=str2double(get(handles.nilaiawal2,'string'));
a3=str2double(get(handles.nilaiawal3,'string'));
tic;
b=[a1 a2 a3]';
%b=[0.0345 0.00021 -0.01542]'; %initial value data 1

% Data tahun ke 2
% y=gaji(:,4); %log gaji 2
% x=gaji(:,2); %usia 2
% tic;
% b=[0.0219 0.0259 -0.2253]'; %initial value data 2


% Levenberg-Marquardt Iterations
T=length(x);
rep = 70000 ;%the size of 'rep' depends on the convergence the initial value
k = length(b);
```

```
e = eye(k);
f = f2(b,x);
S = (y-f)'*(y-f);

j1 = 0;
j2 = 0;
tn = 50; % dapat melakukan perubahan tn
lamda = 0.1 ;% dapat melakukan perubahan nilai lamda

h=waitbar(0,'Waiting Hasil Estimasi...');
for i = 1:rep ;
      z = numgradf2(b,x);%Numerical gradient of f1
      zS = numgradS2(b,x,y) ;%Numerical gradient of S1
      step  = -0.5.*inv(z'*z + lamda*eye(k))*zS ; %  Marquardt-Levenberg
Iterations
      bnext = b + step;
      fnext = f2(bnext,x) ;
      Snext = (y-fnext)'*(y-fnext);


     while Snext < S && j1 <=100;
     step  = step*tn;  % Perubahan tn
     bnext = b+step;
     fnext = f2(bnext,x);
     Snext = (y-fnext)'*(y-fnext);
     j1    = j1+1;
     w1    = i;
     end;

   while Snext > S && j2 <=100;
      step  = step/tn;  % Perubahan tn
      bnext = b+step;
      fnext = f2(bnext,x);
      Snext = (y-fnext)'*(y-fnext);
      j2    = j2+1;
      w2    = i;
   end;

    if norm(bnext-b) <= 1e-9 && abs(S-Snext) <= 1e-9
       disp('Sudah konvergen. Dengan jumlah iterasinya adalah:') ;
       disp(i) ;
         break ;
    end ;

      if i == rep
         disp('Belum konvergen, iterasinya perlu ditambah lagi.') ;
         disp('Atau ubahlah initial values-nya') ;
         disp(' ') ;
      end ;

      b = bnext;
      f = f2(b,x) ;
      S = (y-f)'*(y-f);
waitbar((i/rep),h);
end ;
```

```
close(h)

p=length(y);
bK=bnext;
out=bK';
fin=f2(bK,x);
s2=(y-fin)'*(y-fin)/(p-k);
sse=S;

% Menentukan AIC dan SC
% Menggunakan file L2.m
LL    = L2(b,x,y);
AIC   = abs(-2*LL+2*k);
SC    = abs(-2*LL+log(T)*k);

set(handles.hasil1,'String',bnext(1));
set(handles.hasil2,'String',bnext(2));
set(handles.hasil3,'String',bnext(3));
set(handles.sse,'String',sse);
set(handles.aic,'String',AIC);
set(handles.sc,'String',SC);


% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes1


% --- Executes on button press in grafik.
function grafik_Callback(hObject, eventdata, handles)
% hObject    handle to grafik (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global x y
c1=str2double(get(handles.hasil1,'string'));
c2=str2double(get(handles.hasil2,'string'));
c3=str2double(get(handles.hasil3,'string'));
D=c1+(c2.*exp(-c3.*x));
plot(x,D);
hold on
plot (x,y,'m.');
axes(handles.axes2);
```

b. Source Code f2

```
%File f2.m
function f = f2(b,x)
t=x(:,1);
b1=b(1,:);
b2=b(2,:);
b3=b(3,:);
f=b1+(b2*exp(-b3*t));
end
```

c. Source Code Numgradf2

```
%File Numgradf2.m
function z = numgradf2(b,x)
% Numerical z (numerical gradient of
k = length(b);
d = 1e-7;
e = eye(k);

for j=1:k ;
    bplus = b + d*e(:,j);
    fplus = feval('f2',bplus,x) ;
    bmin  = b - d*e(:,j) ;
    fmin  = feval('f2',bmin,x) ;
    z(:,j)= (fplus - fmin)/(2*d);
end;
```

d. Source Code L2

```
%File L2.m
function LL = L2(b,x,y)
T   = length(x);
f   = f2(b,x);
s2 = ((y-f)'*(y-f))/T;
LL   = -0.5*(log (2*pi*s2) + (y-f)'*(y-f)/s2);
end
```

e. Source Code NumgradS2

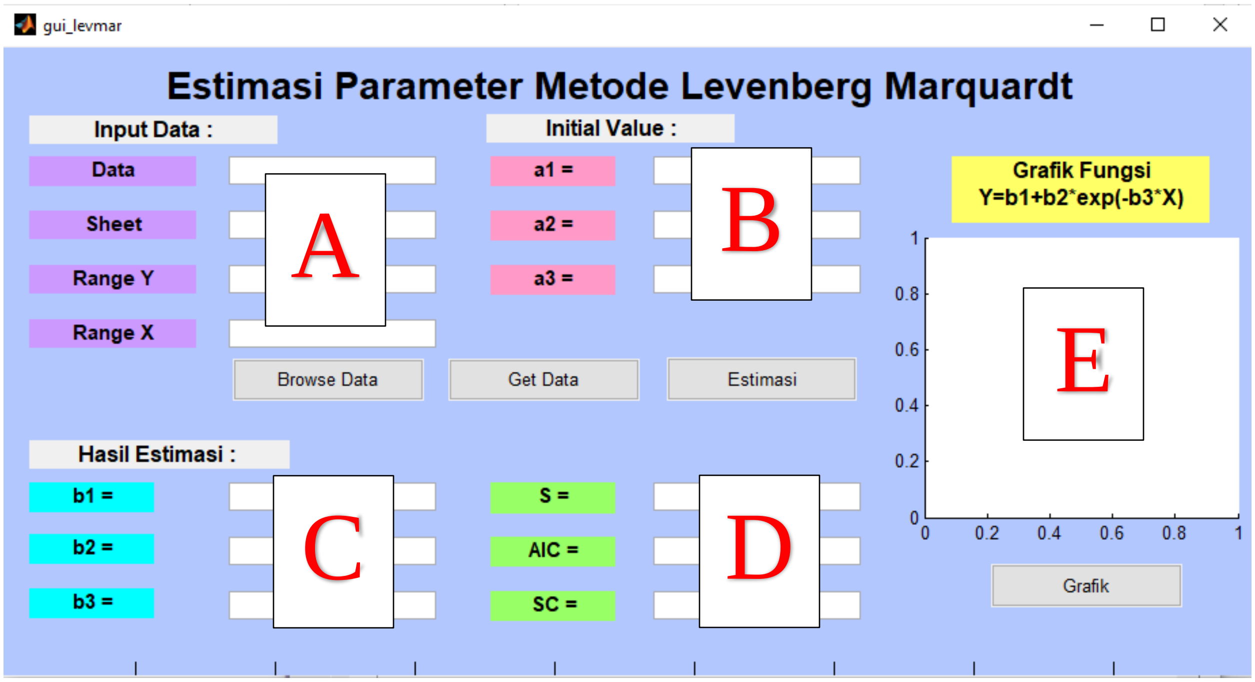
```
%File Numgrads2.m
function z = numgradS2(b,x,y)
% Numerical z (numerical gradient of L)
% Output berupa vector dengan dimensi Kx1

k = length(b);
d = 1e-6;
e = eye(k);

for j=1:k;% Numerical gradients
    bplus = b + d*e(:,j) ;
    fplus = feval('f2',bplus,x) ;
    Splus = (y-fplus)'*(y-fplus);
    bmin  = b - d*e(:,j) ;
    fmin  = feval('f2',bmin,x) ;
    Smin  = (y-fmin)'*(y-fmin);
    z(j,:) = (Splus - Smin)/(2*d);
end;
```

2.2. Hasil Visualisasi

Tampilan Estimasi Parameter Model Regresi Non Linier Metode Levenberg Marquardt ditunjukkan pada Gambar 1.



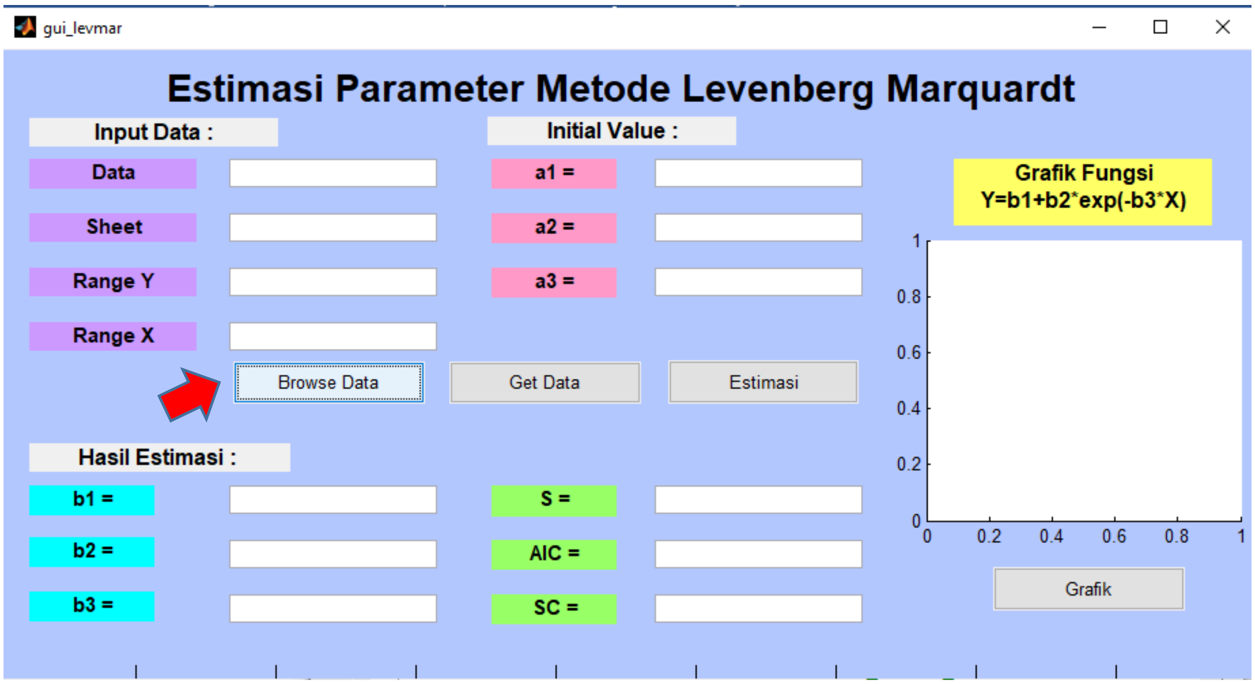
Deskripsi dari masing-masing komponen di dalam tampilan antara lain:

a. ***Browse Data*** merupakan *pushbutton* yang berfungsi untuk mengambil data yang pada folder tertentu untuk kemudian diproses dengan algoritma Levenberg-Marquardt.
b. ***Get Data*** merupakan *pushbutton* yang akan digunakan untuk mengambil nilai awal data (*initial value*).
c. ***Estimasi*** merupakan *pushbutton* untuk proses estimasi parameter model regresi non linear dengan algoritma Levenberg-Marquardt.
d. ***Grafik*** merupakan *pushbutton* untuk proses menampilkan grafik hasil estimasi parameter model regresi non linear dengan algoritma Levenberg-Marquardt.
e. **A** merupakan *blankspace* yang akan menampilkan data yang akan diestimasi.
f. **B** merupakan *blankspace* yang akan menampilkan nilai awal data (*initial value*).
g. **C** merupakan *blankspace* yang akan menampilkan hasil estimasi parameter model regresi non linear dengan algoritma Levenberg-Marquardt.
h. **D** merupakan *blankspace* yang akan menampilkan hasil nilai S, AIC dan SC.
i. **E** merupakan *blankspace* yang akan menampilkan hasil grafik hasil estimasi parameter model regresi non linear dengan algoritma Levenberg-Marquardt.

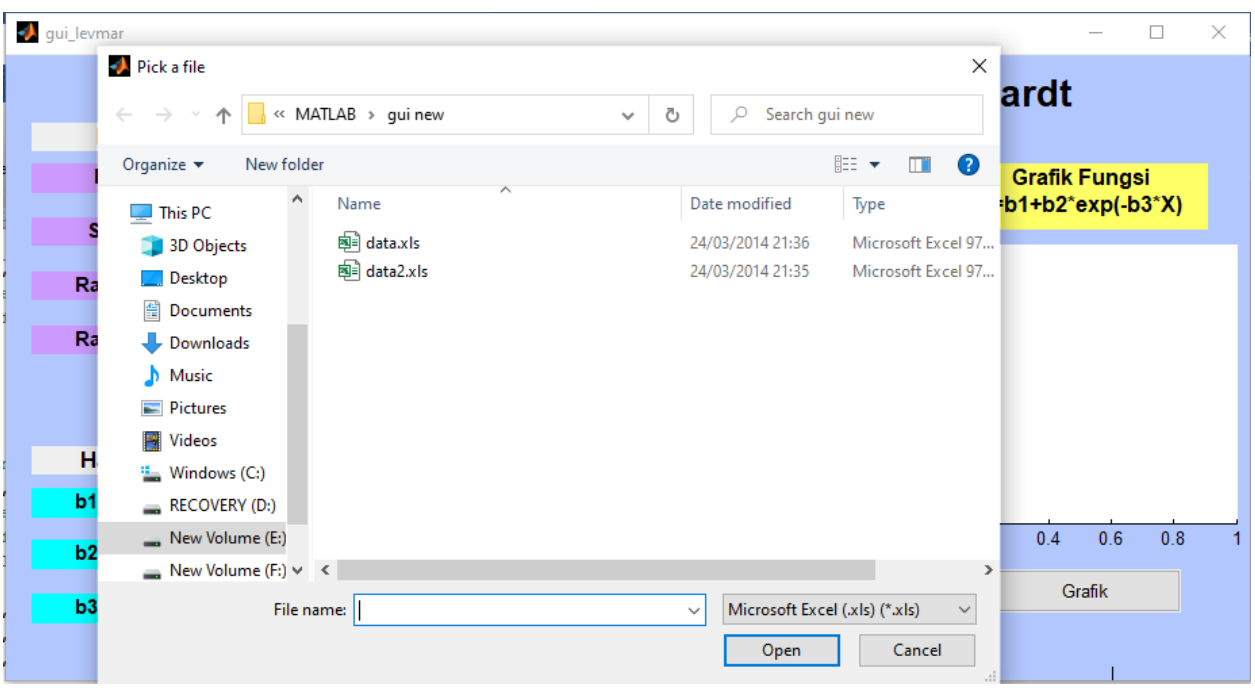
2.3. PROSES PENGGUNAAN GUI

Langkah-langkah dari penggunaan GUI sebagai berikut:

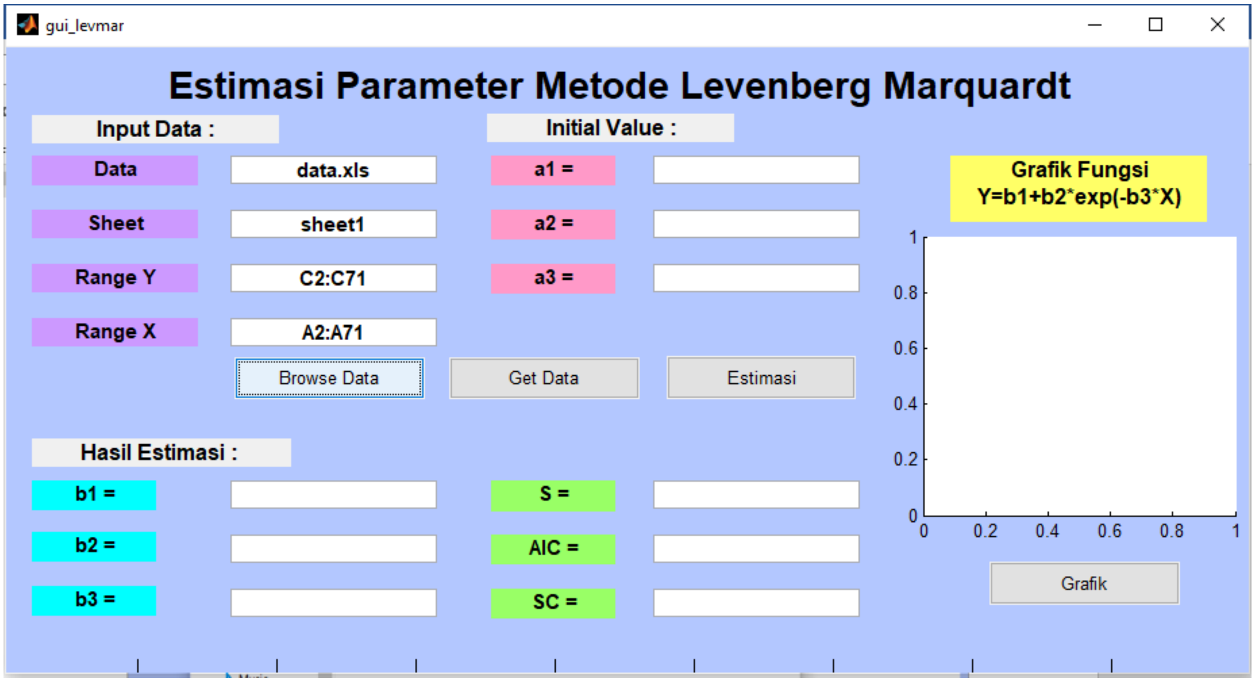
1. Mengambil data yang akan diestimasi dengan menekan tombol **BROWSE DATA**.



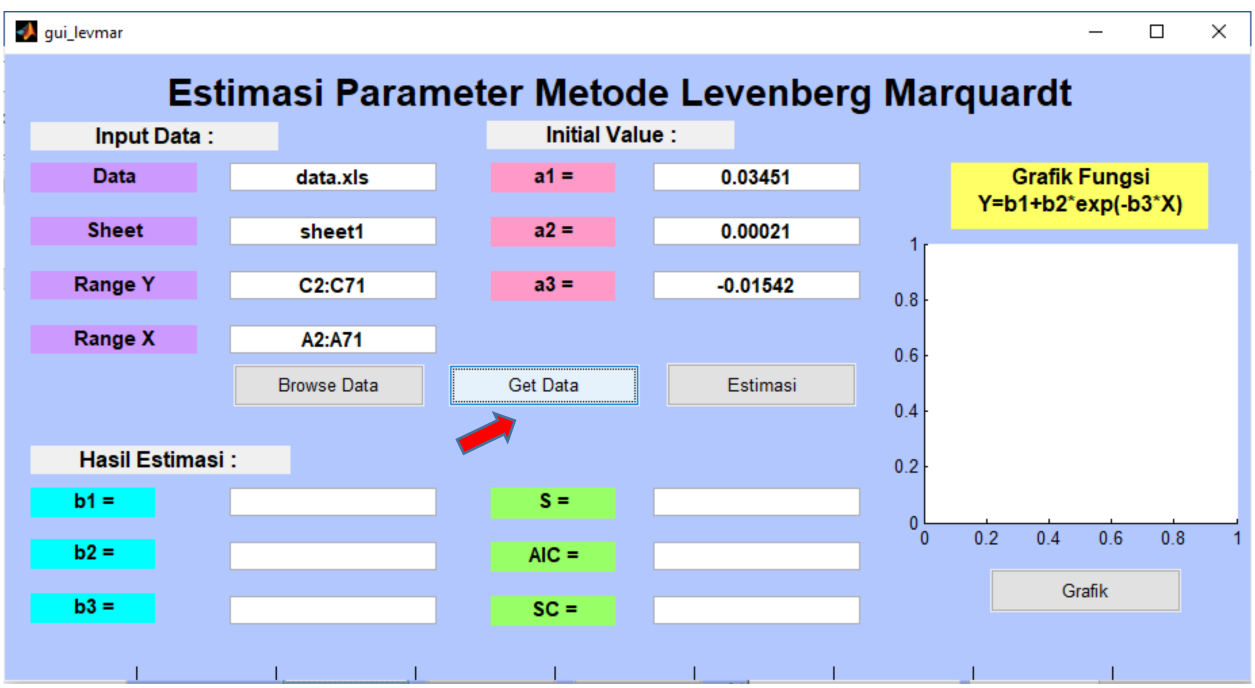
2. Setelah menekan tombol Browse Data maka akan muncul tampilan sebagai berikut dan kemudian pilih file yang akan digunakan pada proses estimasi :



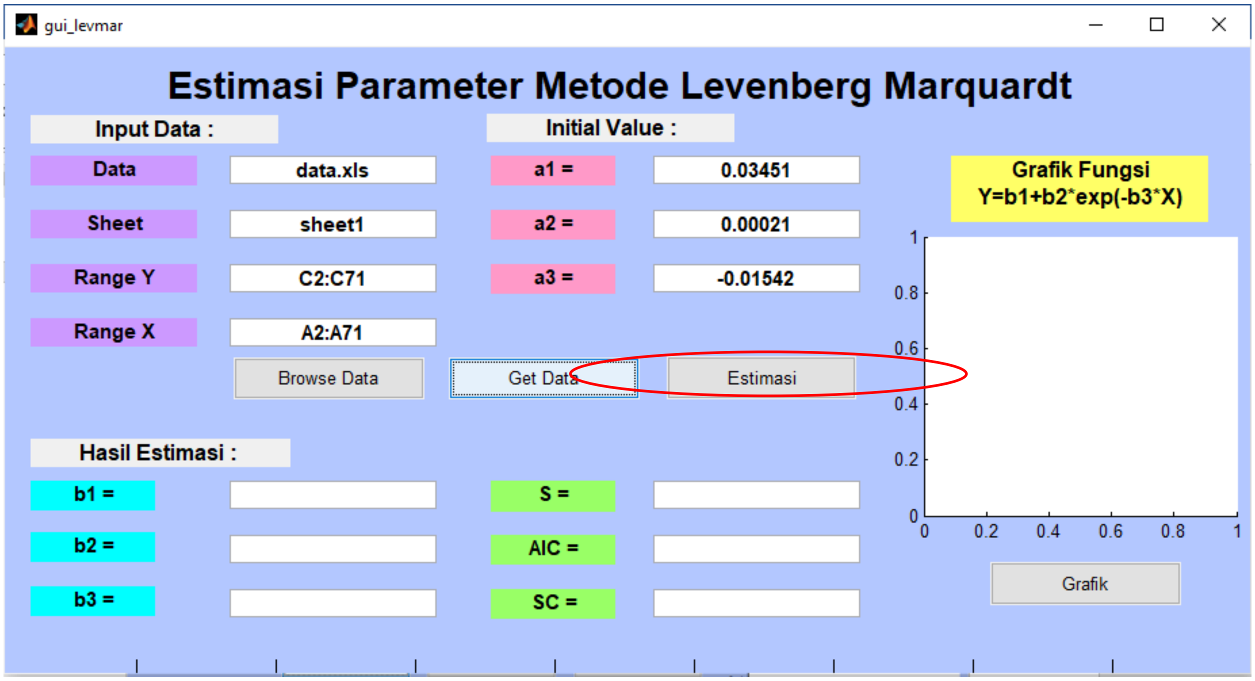
3. Data yang dipilih akan ditampilkan pada *blankspace* yang telah disediakan seperti gambar berikut :

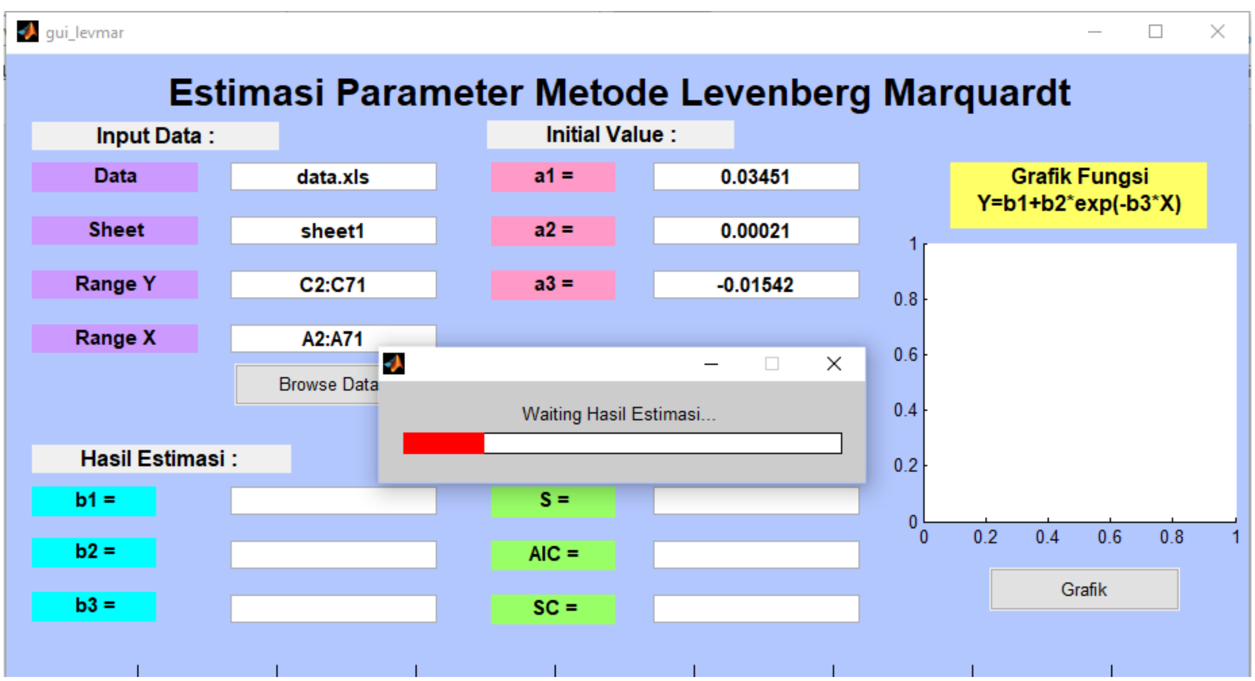


4. Menampilkan nilai awal data (*initial value*) yang akan digunakan untuk estimasi parameter dengan menekan tombol **GET DATA**.

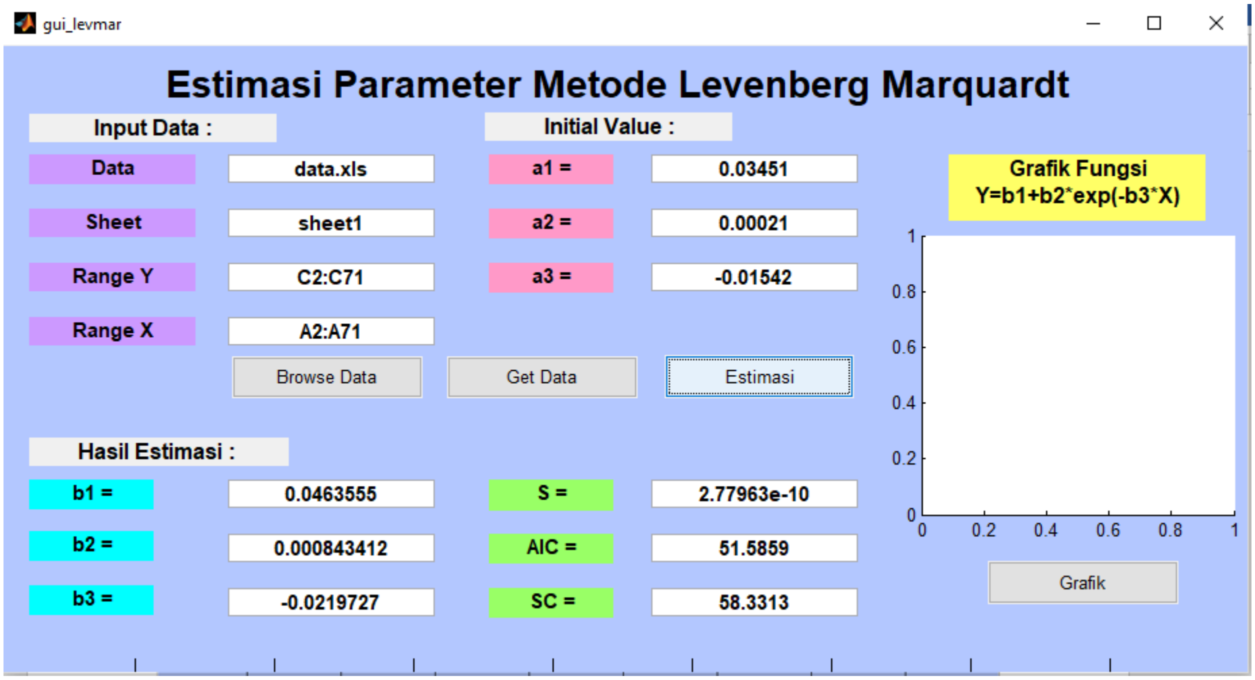


5. Klik tombol *Estimasi* dan tunggu hasil estimasi.

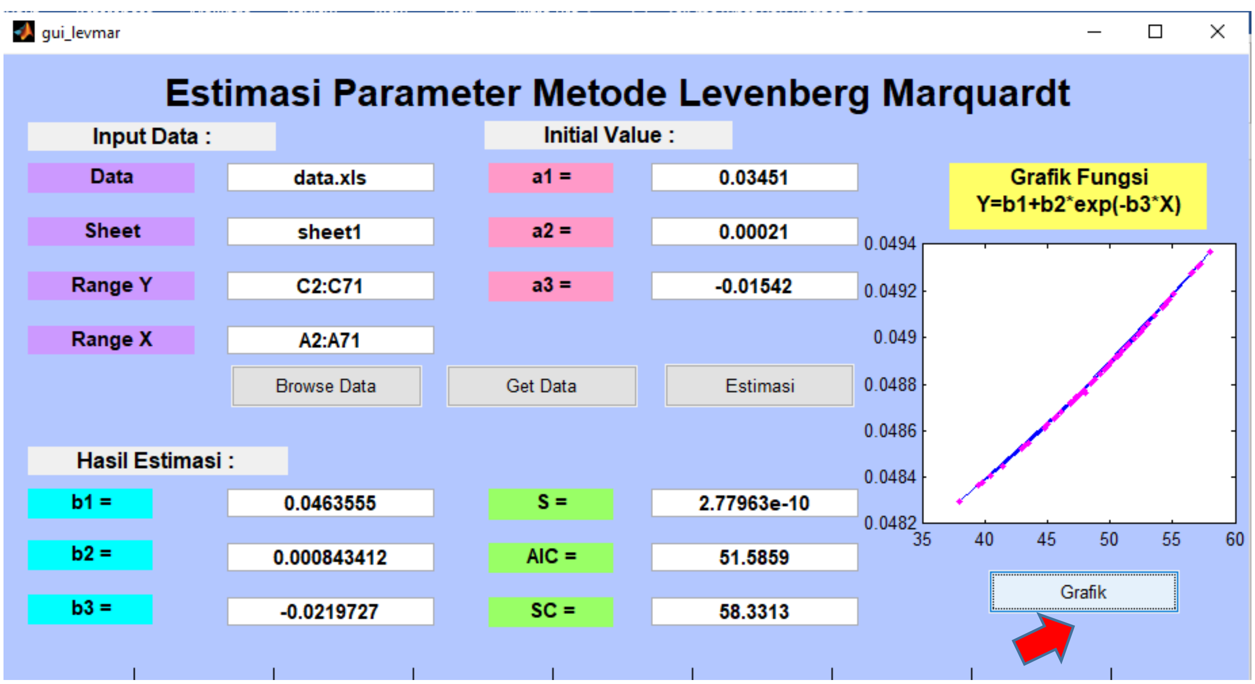




6. Hasil estimasi parameter model regresi non linear dengan algoritma Levenberg-Marquardt beserta nilai S, AIC dan SC ditampilkan pada kolom *blankspace* yang sudah disediakan seperti gambar berikut :



7. Klik tombol Grafik untuk menampilkan grafik model regresi non linear dengan nilai parameter yang telah diperoleh.



DAFTAR PUSTAKA

[1] Yusnandar, 2004. Aplikasi Analisis Regresi Non Linear Model Kuadratik Terhadap Produksi Susu Kambing Peranakan Etawah (PE) Selama 90 Hari Pertama Laktasi. Informatika Pertanian, 13, 736-743

[2] Griva, I., Nash, S.G. and Sofer, A., 2009, *Linear and Nonlinear Optimization*, Second Edition, The Society for Industrial and Applied Mathematics, Philadelphia.

[3] Yu, H. dan Wilamowski, B.M., 2011, *Advanced Learning Algorithms of Neural Networks*, Disertasi, Auburn University, USA.

[4] Marquardt, D., 1963, *An Algorithm for Least Squares Estimation of Nonlinear Parameter*. Journal of the Society for Industrial and Applied Mathematics, 2, 11, 431-441.

[5] Gavin, H.P., 2019, *The Levenberg-Marquardt Algorithm for Nonlinear Least Squares Curve-Fitting Problems*, Duke University.

PAPER NAME

HKI GUI Sistem Regresi Non Linear_Atika .pdf

WORD COUNT

3800 Words

CHARACTER COUNT

22538 Characters

PAGE COUNT

20 Pages

FILE SIZE

413.5KB

SUBMISSION DATE

Aug 4, 2022 10:47 AM GMT+7

REPORT DATE

Aug 4, 2022 10:48 AM GMT+7

18% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 18% Internet database
- Crossref database
- 1% Submitted Works database
- 1% Publications database
- Crossref Posted Content database

Excluded from Similarity Report

- Manually excluded text blocks

Aplikasi GUI Matlab untuk Menaksir Koefisien Parameter Model Regresi Non Linier Menggunakan Algoritma Levenberg-Marquardt

REKAYASA DATA

PENGUSUL

Atika Ratna Dewi, S.Si, M.Sc 0615128703

INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2020

1. Pendahuluan

Regresi non linier merupakan suatu analisis regresi dimana data penelitian digambarkan oleh suatu fungsi yang merupakan kombinasi non linier dari parameter-parameter atau dapat diartikan sebagai hubungan antara variabel independen (X) terhadap variabel dependen (Y) [1]. Penaksiran parameter pada regresi non linier dapat dilakukan dengan beberapa macam cara salah satunya menggunkan algoritma *Levenberg-Marquardt.*

Algoritma *Levenberg-Marquardt* dikembangkan pertama kali pada tahun 1963 untuk menyelesaikan masalah *nonlinear least square* [4]. Algoritma *Levenberg-Marquardt* merupakan gabungan antara algoritma *Gauss-Newton* dan algoritma *Steepest Descent*. Berdasarkan penelitian terdahulu menunjukkan bahwa algoritma Levenberg-Marquardt adalah algoritma yang cepat dan memiliki konvergensi yang stabil [3]. Metode *Levenberg-Marquardt* menggunakan metode algoritma seperti algoritma *Gauss-Newton* yaitu menggunakan *first order condition* (FOC) dari *sum of least square error* [4]. Perbedaannya adalah ada penambahan perkalian skalar dan matriks identitas μI_K pada algoritma *Levenberg-Marquardt*. Selain itu, penentuan panjang langkah atau *step length* (t_n) dalam algoritma *Levenberg-Marquardt* dapat bervariasi.Riset lanjutan tentang estimasi parameter juga telah banyak diteliti, salah satunya penelitian [5].

Penaksiran parameter β yang dilakukan secara manual akan cukup rumit dan memakan waktu yang lama. Untuk memudahkan penaksiran parameter β pada algoritma *Levenberg-Marquardt* dikembangkan aplikasi berbasis Matlab. Matlab merupakan salah satu software matematika yang dapat digunakan menyelesaikan berbagai persoalan matematika. Pengembangan aplikasi GUI (*Graphical User Interface*) pada Matlab hadir sebagai alat yang dapat memudahkan dan mempercepat pengguna matlab dalam menyelesaikan masalah matematika. Oleh karena itu akan dikembangkan aplikasi GUI Matlab guna menaksir parameter pada model regresi non linier menggunakan algoritma *Levenberg Marquardt*.

2. Deskripsi

2.1. Sistem Estimasi Parameter Model Regresi Non Linear menggunakan Algoritma Levenberg-Marquardt

Bentuk umum dari model regresi non linier adalah

$y = f(X, \beta) + e,$ (1)

dengan fungsi non linier dalam parameter β dan $e \sim N(0, \sigma^2 I_T)$. Ada 2 cara untuk menaksir β pada model regresi non linier yaitu dengan metode *nonlinear least square* dan *maximum likelihood*. Kedua metode tersebut menghasilkan penaksiran β yaitu:

$\hat{\beta} = f(X, \beta) + e.$ (2)

Penaksiran β dengan metode *nonlinear least square* bertujuan untuk mendapatkan nilai β yang meminimumkan residual *sum of squares* $S(\beta)$ [2].

$$\min_{\beta} S(\beta) = e'e, \tag{3}$$

$$= \left(y - f(x, \beta)\right)' \left(y - f(x, \beta)\right). \tag{4}$$

Syarat perlu untuk minimisasi adalah

$$\frac{\partial S}{\partial \beta} = -2\,[Z(\beta)]'\big(y - f(X,\beta)\big), \qquad (5)$$

$$[Z(\beta)]'(y - f(X,\beta)) = 0. \qquad (6)$$

Fungsi $f(X,\beta)$ dalam persamaan (6) adalah fungsi non linier sehingga penaksiran nilai β memerlukan proses iterasi yang memberikan global minimum. Secara umum, iterasi untuk mendapatkan taksiran β dengan *nonlinear least square* adalah

$$\beta^{(n+1)} = \beta^{n} + t_n P_n \gamma_n. \qquad (7)$$

Jenis iterasi yang dapat digunakan untuk mendapatkan taksiran β dengan *nonlinear least square* antara lain:

1. Algoritma *Gauss-Newton*
2. Algoritma *Steepest Descent*
3. Algoritma *Levenberg-Marquardt*

Aproksimasi $y = f(X,\beta)$ di sekitar *initial value* $\beta^{(1)}$ dilakukan dengan deret *Taylor* orde 1, yaitu

$$f(X,\beta) = f\big(X,\beta^{(1)}\big) + \frac{\partial f(X,\beta)}{\partial \beta'}\Big|_{\beta^{(1)}}\big(\beta - \beta^{(1)}\big), \qquad (3)$$

Misalkan $\frac{\partial f(X,\beta)}{\partial \beta'}\Big|_{\beta^{(1)}} = Z\big(\beta^{(1)}\big)$, maka:

$$y = f\big(X,\beta^{(1)}\big) + Z\big(\beta^{(1)}\big)\beta - Z\big(\beta^{(1)}\big)\beta^{(1)} + e, \qquad (4)$$

atau

$$y - f\big(X,\beta^{(1)}\big) + Z\big(\beta^{(1)}\big)\beta^{(1)} = Z\big(\beta^{(1)}\big)\beta + e,$$

$$\bar{y}\big(\beta^{(1)}\big) = Z\big(\beta^{(1)}\big)\beta + e. \qquad (5)$$

Nilai parameter β dapat ditaksir dengan menggunakan metode *least square*, diperoleh

$$\beta^{(2)} = \Big(Z\big(\beta^{(1)}\big)'Z\big(\beta^{(1)}\big)\Big)^{-1} Z\big(\beta^{(1)}\big)'\bar{y}\big(\beta^{(1)}\big) \qquad (6)$$

Algoritma secara umum yang diperoleh adalah

$$\beta^{(n+1)} = \beta^{(n)} + \Big(Z\big(\beta^{(n)}\big)'Z\big(\beta^{(n)}\big)\Big)^{-1} Z(\beta^{(n)})'\Big(y - f\big(X,\beta^{(n)}\big)\Big). \qquad (7)$$

Persamaan (7) disebut sebagai persamaan algoritma *Gauss-Newton*, sedangkan persamaan algoritma *Levenberg-Marquardt* adalah persamaan yang diperoleh dengan cara memodifikasi persamaan (7) menjadi :

$$\beta^{(n+1)} = \beta^{(n)} + \Big(Z\big(\beta^{(n)}\big)'Z\big(\beta^{(n)}\big) + \mu I\Big)^{-1} Z(\beta^{(n)})'\Big(y - f\big(X,\beta^{(n)}\big)\Big). \qquad (8)$$

Persamaan (8) disebut sebagai algoritma *Levenberg-Marquardt*, dimana μ adalah *damping parameter* yang nilainya tidak boleh negatif dan biasanya nilai μ merupakan faktor dari 10. Sedangkan I adalah matriks identitas. algoritma *Levenberg-Marquardt* akan berhenti pada saat nilai algoritma tersebut konvergen yaitu jika memenuhi :

$$\|\beta^{n+1} - \beta^{n}\| \le \varepsilon. \qquad (9)$$

Koding dari sistem yang dibangun disajikan pada uraian di bawah ini.

2.2 Source Code

a. Main Source Code

```
function varargout = gui_levmar(varargin)
```

```
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @gui_levmar_OpeningFcn, ...
                   'gui_OutputFcn',  @gui_levmar_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 gui_levmar is made visible.
function gui_levmar_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 gui_levmar (see VARARGIN)

% Choose default command line output for gui_levmar
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes gui_levmar wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = gui_levmar_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 data_Callback(hObject, eventdata, handles)
% hObject    handle to data (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 data as text
%        str2double(get(hObject,'String')) returns contents of data as a
double


% --- Executes during object creation, after setting all properties.
function data_CreateFcn(hObject, eventdata, handles)
% hObject    handle to data (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 sheet_Callback(hObject, eventdata, handles)
% hObject    handle to sheet (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 sheet as text
%        str2double(get(hObject,'String')) returns contents of sheet as a
double


% --- Executes during object creation, after setting all properties.
function sheet_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sheet (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 rangey_Callback(hObject, eventdata, handles)
% hObject    handle to rangey (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 rangey as text
%        str2double(get(hObject,'String')) returns contents of rangey as a
double
```

```
% --- Executes during object creation, after setting all properties.
function rangey_CreateFcn(hObject, eventdata, handles)
% hObject    handle to rangey (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 rangex_Callback(hObject, eventdata, handles)
% hObject    handle to rangex (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 rangex as text
%        str2double(get(hObject,'String')) returns contents of rangex as a
double


% --- Executes during object creation, after setting all properties.
function rangex_CreateFcn(hObject, eventdata, handles)
% hObject    handle to rangex (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 nilaiawal1_Callback(hObject, eventdata, handles)
% hObject    handle to nilaiawal1 (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 nilaiawal1 as text
%        str2double(get(hObject,'String')) returns contents of nilaiawal1 as
a double


% --- Executes during object creation, after setting all properties.
function nilaiawal1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nilaiawal1 (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 nilaiawal2_Callback(hObject, eventdata, handles)
% hObject    handle to nilaiawal2 (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 nilaiawal2 as text
%        str2double(get(hObject,'String')) returns contents of nilaiawal2 as
a double


% --- Executes during object creation, after setting all properties.
function nilaiawal2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nilaiawal2 (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 nilaiawal3_Callback(hObject, eventdata, handles)
% hObject    handle to nilaiawal3 (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 nilaiawal3 as text
%        str2double(get(hObject,'String')) returns contents of nilaiawal3 as
a double


% --- Executes during object creation, after setting all properties.
function nilaiawal3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to nilaiawal3 (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 hasil1_Callback(hObject, eventdata, handles)
% hObject    handle to hasil1 (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 hasil1 as text
%        str2double(get(hObject,'String')) returns contents of hasil1 as a
double


% --- Executes during object creation, after setting all properties.
function hasil1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hasil1 (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 hasil2_Callback(hObject, eventdata, handles)
% hObject    handle to hasil2 (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 hasil2 as text
%        str2double(get(hObject,'String')) returns contents of hasil2 as a
double


% --- Executes during object creation, after setting all properties.
function hasil2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hasil2 (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 hasil3_Callback(hObject, eventdata, handles)
% hObject    handle to hasil3 (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 hasil3 as text
%        str2double(get(hObject,'String')) returns contents of hasil3 as a
double


% --- Executes during object creation, after setting all properties.
function hasil3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hasil3 (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 sse_Callback(hObject, eventdata, handles)
% hObject    handle to sse (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 sse as text
%        str2double(get(hObject,'String')) returns contents of sse as a
double


% --- Executes during object creation, after setting all properties.
function sse_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sse (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 aic_Callback(hObject, eventdata, handles)
% hObject    handle to aic (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 aic as text
%        str2double(get(hObject,'String')) returns contents of aic as a
double


% --- Executes during object creation, after setting all properties.
function aic_CreateFcn(hObject, eventdata, handles)
% hObject    handle to aic (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 sc_Callback(hObject, eventdata, handles)
% hObject    handle to sc (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 sc as text
%        str2double(get(hObject,'String')) returns contents of sc as a double


% --- Executes during object creation, after setting all properties.
function sc_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sc (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 browsedata.
function browsedata_Callback(hObject, eventdata, handles)
% hObject    handle to browsedata (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
filename=get(handles.data,'string');
[data,pathname]=uigetfile({'*.xls','Microsoft Excel (.xls)';...
    '*.*','All Files (*.*)'},'Pick a file');
if data == 0
    set(handles.file_name,'string',filename);
else
    set(handles.data,'string',data);
    set(handles.sheet,'string','sheet1');
    set(handles.rangey,'string','C2:C71');
    set(handles.rangex,'string','A2:A71');
end


% --- Executes on button press in getdata.
function getdata_Callback(hObject, eventdata, handles)
% hObject    handle to getdata (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global y x
data=get(handles.data,'string');
sheet=get(handles.sheet,'string');
rangey=get(handles.rangey,'string');
rangex=get(handles.rangex,'string');
y=xlsread(data,sheet,rangey);
x=xlsread(data,sheet,rangex);

set(handles.nilaiawal1,'string',0.03451);
set(handles.nilaiawal2,'string',0.00021);
set(handles.nilaiawal3,'string',-0.01542);


% --- Executes on button press in estimasi.
function estimasi_Callback(hObject, eventdata, handles)
% hObject    handle to estimasi (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global y x
a1=str2double(get(handles.nilaiawal1,'string'));
a2=str2double(get(handles.nilaiawal2,'string'));
a3=str2double(get(handles.nilaiawal3,'string'));
tic;
b=[a1 a2 a3]';
%b=[0.0345 0.00021 -0.01542]'; %initial value data 1

% Data tahun ke 2
% y=gaji(:,4); %log gaji 2
% x=gaji(:,2); %usia 2
% tic;
% b=[0.0219 0.0259 -0.2253]'; %initial value data 2


% Levenberg-Marquardt Iterations
T=length(x);
rep = 70000 ;%the size of 'rep' depends on the convergence the initial value
k = length(b);
```

```
e = eye(k);
f = f2(b,x);
S = (y-f)'*(y-f);

j1 = 0;
j2 = 0;
tn = 50; % dapat melakukan perubahan tn
lamda = 0.1 ;% dapat melakukan perubahan nilai lamda

h=waitbar(0,'Waiting Hasil Estimasi...');
for i = 1:rep ;
      z = numgradf2(b,x);%Numerical gradient of f1
      zS = numgradS2(b,x,y) ;%Numerical gradient of S1
      step  = -0.5.*inv(z'*z + lamda*eye(k))*zS ; %  Marquardt-Levenberg
Iterations
      bnext = b + step;
      fnext = f2(bnext,x) ;
      Snext = (y-fnext)'*(y-fnext);


     while Snext < S && j1 <=100;
     step  = step*tn;  % Perubahan tn
     bnext = b+step;
     fnext = f2(bnext,x);
     Snext = (y-fnext)'*(y-fnext);
     j1    = j1+1;
     w1    = i;
     end;

   while Snext > S && j2 <=100;
      step  = step/tn;  % Perubahan tn
      bnext = b+step;
      fnext = f2(bnext,x);
      Snext = (y-fnext)'*(y-fnext);
      j2    = j2+1;
      w2    = i;
   end;

    if norm(bnext-b) <= 1e-9 && abs(S-Snext) <= 1e-9
       disp('Sudah konvergen. Dengan jumlah iterasinya adalah:') ;
       disp(i) ;
         break ;
    end ;

      if i == rep
         disp('Belum konvergen, iterasinya perlu ditambah lagi.') ;
         disp('Atau ubahlah initial values-nya') ;
         disp(' ') ;
      end ;

      b = bnext;
      f = f2(b,x) ;
      S = (y-f)'*(y-f);
waitbar((i/rep),h);
end ;
```

```
close(h)

p=length(y);
bK=bnext;
out=bK';
fin=f2(bK,x);
s2=(y-fin)'*(y-fin)/(p-k);
sse=S;

% Menentukan AIC dan SC
% Menggunakan file L2.m
LL    = L2(b,x,y);
AIC   = abs(-2*LL+2*k);
SC    = abs(-2*LL+log(T)*k);

set(handles.hasil1,'String',bnext(1));
set(handles.hasil2,'String',bnext(2));
set(handles.hasil3,'String',bnext(3));
set(handles.sse,'String',sse);
set(handles.aic,'String',AIC);
set(handles.sc,'String',SC);


% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes1


% --- Executes on button press in grafik.
function grafik_Callback(hObject, eventdata, handles)
% hObject    handle to grafik (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global x y
c1=str2double(get(handles.hasil1,'string'));
c2=str2double(get(handles.hasil2,'string'));
c3=str2double(get(handles.hasil3,'string'));
D=c1+(c2.*exp(-c3.*x));
plot(x,D);
hold on
plot (x,y,'m.');
axes(handles.axes2);
```

b. Source Code f2

```
%File f2.m
function f = f2(b,x)
t=x(:,1);
b1=b(1,:);
b2=b(2,:);
b3=b(3,:);
f=b1+(b2*exp(-b3*t));
end
```

c. Source Code Numgradf2

```
%File Numgradf2.m
function z = numgradf2(b,x)
% Numerical z (numerical gradient of
k = length(b);
d = 1e-7;
e = eye(k);

for j=1:k ;
    bplus = b + d*e(:,j);
    fplus = feval('f2',bplus,x) ;
    bmin  = b - d*e(:,j) ;
    fmin  = feval('f2',bmin,x) ;
    z(:,j)= (fplus - fmin)/(2*d);
end;
```

d. Source Code L2

```
%File L2.m
function LL = L2(b,x,y)
T  = length(x);
f  = f2(b,x);
s2 = ((y-f)'*(y-f))/T;
LL  = -0.5*(log (2*pi*s2) + (y-f)'*(y-f)/s2);
end
```

e. Source Code NumgradS2

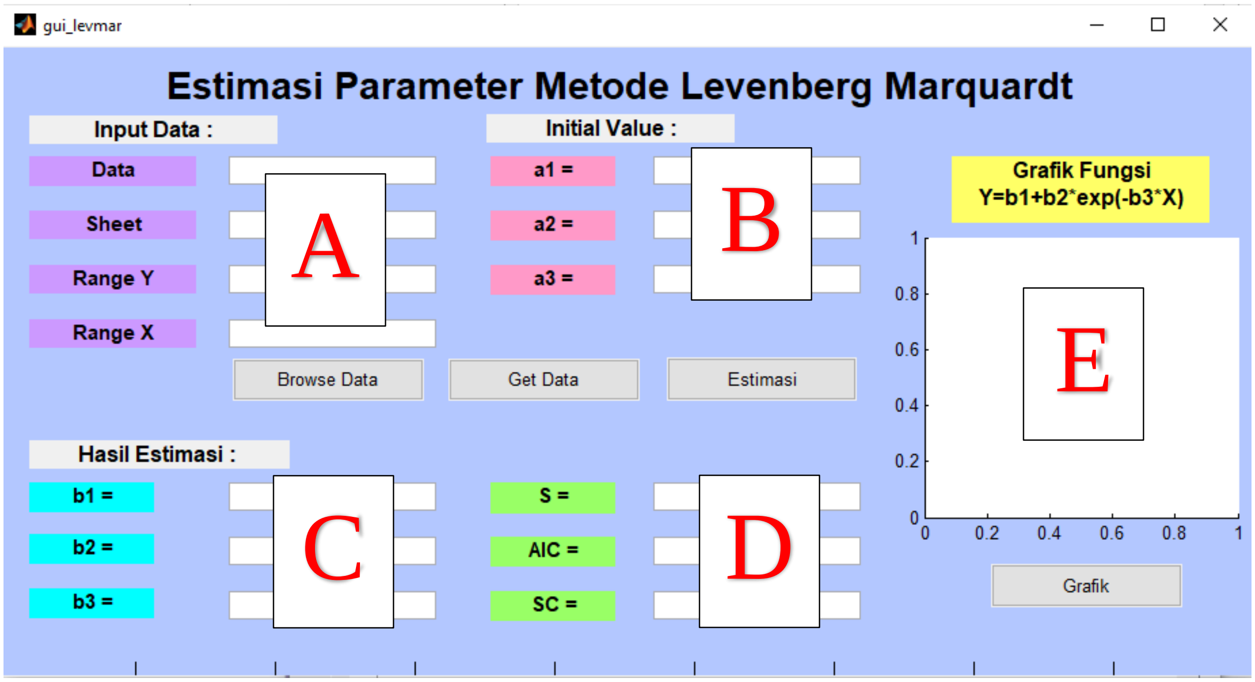
```
%File Numgrads2.m
function z = numgradS2(b,x,y)
% Numerical z (numerical gradient of L)
% Output berupa vector dengan dimensi Kx1

k = length(b);
d = 1e-6;
e = eye(k);

for j=1:k;% Numerical gradients
    bplus = b + d*e(:,j) ;
    fplus = feval('f2',bplus,x) ;
    Splus = (y-fplus)'*(y-fplus);
    bmin  = b - d*e(:,j) ;
    fmin  = feval('f2',bmin,x) ;
    Smin  = (y-fmin)'*(y-fmin);
    z(j,:) = (Splus - Smin)/(2*d);
end;
```

2.2. Hasil Visualisasi

Tampilan Estimasi Parameter Model Regresi Non Linier Metode Levenberg Marquardt ditunjukkan pada Gambar 1.



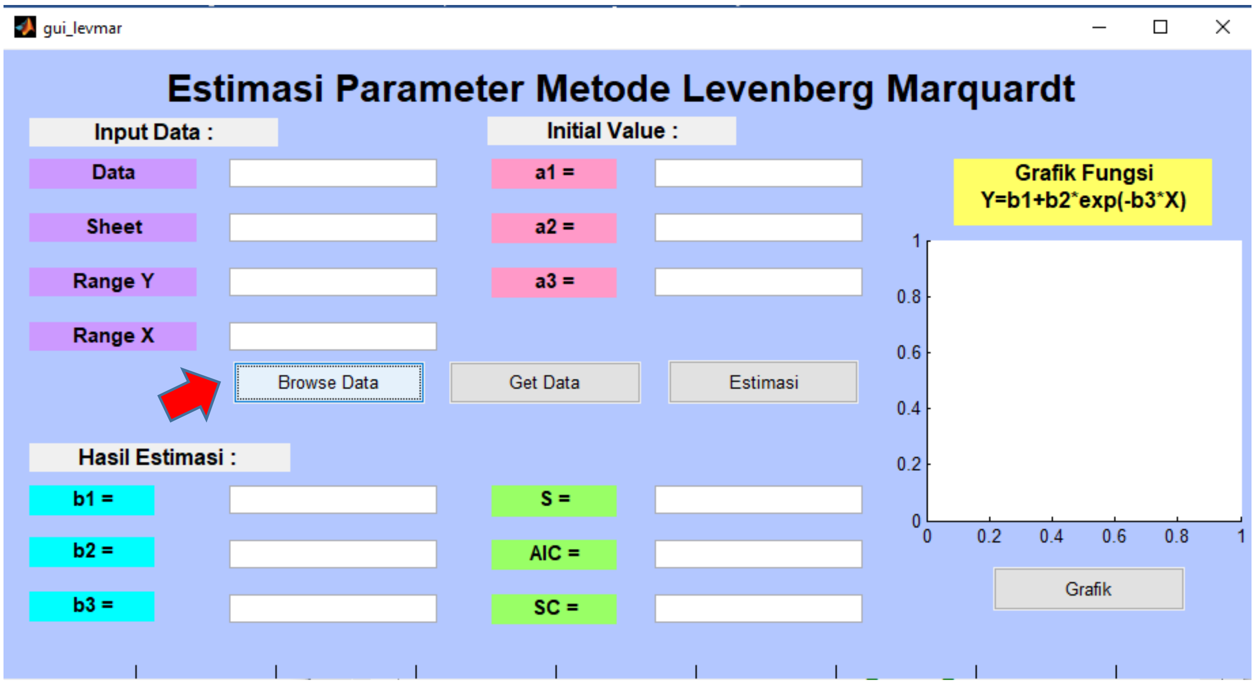
Deskripsi dari masing-masing komponen di dalam tampilan antara lain:

a. ***Browse Data*** merupakan *pushbutton* yang berfungsi untuk mengambil data yang pada folder tertentu untuk kemudian diproses dengan algoritma Levenberg-Marquardt.
b. ***Get Data*** merupakan *pushbutton* yang akan digunakan untuk mengambil nilai awal data (*initial value*).
c. ***Estimasi*** merupakan *pushbutton* untuk proses estimasi parameter model regresi non linear dengan algoritma Levenberg-Marquardt.
d. ***Grafik*** merupakan *pushbutton* untuk proses menampilkan grafik hasil estimasi parameter model regresi non linear dengan algoritma Levenberg-Marquardt.
e. **A** merupakan *blankspace* yang akan menampilkan data yang akan diestimasi.
f. **B** merupakan *blankspace* yang akan menampilkan nilai awal data (*initial value*).
g. **C** merupakan *blankspace* yang akan menampilkan hasil estimasi parameter model regresi non linear dengan algoritma Levenberg-Marquardt.
h. **D** merupakan *blankspace* yang akan menampilkan hasil nilai S, AIC dan SC.
i. **E** merupakan *blankspace* yang akan menampilkan hasil grafik hasil estimasi parameter model regresi non linear dengan algoritma Levenberg-Marquardt.

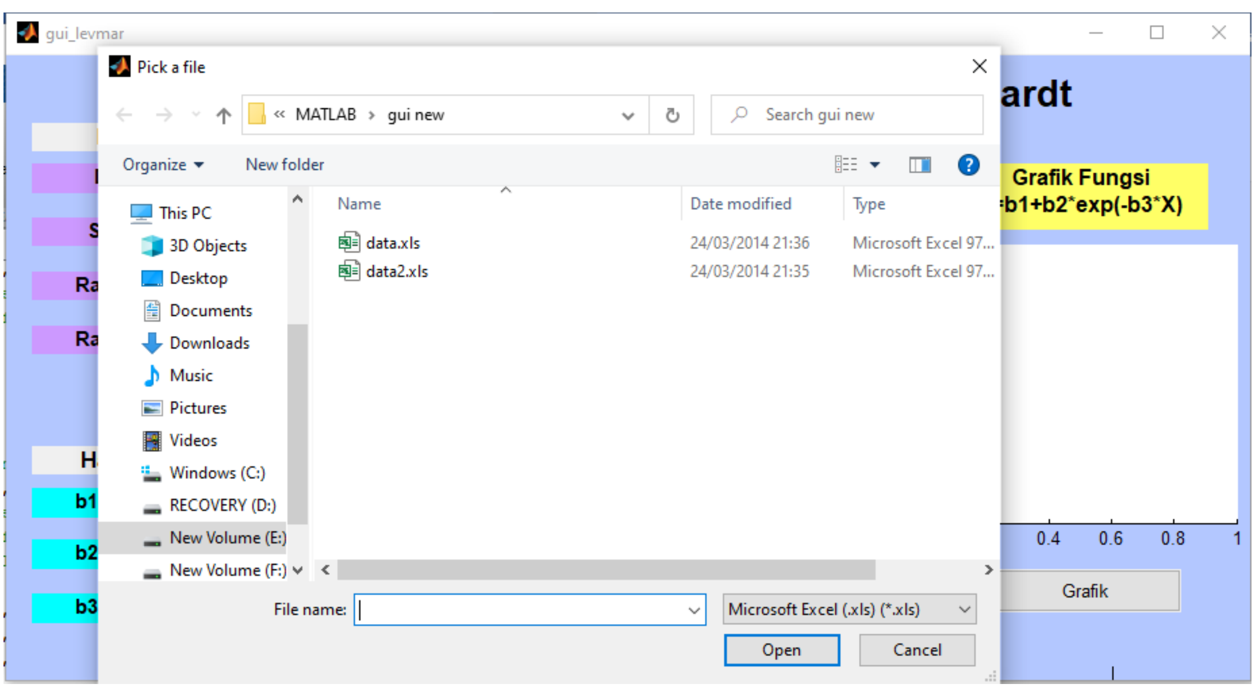
2.3. PROSES PENGGUNAAN GUI

Langkah-langkah dari penggunaan GUI sebagai berikut:

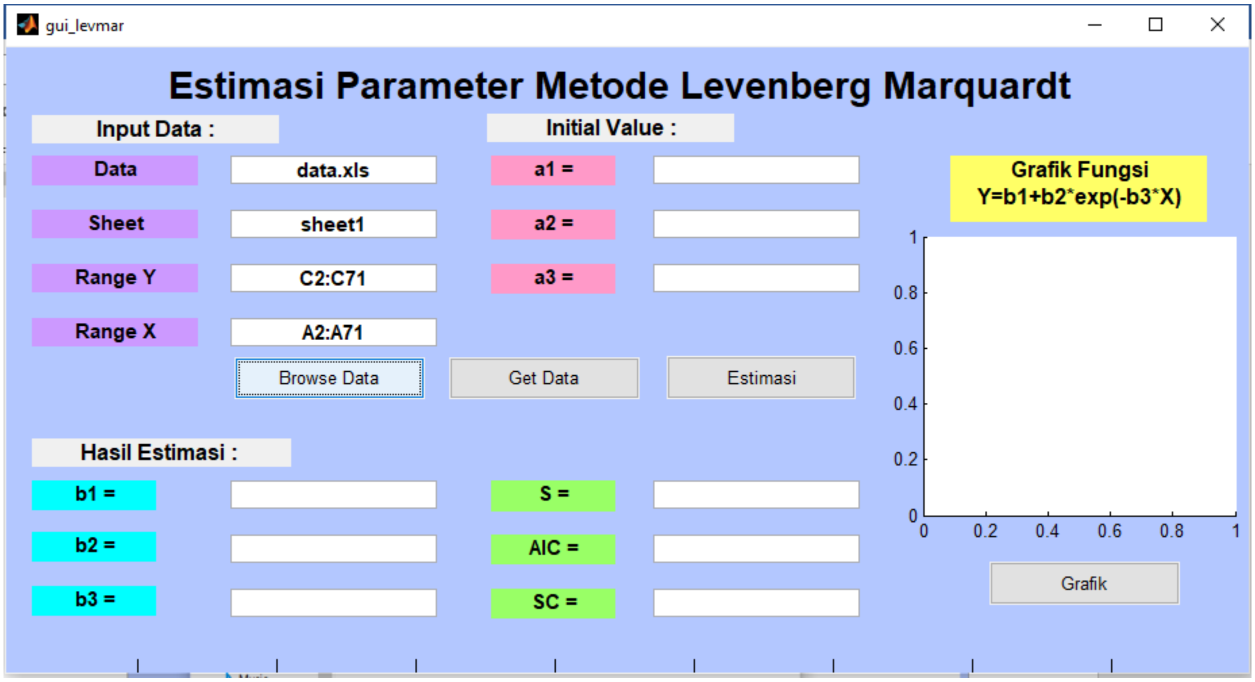
1. Mengambil data yang akan diestimasi dengan menekan tombol **BROWSE DATA**.



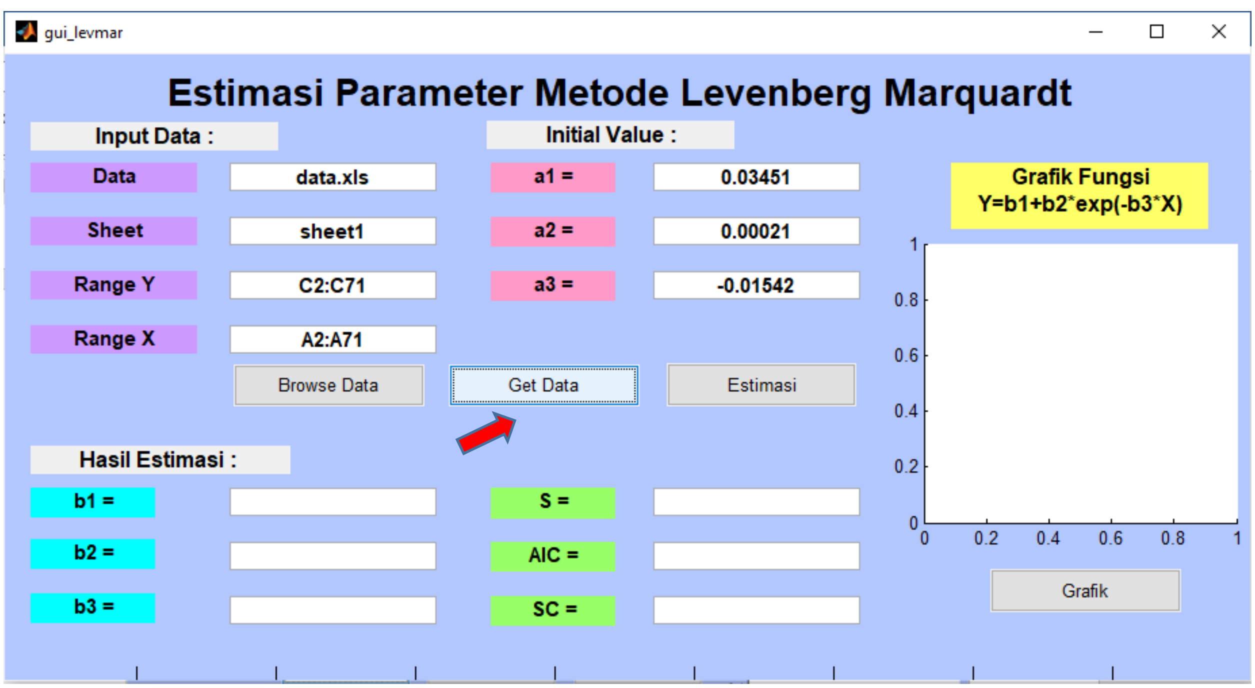
2. Setelah menekan tombol Browse Data maka akan muncul tampilan sebagai berikut dan kemudian pilih file yang akan digunakan pada proses estimasi :



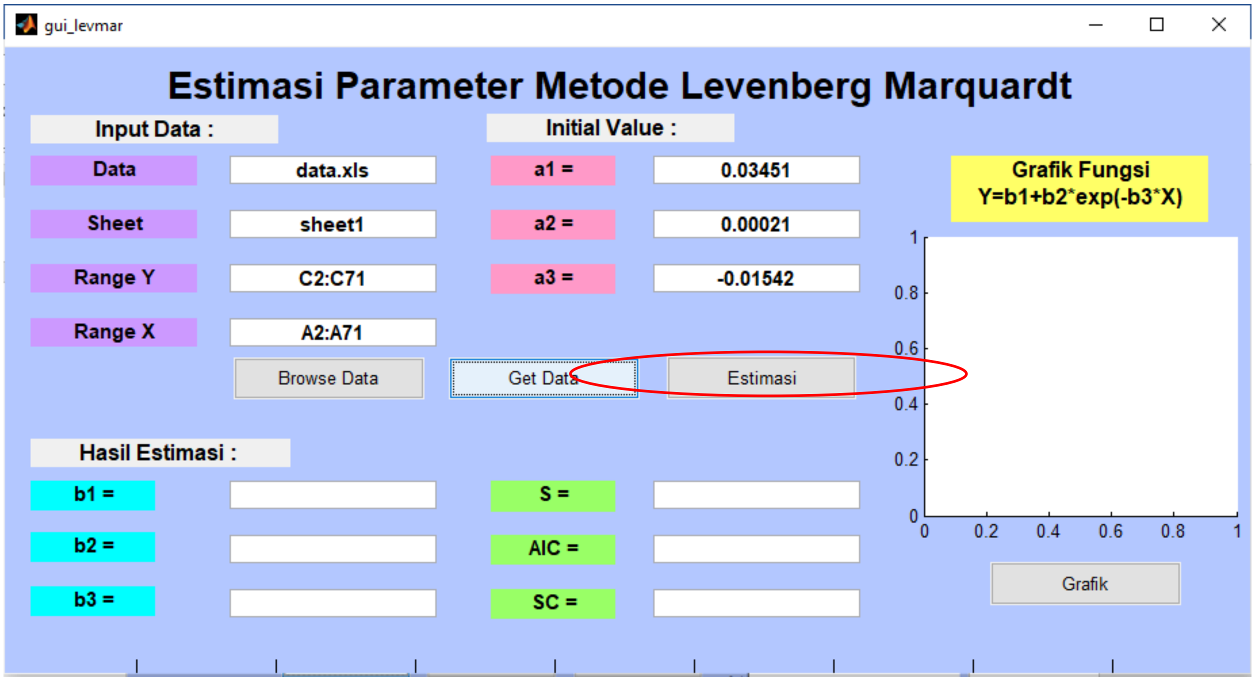
3. Data yang dipilih akan ditampilkan pada *blankspace* yang telah disediakan seperti gambar berikut :

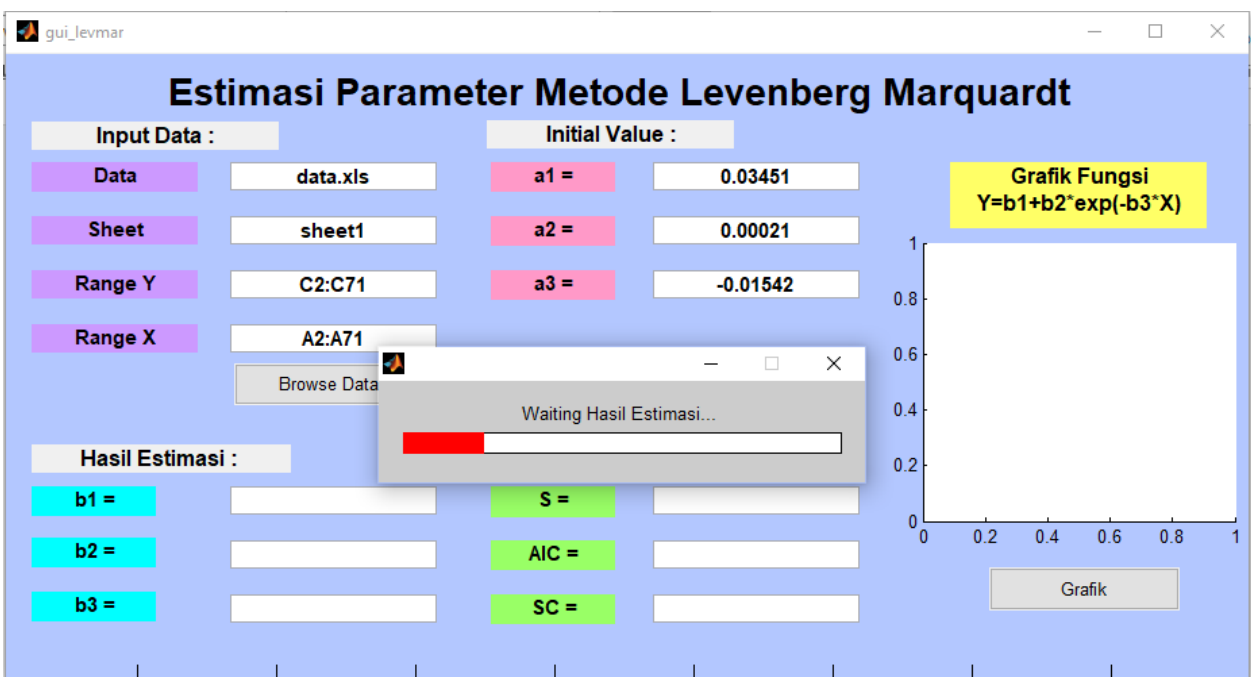


4. Menampilkan nilai awal data (*initial value*) yang akan digunakan untuk estimasi parameter dengan menekan tombol **GET DATA**.

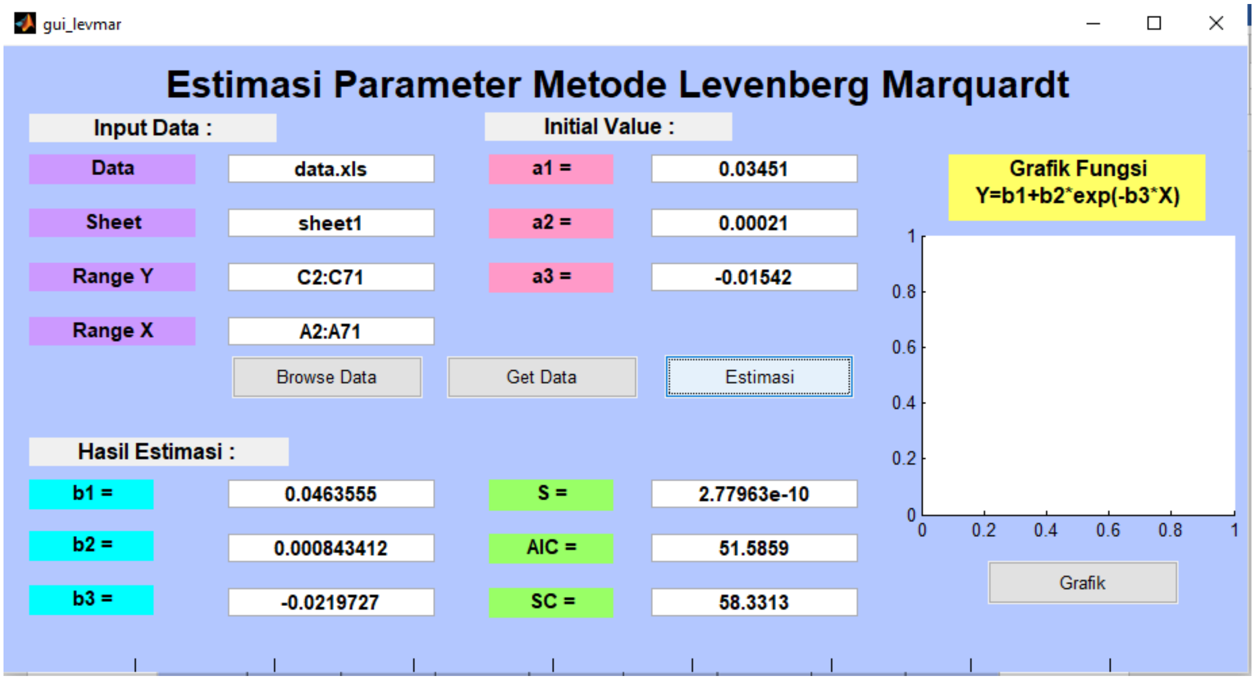


5. Klik tombol *Estimasi* dan tunggu hasil estimasi.

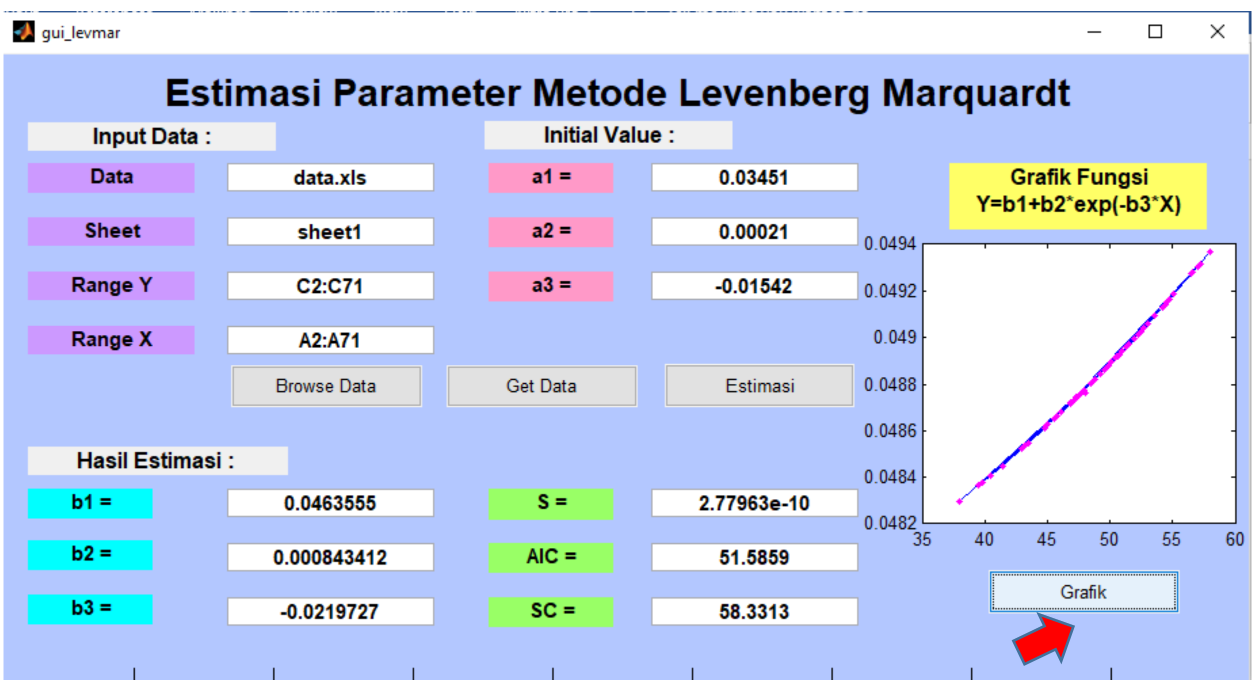




6. Hasil estimasi parameter model regresi non linear dengan algoritma Levenberg-Marquardt beserta nilai S, AIC dan SC ditampilkan pada kolom *blankspace* yang sudah disediakan seperti gambar berikut :



7. Klik tombol Grafik untuk menampilkan grafik model regresi non linear dengan nilai parameter yang telah diperoleh.



DAFTAR PUSTAKA

[1] Yusnandar, 2004. Aplikasi Analisis Regresi Non Linear Model Kuadratik Terhadap Produksi Susu Kambing Peranakan Etawah (PE) Selama 90 Hari Pertama Laktasi. Informatika Pertanian, 13, 736-743

[2] Griva, I., Nash, S.G. and Sofer, A., 2009, *Linear and Nonlinear Optimization*, Second Edition, The Society for Industrial and Applied Mathematics, Philadelphia.

[3] Yu, H. dan Wilamowski, B.M., 2011, *Advanced Learning Algorithms of Neural Networks*, Disertasi, Auburn University, USA.

[4] Marquardt, D., 1963, *An Algorithm for Least Squares Estimation of Nonlinear Parameter*. Journal of the Society for Industrial and Applied Mathematics, 2, 11, 431-441.

[5] Gavin, H.P., 2019, *The Levenberg-Marquardt Algorithm for Nonlinear Least Squares Curve-Fitting Problems*, Duke University.

18% Overall Similarity

Top sources found in the following databases:

- 18% Internet database
- Crossref database
- 1% Submitted Works database
- 1% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	repository.ittelkom-pwt.ac.id	Internet	14%
2	repository.uin-malang.ac.id	Internet	1%
3	docplayer.info	Internet	1%
4	id.123dok.com	Internet	<1%
5	elinyulianti97.wordpress.com	Internet	<1%
6	SDM Universitas Gadjah Mada on 2021-10-08	Submitted works	<1%
7	Wright State University on 2016-04-07	Submitted works	<1%
8	blog.chinaunix.net	Internet	<1%

9 gitlab.gwdg.de
Internet
<1%

10 etheses.uin-malang.ac.id
Internet
<1%

11 University of the West Indies on 2014-04-03
Submitted works
<1%

Excluded from Similarity Report

- Manually excluded text blocks

EXCLUDED TEXT BLOCKS

gui_Singleton = 1;gui_State = struct('gui_Name',mfilename, ...'gui_Singleton', gui_Si...

gitlab.gwdg.de

OutputFcn(hObject, eventdata, handles)% varargoutcell array for returning output a...

ir.swu.ac.th

end% --- Executes on button press in

se.mathworks.com

uigetfile({'*.xls

psasir.upm.edu.my

set(handles

University of Liverpool on 2021-08-16

end% --- Executes on button press in

id.123dok.com

set(handles

www.mathworks.com

Executes on button press in

id.123dok.com

e = eye(k);f = f2(b,x);S = (y-f)'*(y-f

etheses.uin-malang.ac.id

for i = 1

repository.uin-malang.ac.id

f = f2(b,x) ;S = (y-f)'*(y-f

etheses.uin-malang.ac.id

if norm(bnext-b) <= 1e-9

etheses.uin-malang.ac.id

bnext

id.scribd.com

bnext = b+step;fnext

id.scribd.com

bnext = b+step;fnext

id.scribd.com

Executes during object creation, after setting all properties.function axes1_Create...

rismonhasiholansianipar.blogspot.com

set(handles

pages.cs.wisc.edu

b,x,y);AIC = abs(-2*LL+2*k);SC = abs(-2*LL+log(T)*k

etheses.uin-malang.ac.id

c1=str2double(get(handles

University of Greenwich on 2020-04-06

plot(x

Auburn University - Engineering on 2006-03-17

function f = f2(b,x

etheses.uin-malang.ac.id

k = length(b

etheses.uin-malang.ac.id

fmin = feval('f2',bmin,x) ;z(:,j)= (fplus - fmin)/(2

etheses.uin-malang.ac.id

fplus = feval('f2',bplus,x) ;bmin

etheses.uin-malang.ac.id

T= length(x);f

repository.uin-malang.ac.id

function z = numgradS2(b,x,y)% Numerical

repository.uin-malang.ac.id

Kx1k = length(b

adoc.pub

feval('f2

etheses.uin-malang.ac.id

fmin= feval('f2',bmin,x

etheses.uin-malang.ac.id

z = numgradf2(b,x)% Numerical

repository.uin-malang.ac.id

Source Code

University of Newcastle on 2006-09-22

LEMBAR

HASIL PENILAIAN SEJAWAT SEBIDANG ATAU *PEER REVIEW*

KARYA ILMIAH : RANCANGAN DAN KARYA TEKNOLOGI YANG DIPATENKAN ATAU SENI YANG TERDAFTAR DI HKI

Judul Paten/HKI : Sistem Estimasi Parameter Model Regresi Non Linier
Jumlah Pencipta/Inventor : 1
Nama Pencipta/Inventor : Atika Ratna Dewi
Identitas Karya Paten :
a. Jenis Ciptaan : Program Komputer
b. Nomor Permohonan : EC00202054317
c. Tanggal Permohonan : 30 November 2020
d. Nomor Pencatatan/Paten : 000222858
e. Tahun : 2020

Kategori Karya Teknologi Yang Dipatenkan (beri √pada kategori yang tepat)

☐ Nasional (yang sudah diimplementasikan di Industri)
☐ Nasional
☐ Nasional dalam bentuk paten sederhana yang telah memiliki sertifikat dari Direktorat Jenderal Kekayaan Intelektual, Kemenkumham
√ Karya ciptaan, design industri,indikasi geografis yang telah memiliki sertifikat dari DireJen KI, Kemenkumham

Hasil Penilaian *Peer Review* :

Komponen yang dinilai	Nilai Maksimal				Nilai Akhir yang Diperoleh
	Nasional sudah diimplementasikan di industri	Nasional	Nasional dalam bentuk paten sederhana yang telah memiliki sertifikat dari DireJen KI, Kemenkumham	Karya ciptaan, design industri,indikasi geografis yang telah memiliki sertifikat dari DireJen KI, Kemenkumham	
a. Kelengkapan unsur isi artikel (10%)				1,5	1,4
b. Ruang lingkup dan kedalaman pembahasan (30%)				4,5	4,3
c. Kecukupan dan kemutakhiran data /informasi dan metodologi (30%)				4,5	4,3
d. Kelengkapan unsur dan kualitas terbitan/jurnal (30%)				4,5	4,5
Total = 100%				**15**	**14,5**
Nilai Pengusul				**15**	**14,5**

Catatan penilaian artikel oleh Reviewer 1 :

1. **Kelengkapan dan kesesuaian unsur :** Karya ciptaan yang diajukan telah memenuhi kelengkapan dan sesuai dengan standar yang harus dipenuhi karya ciptaan (HKI non paten).

2. **Ruang lingkup dan kedalaman :** Karya ciptaan yang diajukan merupakan aplikasi dari algoritma Levenberg-Marquardt untuk menaksir nilai koefisien parameter pada model regresi non linear. Ruang lingkup sudah cukup jelas dan khusus. Dilihat dari metode yang digunakan, dapat diambil kesimpulan bahwa karya ciptaan yang diajukan memiliki kedalaman materi yang memadai.

3. **Kecukupan dan kemutakhiran data serta metodologi :** Metode atau algoritma Levenberg-Marquardt merupakan metode yang sering digunakan. Begitu pula model regresi non linear. Namun

mengingat aplikasinya yang cukup banyak dan data yang digunakan sudah cukup, maka dapat dikatakan bahwa karya ciptaan yang diajukan sudah cukup mutakhir.

4. **Kelengkapan unsur kualitas penerbit :** Hak Kekayaan Intelektual non Paten yang diajukan diterbitkan oleh Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia, sehingga kualitas penerbit sudah sangat baik.

5. **Indikasi Plagiasi :** Dari hasil cek plagiarisme tidak ditemukan adanya indikasi plagiasi dari karya ciptaan yang diajukan.

6. **Kesesuaian Bidang Ilmu:** Bidang riset pengusul adalah data sains dan karya ciptaan yang diajukan adalah GUI Matlab dari model regresi non linear. Dapat disimpulkan bahwa bidang ilmu yang diajukan sudah sesuai dengan bidang pengusul.

Purwokerto, 18 Juli 2022
Reviewer 1,

Nama	: Dr. Ridwan Pandiya, M.Sc.
NIP/NIDN	: 0625088202
Unit Kerja	: Institut Teknologi Telkom Purwokerto
Jabatan Fungsional	: Lektor
Bidang Ilmu	: Matematika

Prosentase Angka Kredit Penulis untuk :

- **jurnal dan prosiding :**
  1. Penulis Pertama sekaligus korespondensi = 60%
  2. Terdiri dari : Penulis pertama; Korespondensi; Pendamping = : 40% ; 40%; 20%
  3. Terdiri dari : Penulis pertama; korespondensi = 50% ; 50%
- Karya ilmiah lain : Penulis pertama; Pendamping= 60%;40%

LEMBAR

HASIL PENILAIAN SEJAWAT SEBIDANG ATAU *PEER REVIEW*

KARYA ILMIAH : RANCANGAN DAN KARYA TEKNOLOGI YANG DIPATENKAN ATAU SENI YANG TERDAFTAR DI HKI

Judul Paten/HKI : Sistem Estimasi Parameter Model Regresi Non Linier
Jumlah Pencipta/Inventor : 1
Nama Pencipta/Inventor : Atika Ratna Dewi

Identitas Karya Paten :
a. Jenis Ciptaan : Program Komputer
b. Nomor Permohonan : EC00202054317
c. Tanggal Permohonan : 30 November 2020
d. Nomor Pencatatan/Paten : 000222858
e. Tahun : 2020

Kategori Karya Teknologi Yang Dipatenkan (beri √pada kategori yang tepat)

☐ Nasional (yang sudah diimplementasikan di Industri)
☐ Nasional
☐ Nasional dalam bentuk paten sederhana yang telah memiliki sertifikat dari Direktorat Jenderal Kekayaan Intelektual, Kemenkumham
√ Karya ciptaan, design industri,indikasi geografis yang telah memiliki sertifikat dari DireJen KI, Kemenkumham

Hasil Penilaian *Peer Review* :

Komponen yang dinilai	Nilai Maksimal				Nilai Akhir yang Diperoleh
	Nasional sudah diimplementasikan di industri	Nasional	Nasional dalam bentuk paten sederhana yang telah memiliki sertifikat dari DireJen KI, Kemenkumham	Karya ciptaan, design industri,indikasi geografis yang telah memiliki sertifikat dari DireJen KI, Kemenkumham	
a. Kelengkapan unsur isi artikel (10%)				1,5	1,4
b. Ruang lingkup dan kedalaman pembahasan (30%)				4,5	4,4
c. Kecukupan dan kemutakhiran data /informasi dan metodologi (30%)				4,5	4,4
d. Kelengkapan unsur dan kualitas terbitan/jurnal (30%)				4,5	4,4
Total = 100%				**15**	**14,6**
Nilai Pengusul				**15**	**14,6**
Nilai rata rata Reviewer 1 dan 2					**14,55**

Catatan penilaian artikel oleh Reviewer 2 :

1. **Kelengkapan dan kesesuaian unsur :**
   HKI Sistem Estimasi Parameter Model Regresi Non Linier ini dilihat dari aspek kelengkapan sudah memenuhi standar yang diharapkan. Selain itu, indikator-indikator kesesuaian unsur HKI telah dipenuhi.

2. **Ruang lingkup dan kedalaman :**
   Dalam HKI ini ruang lingkupnya sudah jelas. Kedalaman baik dari sisi teori maupun aplikasi sudah cukup baik.

3. **Kecukupan dan kemutakhiran data serta metodologi :**

HKI Sistem Estimasi Parameter Model Regresi Non Linier yang diajukan termasuk dalam kategori mutakhir. Metodologi yang digunakan sudah jelas dan sistematis.

4. **Kelengkapan unsur kualitas penerbit :**
HKI diterbitkan oleh lembaga pemerintah (Kemenkumham), sehingga kualitas penerbit sudah baik.

5. **Indikasi Plagiasi :**
Mengacu pada hasil tes plagiasi, tidak ditemukan adanya indikasi plagiarisme.

6. **Kesesuaian Bidang Ilmu:**
Karya ciptaan yang diajukan adalah Sistem Estimasi Parameter Model Regresi Non Linier, bidang ilmu yang diajukan sudah sesuai dengan bidang ilmu pengusul.

Purwokerto, 18 Juli 2022
Reviewer 2,

Nama : Diandra Chika Fransisca, S.Si., M.Sc
NIP/NIDN : 0618109301
Unit Kerja : IT Telkom Purwokerto
Jabatan Fungsional : Lektor
Bidang Ilmu : Matematika

Prosentase Angka Kredit Penulis untuk :
- **jurnal dan prosiding :**
  1. Penulis Pertama sekaligus korespondensi = 60%
  2. Terdiri dari : Penulis pertama; Korespondensi; Pendamping = : 40% ; 40%; 20%
  3. Terdiri dari : Penulis pertama; korespondensi = 50% ; 50%
- Karya ilmiah lain : Penulis pertama; Pendamping= 60%;40%

Lembaga Penelitian dan Pengabdian Masyarakat

Institut Teknologi Telkom Purwokerto

Jl. D.I. Panjaitan No. 128 Purwokerto 53147 Telp. (0281) 641629

SURAT TUGAS

Nomor: IT Tel883/LPPM-000/Ka. LPPM/III/2021

Bersama ini Kepala Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) IT Telkom Purwokerto menugaskan kepada Dosen yang namanya tersebut di bawah ini:

No	NIDN	Nama	JFA
1	0615128703	Atika Ratna Dewi, S.Si., M.Sc	Asisten Ahli
2	0626049003	Ridho Ananda, S.Pd., M.Si	Asisten Ahli
3	0617029101	Utti Marina, S.Si., M.Sc.	Asisten Ahli

Untuk mendaftarkan karya cipta sebagai Hak Kekayaan Intelektual (HKI) di Kementrian Hukum dan HAM, dengan judul

"Sistem GUI Matlab Simulasi Model Penyebaran Covid-19 Dengan Isolasi"

Pencipta :

1. Atika Ratna Dewi, S.Si., M.Sc
2. Ridho Ananda, S.Pd., M.Si
3. Utti Marina, S.Si., M.Sc.

Pemegang Hak Cipta : Rektor Institut Teknologi Telkom Purwokerto

Demikian surat tugas ini diberikan untuk dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.

Purwokerto, 3 Maret 2021

Kepala Bagian LPPM,

LPPM
Lembaga Penelitian
dan Pengabdian Masyarakat
ITTelkom Purwokerto

(Danny Kurnianto, S.T., M.Eng.)
NIDN. 0619048201

Tembusan:

1. Yth Rektor IT Telkom Purwokerto
2. Arsip