Lampiran 1. ***Routine*** **Program Simulasi**

```
%========== Membuat jendela halaman utama ==========%

figure1 = figure('PaperSize',[1 1],'PaperUnits','normalized',...
                 'NumberTitle','off',...
                 'Name','Halaman Utama',...
                 'menubar','none',...
                 'Color',[0.8 0.8 0.8]);
panel_utama=uipanel('parent',figure1,'Title','Thesis','FontSize',12,...
                 'BackgroundColor',[0.8 0.8 0.8], ...
                 'FontWeight','bold','Fontangle', ...
                 'Italic','shadowcolor',[1 1 0],...
                 'ForegroundColor',[1 1 0],'Position',[.02 .02 0.96 0.96]);
axes1=axes('Position',[0.4326 0.2982 0.1738 0.1])
logo_TI=imread('logo TI.jpg');
image(logo_TI);
axis off;

teks1=annotation(figure1,'textbox',[0.4424 0.505 0.1553 0.09955],...
   'String',{'Oleh :','Triyono Budi Santoso','0806422813'},...
   'HorizontalAlignment','center',...
   'FontWeight','bold',...
   'FontSize',12,...
   'FitBoxtoText','on',...
   'FontName','Times New Roman',...
   'LineStyle','none');
teks2=annotation(figure1,'textbox',[0.3041 0.1651 0.4393 0.09007],...
   'String',{'DEPARTEMEN TEKNIK INDUSTRI','PROGRAM PASCA
   SARJANA','UNIVERSITAS INDONESIA','2010'},...
   'HorizontalAlignment','center',...
   'FontWeight','bold',...
   'FontSize',14,...
   'FontName','Times New Roman',...
   'FitBoxToText','on',...
   'LineStyle','none');
teks3=annotation(figure1,'textbox',[0.2827 0.80 0.4561 0.1085],...
   'String',{' ANALISIS KUALITAS REDAMAN SERAT OPTIK UNTUK',
             'MENINGKATKAN KINERJA SISTEM TELEKOMUNIKASI',
                       'DENGAN MENGGUNAKAN DOE'},...
   'HorizontalAlignment','center',...
   'FontWeight','bold',...
   'FontSize',14,...
   'FitBoxtoText','on',...
   'FontName','Times New Roman',...
   'LineStyle','none',...
   'EdgeColor',[0 0 1]);
```

```
align_teks=[teks1 teks2 teks3]
align(align_teks,'distribute','centre')
tombol_simulasi=uicontrol(panel_utama,'style','pushbutton','string','Simulasi','unit
s','normalized','callback','jendela_doe',…
'Position',[0.87 0.01 0.1 0.05])

close;
%jendela pertanyaan
fig_pertanyaan_jum_per_bending=figure('Units','normalized',...
        'Position',[0.3 0.5 0.4 0.1],...
        'Color',[1 1 0],...
        'HandleVisibility','callback',...
        'NumberTitle','off',...
        'Name','Masukkan Jumlah Pengukuran',...
        'Menubar','none');

 %Text pertanyaan
jum_per_splicing=annotation(fig_pertanyaan_jum_per_bending,'textbox',[0.275
0.6028 0.1852 0.06048],...
    'String',{'Jumlah Pengukuran =              kali'},...
    'LineStyle','none',...
    'FitBoxtotext','on');

%Edit jumlah percobaan
edit_jumlah_pertanyaan = uicontrol(fig_pertanyaan_jum_per_bending,'Style',
'edit','Unit','normalized','Position',[0.55 0.35 0.06 0.26048]);
%Tombol Lanjutkan
tombol_lanjutkan =
uicontrol(fig_pertanyaan_jum_per_bending,'style','pushbutton',...
'string','Lanjutkan','Unit','normalized','Position', [0.8 0.15 0.15 0.26048],
'Callback', 'data_percobaan_faktor_tunggal_bending');

%========Mengumpulkan data percobaan pengukuran
splicing==========%
%Membaca input jumlah percobaan
edit_jumlah_percobaan=get(edit_jumlah_pertanyaan,'string')
jumlah=str2num(edit_jumlah_percobaan)
close;
fig_percobaan = figure('PaperSize',[1 1],'PaperUnits','normalized',...
    'NumberTitle','off',...
    'Name','Data Percobaan Splicing','resize','off', ...
    'menubar','none',...
    'Color',[0.8 0.8 0.8]);
uipanel('parent',figure1,'Title','Kombinasi Faktor (F) & Level
(L)','FontSize',12,'BackgroundColor',[0.8 0.8 0.8], ...
'FontWeight','bold','Fontangle','Italic','shadowcolor',[1 1 0],...
'ForegroundColor',[1 1 0],'Position',[.02 .02 0.96 0.96]);
matrix_percobaan = zeros (jumlah,6); % 6 adalah 2 x 3 level
```

```
cnames = {'F1L1-F2L1','F1L1-F2L2','F1L1-F2L3','F1L2-F2L1','F1L2-
F2L2','F1L2-F2L3'};

%=========== Menganalisis data percobaan pengukuran bending
==========%
h = waitbar(0,'Sedang Memproses, Silakan Tunggu....');
for i=1:4000,
waitbar(i/4000)
end
close(h)
%data_retrieve=get(tabel_percobaan,'data')
data_retrieve=xlsread('data_bending_final.xlsx');
close
%Menguji normalitas
input('1. Analisis Normalitas')
baca_data_bending =data_retrieve;
figure
plot_bending1=subplot(3,1,1);
normplot(baca_data_bending(:,1));
set(get(plot_bending1,'Xlabel'),'string','Brand 1')
title('')
plot_bending2=subplot(3,1,2);
normplot(baca_data_bending(:,2));
set(get(plot_bending2,'Xlabel'),'string','Brand 2')
title('')
plot_bending3=subplot(3,1,3);
normplot(baca_data_bending(:,3));
set(get(plot_bending3,'Xlabel'),'string','Brand 3')
title('')
set(gcf,'numbertitle','on','name','Plot normalitas data redaman bending untuk tiap
brand FO')
pause
close all;
%Menguji homogenitas varians
input('2. Analisis Homogenitas Varians')
data_retrieve_homogen = data_retrieve (:);

uji_homogen_bartlett=uji_varian (data_retrieve_homogen,group_respon);% uji
Bartlett
close;
set(gcf,'numbertitle','on','name','Uji Varians metode Bartlett');
uji_homogen_levene=uji_varian
(data_retrieve_homogen,group_respon,'on','robust'); % uji Levene
close;
set(gcf,'numbertitle','on','name','Uji Varians metode Levene');
pause
close all;
```

```
%Menganalisis data pengukuran dengan menggunakan ANOVA 1 Arah
input('3. Analisis Varians Satu Arah')
data_matrix=data_retrieve;

[p,tbl,stats] = anova_satu_arah_bending(data_matrix);
set(get(gca,'XLabel'),'String','Brand FO')
set(get(gca,'YLabel'),'String','Nilai Rata-rata')
figure
[c,m] = multcompare(stats); %untuk menguji perbedaan mean
set(gcf,'numbertitle','on','name','Perbandingan rata-rata redaman bending untuk
tiap brand FO')
set(get(gca,'XLabel'),'String','Redaman')
set(get(gca,'YLabel'),'String','Brand FO')
set(get(gca,'Title'),'String','')

%=========== Menganalisis data percobaan pengukuran patching
==========%
h = waitbar(0,'Sedang Memproses, Silakan Tunggu....');
for i=1:4000,
waitbar(i/4000)
end
close(h)
%data_retrieve=get(tabel_percobaan,'data')
data_retrieve=xlsread('data_patching_final.xlsx');
close;
%Menguji normalitas
input('1. Analisis Normalitas')
baca_data_patching =data_retrieve;
figure
plot_patching1=subplot(2,2,1);
normplot(baca_data_patching(:,1));
set(get(plot_patching1,'Xlabel'),'string','Outdoor')
title('ODF Tipe 1','Color','blue','FontWeight','bold')
plot_patching2=subplot(2,2,2);
normplot(baca_data_patching(:,3));
set(get(plot_patching2,'Xlabel'),'string','Outdoor')
title('ODF Tipe 2','Color','blue','FontWeight','bold')
plot_patching3=subplot(2,2,3);
normplot(baca_data_patching(:,2));
set(get(plot_patching3,'Xlabel'),'string','Indoor')
title('')
plot_patching4=subplot(2,2,4);
normplot(baca_data_patching(:,4));
set(get(plot_patching4,'Xlabel'),'string','Indoor')
title('')
set(gcf,'numbertitle','on','name','Plot normalitas data redaman patching untuk tiap
jenis ODF dan lokasi penempatan')
pause
```

```
close all;

data_retrieve1=data_retrieve(:,1:2);
data_matrix1=data_retrieve1(:);
data_retrieve2=data_retrieve(:,3:4);
data_matrix2=data_retrieve2(:);
data_matrix=[data_matrix1 data_matrix2];

%Menganalisis data pengukuran dengan menggunakan ANOVA 2 Arah
input('3. Analisis Varians Dua Arah')
reps=jumlah;
anova_dua_arah_patching(data_matrix,jumlah);
set(gcf,'numbertitle','on')

%Plotting Interaksi
n=jumlah;
a=2;
b=2;
t=a*b*n;
respon1=[data_retrieve(:,1); data_retrieve(:,3)];
respon2=[data_retrieve(:,2); data_retrieve(:,4)];
respon =[respon1;respon2];
group1=zeros(t,1);
m=0;
for i=1:b
   u=n*a*m+1;
   o=n*a*i;
   group1(u:o,1)=m+1;
   m=i;
end
group2=zeros(t,1);
m=0;
for i=1:b
   k=n*m+1;
   l=n*i;
   group2(k:l,1)=m+1;
   m=i;
end
h=0;
for i=1:b
   group2(n*a*h+1:n*a*i,1)=group2(1:n*a,1);
   h=i;
end
group=[group1 group2];
figure1 = figure('PaperSize',[1 1],'PaperUnits','normalized',...
            'NumberTitle','on',...
            'Name','Grafik Interaksi',...
            'menubar','none',...
```

```
             'Color',[0.8 0.8 0.8]);
interactionplot(respon,group,'varnames',{'Lokasi ODF','Jenis ODF'})
%Plotting Pengaruh Utama
figure1 = figure('PaperSize',[1 1],'PaperUnits','normalized',...
             'NumberTitle','on',...
             'Name','Pengaruh Utama',...
             'menubar','none',...
             'Color',[0.8 0.8 0.8]);
maineffectsplot(respon,{group1, group2},'varnames',{'Lokasi ODF','Jenis ODF'})
set(get(gca,'YLabel'),'String','Redaman')

% fungsi ANOVA %
function [p,anovatab,stats] = anova1(x,group,displayopt,extra)
classical = 1;
nargs = nargin;
if (nargin>0 && strcmp(x,'kruskalwallis'))
  % Fungsi kruskalwallis adjust input
  classical = 0;
  if (nargin >= 2), x = group; group = []; end
  if (nargin >= 3), group = displayopt; displayopt = []; end
  if (nargin >= 4), displayopt = extra; end
  nargs = nargs-1;
end
error(nargchk(1,3,nargs,'struct'));

if (nargs < 2), group = []; end
if (nargs < 3), displayopt = 'on'; end

willdisplay = ~(strcmp(displayopt,'nodisplay') | strcmp(displayopt,'n') ...
             | strcmp(displayopt,'off'));

% Merubah group ke cell array dari karakter array dan membuatnya dalam 1
kolom
if (ischar(group) && ~isempty(group)), group = cellstr(group); end
if (size(group, 1) == 1), group = group'; end

% Mengambil ukuran group dan mean/rata-rata
    k = find(groupnum == j);
    lk = length(k);
    countx(j) = lk;
    xm(j) = mean(xr(k));       % kolom mean

    if (willdisplay)           % boxplot
      M(1:lk,j) = xorig(k);
    end
  end

  gm = mean(xr);                    % grand mean
```

```
  df1 = sum(countx>0) - 1;          % dof kolom
  df2 = lx - df1 - 1;               % dof error
  xc = xm - gm;
  xc(countx==0) = 0;
  RSS = dot(countx, xc.^2);         % Regressi Sum of Squares
else

  [r,c] = size(x);
  lx = r * c;
  if (classical)
    xr = x;
    mu = mean(xr(:));
    xr = xr - mu;
  else
    [xr,tieadj] = tiedrank(x(:));
    xr = reshape(xr, size(x));
  end
  countx = repmat(r, 1, c);
  xorig = x;                  % save boxplot
  xm = mean(xr);              % kolom mean/rata-rata
  gm = mean(xm);              % grand mean
  df1 = c-1;                  % dof kolom
  df2 = c*(r-1);              % dof error
  RSS = r*(xm - gm)*(xm-gm)';       % Regressi Sum of Squares
end

TSS = (xr(:) - gm)'*(xr(:) - gm);  % Total Sum of Squares
SSE = TSS - RSS;                   % Error Sum of Squares

if (df2 > 0)
  mse = SSE/df2;
else
  mse = NaN;
end

if (classical)
  if (SSE~=0)
    F = (RSS/df1) / mse;
    p = 1 - fcdf(F,df1,df2);     % Probability dari F
  elseif (RSS==0)
    F = NaN;
    p = NaN;
  else
    F = Inf;
    p = 0;
  end
else
  F = (12 * RSS) / (lx * (lx+1));
```

```
  if (tieadj > 0)
    F = F / (1 - 2 * tieadj/(lx^3-lx));
  end
  p = 1 - chi2cdf(F, df1);
end

Table=zeros(3,5);              %Format Tabel ANOVA
Table(:,1)=[ RSS SSE TSS]';
Table(:,2)=[df1 df2 df1+df2]';
Table(:,3)=[ RSS/df1 mse Inf ]';
Table(:,4)=[ F Inf Inf ]';
Table(:,5)=[ p Inf Inf ]';

colheads = ['Source       ';'         SS  ';'          df ';...
          '      MS    ';'         F  ';'     Prob>F  '];
if (~classical)
  colheads(5,:) = '     Chi-sq  ';
  colheads(6,:) = '  Prob>Chi-sq';
end
rowheads = ['Brand FO   ';'Error      ';'Total      '];
if (grouped)
  rowheads(1,:) = 'Groups     ';
end

% Tabel cell array
atab = num2cell(Table);
for i=1:size(atab,1)
  for j=1:size(atab,2)
    if (isinf(atab{i,j}))
      atab{i,j} = [];
    end
  end
end
atab = [cellstr(strjust(rowheads, 'left')), atab];
atab = [cellstr(strjust(colheads, 'left'))'; atab];
if (nargout > 1)
  anovatab = atab;
end

% Multicompare
if (nargout > 2)
  if (length(gnames) > 0)
    stats.gnames = gnames;
  else
    stats.gnames = strjust(num2str((1:length(xm))'),'left');
  end
  stats.n = countx;
  if (classical)
```

```
    stats.source = 'anova1';
    stats.means = xm + mu;
    stats.df = df2;
    stats.s = sqrt(mse);
  else
    stats.source = 'kruskalwallis';
    stats.meanranks = xm;
    stats.sumt = 2 * tieadj;
  end
end

if (~willdisplay), return; end

digits = [-1 -1 0 -1 2 4];
if (classical)
  wtitle = 'Karakteristik redaman bending untuk tiap brand FO';
  ttitle = 'Tabel ANOVA';
else
  wtitle = 'ANOVA Kruskal-Wallis';
  ttitle = 'Tabel ANOVA Kruskal-Wallis';
end
tblfig = statdisptable(atab, wtitle, ttitle, '', digits);
set(tblfig,'tag','table','NumberTitle','on');

f1 = figure('pos',get(gcf,'pos') + [0,-200,0,0],'tag','boxplot','Name','Diagram Box
Plot','NumberTitle','on');
ax = axes('Parent',f1);
if (~grouped)
  boxplot(ax,xorig,'notch','on');
else
  boxplot(ax,M,'notch','on');
  h = get(ax,'XLabel');
  set(h,'String','Group Number');
end

colmean = mean(Y,1);        % rata-rata kolom
rowmean = mean(Y,2)';       % rata-rata baris
gm = mean(colmean);         % rata-rata total
df1 = c-1;                  % dof kolom
df2 = m-1;                  % dof baris
if reps == 1,
  edf = (c-1)*(r-1);        % dof error tanpa replikasi
else
  edf = (c*m*(reps-1));     % dof error dengan replikasi
  idf = (c-1)*(m-1);        % dof interaksi
end
CSS = m*reps*(colmean - gm)*(colmean-gm)';          % Sum of Squares kolom
RSS = c*reps*(rowmean - gm)*(rowmean-gm)';          % Sum of Squares baris
```

```
correction = (c*m*reps)*gm^2;
TSS = sum(sum(X .* X)) - correction;                  % Sum of Squares total
ISS = reps*sum(sum(Y .* Y)) - correction - CSS - RSS;   % Sum of Squares
interaksi
if reps == 1,
  SSE = ISS;
else
  SSE = TSS - CSS - RSS - ISS;                        % Sum of Squares error
end

ip = NaN;
if (SSE~=0)
    MSE  = SSE/edf;
    colf = (CSS/df1) / MSE;
    rowf = (RSS/df2) / MSE;
    colp = 1 - fcdf(colf,df1,edf);  % Probability F untuk rata-rata kolom
    rowp = 1 - fcdf(rowf,df2,edf);  % Probability F untuk rata-rata baris
    p    = [colp rowp];

    if (reps > 1),
       intf = (ISS/idf)/MSE;
       ip   = 1 - fcdf(intf,idf,edf);
       p    = [p ip];
    end
else
    if (edf > 0)
       MSE = 0;
    else
       MSE = NaN;
    end
    if CSS==0,         % Tidak ada variansi antar kolom
      colf = NaN;
      colp = NaN;
    else               % Ada variansi antar kolom
      colf = Inf;
      colp = 0;
    end

    if RSS==0,         % Tidak ada variansi antar baris
      rowf = NaN;
      rowp = NaN;
    else               % Ada variansi antar baris
      rowf = Inf;
      rowp = 0;
    end

    p = [colp rowp];
```

```
    if (reps>1) && (ISS==0)  % Replikasi tapi tidak ada interaksi
      intf = NaN;
      p = [p NaN];
    elseif (reps>1)        % Replikasi dengan interaksi
      intf = Inf;
      p = [p 0];
    end
end

if (reps > 1),
  Table{6,6} = [];   %Format tabel ANOVA Table dengan interaksi
  Table(2:6,1)={'Jenis Splicer';'Brand FO';'Interaction';'Error';'Total'};
  Table(2:6,2)={CSS; RSS; ISS; SSE; TSS};
  Table(2:6,3)={df1; df2; idf; edf; r*c-1};
  Table(2:5,4)={CSS/df1; RSS/df2; ISS/idf; SSE/edf;};
  Table(2:4,5)={colf; rowf; intf};
else
  Table{5,6} = [];   %Format tabel ANOVA tanpa interaksi
  Table(2:5,1)={'Jenis Splicer';'Brand FO';'Error';'Total'};
  Table(2:5,2)={CSS; RSS; SSE; TSS};
  Table(2:5,3)={df1; df2; edf; r*c-1};
  Table(2:4,4)={CSS/df1; RSS/df2; SSE/edf;};
  Table(2:3,5)={colf; rowf};
end
Table(1,1:6) = {'Source' 'SS' 'df' 'MS' 'F' 'Prob>F'};
Table(2:(1+length(p)),6) = num2cell(p);

if (isequal(displayopt, 'on'))
   digits = [-1 -1 0 -1 2 4];
   statdisptable(Table, 'Karakteristik redaman splicing dari tiap splicer untuk tiap
brand FO', 'Tabel ANOVA', '', digits);
end

if (nargout > 2)
   stats.source = 'anova2';
   stats.sigmasq = MSE;
   stats.colmeans = colmean;   % rata-rata kolom
   stats.coln = m*reps;        % n untuk estimasi rata-rata kolom
   stats.rowmeans = rowmean;
   stats.rown = c*reps;
   stats.inter = (reps>1);
   stats.pval = ip;            % p-value interaksi
   stats.df = edf;
end

function [p,Table,stats] = anova2(X,reps,displayopt)

if (nargin<3), displayopt = 'on'; end
```

```
if (nargin < 1)
   error('stats:anova2:TooFewInputs','At least one input is required.');
end
if (any(isnan(X(:))))
   error('stats:anova2:DataNotBalanced',...
         'NaN values in input not allowed.  Use anovan instead.');
end
[r,c] = size(X);
if nargin == 1,
  reps = 1;
  m=r;
  Y = X;
elseif reps == 1
  m=r;
  Y = X;
else
  m = r/reps;
  if (floor(m) ~= r/reps),
     error('stats:anova2:BadSize',...
           'The number of rows must be a multiple of reps.');
  end
  Y = zeros(m,c);
  for i=1:m,
     j = (i-1)*reps;
     Y(i,:) = mean(X(j+1:j+reps,:));
  end
end
colmean = mean(Y,1);        % rata-rata kolom
rowmean = mean(Y,2)';       % rata-rata baris
gm = mean(colmean);         % rata-rata total
df1 = c-1;                  % dof kolom
df2 = m-1;                  % dof baris
if reps == 1,
  edf = (c-1)*(r-1);        % dof error tanpa replikasi
else
  edf = (c*m*(reps-1));     % dof error dengan replikasi
  idf = (c-1)*(m-1);        % dof interaksi
end
CSS = m*reps*(colmean - gm)*(colmean-gm)';           % Sum of Squares kolom
RSS = c*reps*(rowmean - gm)*(rowmean-gm)';           % Sum of Squares baris
correction = (c*m*reps)*gm^2;
TSS = sum(sum(X .* X)) - correction;                 % Sum of Squares total
ISS = reps*sum(sum(Y .* Y)) - correction - CSS - RSS;  % Sum of Squares
interaksi
if reps == 1,
  SSE = ISS;
else
  SSE = TSS - CSS - RSS - ISS;                       % Sum of Squares error
```

```
end

ip = NaN;
if (SSE~=0)
   MSE  = SSE/edf;
   colf = (CSS/df1) / MSE;
   rowf = (RSS/df2) / MSE;
   colp = 1 - fcdf(colf,df1,edf);  % Probability F untuk rata-rata kolom
   rowp = 1 - fcdf(rowf,df2,edf);  % Probability F untuk rata-rata baris
   p    = [colp rowp];

   if (reps > 1),
     intf = (ISS/idf)/MSE;
     ip   = 1 - fcdf(intf,idf,edf);
     p    = [p ip];
   end

else
   if (edf > 0)
     MSE = 0;
   else
     MSE = NaN;
   end
   if CSS==0,          % Tidak ada variansi antar kolom
    colf = NaN;
    colp = NaN;
   else              % Ada variansi antar kolom
    colf = Inf;
    colp = 0;
   end

   if RSS==0,          % Tidak ada variansi antar baris
    rowf = NaN;
    rowp = NaN;
   else              % Ada variansi antar baris
    rowf = Inf;
    rowp = 0;
   end

   p = [colp rowp];

   if (reps>1) && (ISS==0)  % Replikasi tapi tidak ada interaksi
    intf = NaN;
    p = [p NaN];
   elseif (reps>1)        % Replikasi dengan interaksi
    intf = Inf;
    p = [p 0];
   end
```

```
end

if (reps > 1),
  Table{6,6} = [];   %Format tabel ANOVA Table dengan interaksi
  Table(2:6,1)={'Jenis ODF    ';'Lokasi  ';'Interaction';'Error';'Total'};
  Table(2:6,2)={CSS; RSS; ISS; SSE; TSS};
  Table(2:6,3)={df1; df2; idf; edf; r*c-1};
  Table(2:5,4)={CSS/df1; RSS/df2; ISS/idf; SSE/edf;};
  Table(2:4,5)={colf; rowf; intf};
else
  Table{5,6} = [];   %Format tabel ANOVA tanpa interaksi
  Table(2:5,1)={'Jenis ODF    ';'Lokasi  ';'Error';'Total'};
  Table(2:5,2)={CSS; RSS; SSE; TSS};
  Table(2:5,3)={df1; df2; edf; r*c-1};
  Table(2:4,4)={CSS/df1; RSS/df2; SSE/edf;};
  Table(2:3,5)={colf; rowf};
end

Table(1,1:6) = {'Source' 'SS' 'df' 'MS' 'F' 'Prob>F'};
Table(2:(1+length(p)),6) = num2cell(p);

if (isequal(displayopt, 'on'))
   digits = [-1 -1 0 -1 2 4];
 statdisptable(Table, 'Karakteristik redaman patching dari tiap ODF untuk tiap
lokasi penempatan', 'Tabel ANOVA', '', digits);
end

if (nargout > 2)
   stats.source = 'anova2';
   stats.sigmasq = MSE;
   stats.colmeans = colmean;   % rata-rata kolom
   stats.coln = m*reps;        % n untuk estimasi rata-rata kolom
   stats.rowmeans = rowmean;
   stats.rown = c*reps;
   stats.inter = (reps>1);
   stats.pval = ip;            % p-value interaksi
   stats.df = edf;
end

%========== Tes data bending ==========%
baca_data_bending =xlsread('data_bending_final.xlsx');
figure
subplot(3,1,1);
normplot(baca_data_bending(:,1));
subplot(3,1,2);
normplot(baca_data_bending(:,2));
subplot(3,1,3);
normplot(baca_data_bending(:,3));
```

```
set(gcf,'numbertitle','on','name','Plot Normalitas Data Bending')

%========== Tes data splicing ==========%
baca_data_splicing =xlsread('data_splicing_final.xlsx');
subplot(3,2,1);
normplot(baca_data_splicing(:,1));
subplot(3,2,2);
normplot(baca_data_splicing(:,4));
subplot(3,2,3);
normplot(baca_data_splicing(:,2));
subplot(3,2,4);
normplot(baca_data_splicing(:,5));
subplot(3,2,5);
normplot(baca_data_splicing(:,3));
subplot(3,2,6);
normplot(baca_data_splicing(:,6));
set(gcf,'numbertitle','on','name','Plot Normalitas Data Splicing')

%========== Tes data patching ==========%
baca_data_patching =xlsread('data_patching_final.xlsx');
figure
subplot(2,2,1);
normplot(baca_data_patching(:,1));
subplot(2,2,2);
normplot(baca_data_patching(:,3));
subplot(2,2,3);
normplot(baca_data_splicing(:,2));
subplot(2,2,4);
normplot(baca_data_patching(:,4));
set(gcf,'numbertitle','on','name','Plot Normalitas Data Patching')

% ==== uji varian ==========%
function [p,stats] = vartestn(x,group,displayopt,testtype)


error(nargchk(1,4,nargin,'struct'));

if (nargin < 2), group = []; end
if (nargin < 3) || isempty(displayopt)
  displayopt = 'on';
elseif ~isequal(displayopt,'on') && ~isequal(displayopt,'off')
  error('stats:vartestn:BadDisplayOpt',...
      'DISPLAYOPT must be "on" or "off".');
end
dodisplay = isequal(displayopt,'on');

if nargin < 4 || isempty(testtype)
  dorobust = false;
```

```
else
  if ischar(testtype)
     i = find(strncmpi(testtype,{'robust';'classical'},length(testtype)));
  else
     i = [];
  end
  if isempty(i)
     error('stats:vartestn:BadTestType', ...
        'TESTTYPE must be one of the strings "robust" or "classical".');
  end
  dorobust = (i==1);
end

% Error jika data tidak ada
if isempty(x)
  error('stats:vartestn:NoData','X must not be empty');
end

% Merubah group ke array
if (ischar(group) && ~isempty(group)), group = cellstr(group); end
if (size(group, 1) == 1), group = group'; end

% jika x matrix,  ubah ke bentuk vektor.
[n,m] = size(x);
if isempty(group)
  x = x(:);
  group = reshape(repmat((1:m), n, 1), n*m, 1);
elseif isvector(x)
  if numel(group) ~= n
     error('stats:vartestn:InputSizeMismatch',...
          'X and GROUP must have the same length.');
  end
else
  error('stats:vartestn:BadGroup',...
        'Cannot specify a GROUP argument unless X is a vector.');
end

% NaN
t = isnan(x);
if any(t)
  x(t) = [];
  group(t,:) = [];
end

% Menghitung summary group
[igroup,gname] = grp2idx(group);
[gmean,gsem,gcount] = grpstats(x,igroup);
df = gcount-1;              % group degrees of freedom
```

```
gvar = gcount .* gsem.^2;       % group variances
sumdf = sum(df);
if sumdf>0
   vp = sum(df.*gvar) / sumdf;  % pooled variance
else
   vp = NaN;
end
k = length(df);

if dorobust
   % Levene's test

   spgroups = find(gcount<2);
   t = ~ismember(igroup,spgroups);
   xc = x(t) - gmean(igroup(t));
   ngroups = length(gcount) - length(spgroups);

   % ANOVA
   if ngroups>1
      [p,atab] = anova1(xc.^2, igroup,'off');
      B = atab{2,5};                          % F statistik
      Bdf = [atab{2,3}, atab{3,3}];           % d.f.
   else
      p = NaN;
      B = NaN;
      Bdf = [0, length(xc)-ngroups];
   end
   Bdftable = sprintf('%d, %d',Bdf(1),Bdf(2));
   testname = 'Levene''s statistic';
   statname = 'fstat';
else
   % Bartlett's test
   Bdf = max(0, sum(df>0)-1);
   t = df>0;
   if Bdf>0 && sumdf>0
      B = log(vp) * sum(df) - sum(df(t).*log(gvar(t)));
      C = 1 + (sum(1./df(t)) - 1/sum(df))/(3*Bdf);
      B = B/C;
   else
      B = NaN;
   end

   p = 1 - chi2cdf(B,Bdf);
   Bdftable = Bdf;
   testname = 'Bartlett''s statistic';
   statname = 'chisqstat';
end
```

```
if dodisplay
   Table = cell(k+6,4);
   Table(1,:) = {'Group' 'Count' 'Mean' 'Std Dev'};
   Table(2:k+1,1) = gname;
   Table(2:k+1,2) = num2cell(gcount);
   Table(2:k+1,3) = num2cell(gmean);
   Table(2:k+1,4) = num2cell(sqrt(gvar));

   Table{k+2,1} = 'Pooled';
   Table{k+2,2} = sum(gcount);
   Table{k+2,3} = sum(gcount.*gmean) / sum(gcount);
   Table{k+2,4} = sqrt(vp);

   Table{k+3,1} = ' ';
   Table{k+4,1} = testname;
   Table{k+4,2} = B;
   Table{k+5,1} = 'Degrees of freedom';
   Table{k+5,2} = Bdftable;
   Table{k+6,1} = 'p-value';
   Table{k+6,2} = p;

   tblfig = statdisptable(Table, 'Variance Test','Tabel Uji Varians');
   set(tblfig,'tag','table');
end

% Membuat struktur output
if (nargout > 1)
   stats = struct(statname,B,'df',Bdf);
end

if ~dodisplay
   return;
end

% Membuat boxplot
f1 = figure('pos',get(gcf,'pos') + [0,-200,0,0],'tag','boxplot');
ax = axes('Parent',f1);
boxplot(ax,x,group);

% ====== Fungsi Multicompare======%

function [comparison,means,h,gnames] = multcompare(stats,varargin)
if (nargin < 1)
   error('stats:multcompare:TooFewInputs',...
       'At least one argument is required.');
end
if isempty(varargin) || (~isempty(varargin{1}) && ischar(varargin{1}))
   okargs =   {'alpha' 'display' 'ctype' 'estimate' 'dimension'};
```

```
  defaults = {0.05    'on'     ''     ''       []};
  [eid,emsg,alpha,displayopt,ctype,estimate,dim] = ...
            statgetargs(okargs,defaults,varargin{:});
  if ~isempty(emsg)
    error(sprintf('stats:multcompare:%s',eid),emsg);
  end
else

  if (nargin >= 2) && ~isempty(varargin{1})
    alpha = varargin{1};
  else
    alpha = 0.05;
  end
  if (nargin >= 3) && ~isempty(varargin{2})
    displayopt = varargin{2};
  else
    displayopt = 'on';
  end
  if (nargin>=4) && ~isempty(varargin{3})
    ctype = varargin{3};
  else
    ctype = '';
  end
  if (nargin>=5) && ~isempty(varargin{4})
    estimate = varargin{4};
  else
    estimate = '';
  end
  if (nargin>=6) && ~isempty(varargin{5})
    dim = varargin{5};
  else
    dim = [];
  end
end

if (~isstruct(stats))
  error('stats:multcompare:BadStats','STATS must be a structure.');
end
if (~isfield(stats,'source'))
  error('stats:multcompare:BadStats',...
      'The first argument is not a STATS output structure from ANOVA1.');
end
if (length(alpha)~=1 || ~isfinite(alpha) || alpha<=0 || alpha>=1)
  error('stats:multcompare:BadAlpha',...
      'ALPHA must be a scalar between 0 and 1, exclusive.');
end
if ~(isequal(displayopt, 'on') || isequal(displayopt, 'off'))
  error('stats:multcompare:BadDisplayOpt',...
```

```
      'DISPLAYOPT must be "on" or "off".');
end
dodisp = isequal(displayopt, 'on');

source = stats.source;

switch(source)
 case 'anova1'
   mname = 'means';
   gmeans = stats.means(:);
   gnames = stats.gnames;
   n = stats.n(:);
   df = stats.df;
   s = stats.s;
   ng = sum(n>0);
   if (df < 1)
      error('stats:multcompare:NotEnoughData',...
         'Cannot compare means with 0 degrees of freedom for error.');
   end

   % Tukey-Kramer
   if (isempty(ctype)), ctype = 'tukey-kramer'; end
   crit = getcrit(ctype, alpha, df, ng);

   gcov = diag((s^2)./n);

 case 'anova2'
   docols = logical(1);
   if (~isempty(estimate))
      if ~(   isequal(estimate,'row') || isequal(estimate,'column')...
          || isequal(estimate,'r')   || isequal(estimate,'c'))
         error('stats:multcompare:BadEstimate',...
            'ESTIMATE must be "column" or "row".');
      end
      docols = isequal(estimate, 'column') || isequal(estimate, 'c');
   end
   if (docols)
      gmeans = stats.colmeans(:);
      n = stats.coln(:);
      mname = 'column means';
   else
      gmeans = stats.rowmeans(:);
      n = stats.rown(:);
      mname = 'row means';
   end
   ng = length(gmeans);
   sigma = sqrt(stats.sigmasq);
   gnames = strjust(num2str((1:ng)'), 'left');
```

```
    df = stats.df;
    if (df < 1)
      error('stats:multcompare:NotEnoughData',...
          'Cannot compare means with 0 degrees of freedom for error.');
    end

    % Tukey-Kramer
    if (isempty(ctype)), ctype = 'tukey-kramer'; end
    crit = getcrit(ctype, alpha, df, ng);

    gcov = ((sigma^2)/n) * eye(ng);

    if (stats.inter && dodisp)
      if (stats.pval < alpha)
        disp(sprintf(...
  ['Note:  Your model includes an interaction term that is significant\n'...
   'at the level you specified.  Testing main effects under these\n'...
   'conditions is questionable.']));
      else
        disp(sprintf(...
  ['Note:  Your model includes an interaction term.  A test of main\n'...
   'effects can be difficult to interpret when the model includes\n'...
   'interactions.']));
      end
    end

   case 'anovan'
    mname = 'population marginal means';

    try
       vnested = stats.vnested;
    catch
       vnested = [];
    end
    if any(vnested(:))
       error('stats:multcompare:NoNesting',...
           'MULTCOMPARE does not support models with nested factors.');
    end


    if ~isempty(stats.ems)
      warning('stats:multcompare:IgnoringRandomEffects',...
           'Ignoring random effects (all effects treated as fixed).')
    end

    nvars = length(stats.nlevels);
    P0 = stats.nullproject;
```

```
if isempty(dim)
    dim = find(stats.nlevels>1,1);
end
dim = dim(:);
if isempty(dim) || any(dim<1 | dim>nvars | dim~=round(dim))
   error('stats:multcompare:BadDim',...
         'Values of DIM must be integers between 1 and %d.',nvars);
end
dim = sort(dim);
dim(diff(dim)==0) = [];
if any(stats.nlevels(dim)<2)
   error('stats:multcompare:BadDim',...
         'DIM must specify only categorical factors with 2 or more degrees of
freedom.');
end

try
    continuous = stats.continuous;
catch
    continuous = zeros(nvars,1);
end
ffdesign = fullfact(stats.nlevels(dim));
nrows = size(ffdesign,1);

dums = cell(nvars, 1);
for j=1:length(dim)
   dj = dim(j);
   dums{dj} = idummy(ffdesign(:,j),3);
end

for j=1:nvars
   if isempty(dums{j});
       if continuous(j)
          dums{j} = repmat(stats.vmeans(j),nrows,1);
       else
          nlev = stats.nlevels(j);
          dums{j} = repmat(1/nlev,nrows,nlev);
       end
   end
end

termcols = stats.termcols(:);
termstart = cumsum([1; termcols]);
terms = [zeros(1,nvars); stats.terms];
ncols = sum(termcols);
x = zeros(size(ffdesign,1), ncols);
x(:,1) = 1;
for j=1:length(termcols)
```

```
    tm = terms(j,:);
    t0 = termstart(j);
    t1 = termstart(j) + termcols(j) - 1;
    if all(tm==0)
      x(:,t0:t1) = 1;
    else
      x0 = [];
      for k=nvars:-1:1
        if tm(k)
          x0 = termcross(x0,dums{k});
        end
      end
      x(:,t0:t1) = x0;
    end
end

% Estimasi error
gmeans = x * stats.coeffs;
xproj = (x*P0)';
tmp = stats.Rtr \ xproj;
if (stats.dfe == 0)
  mse = NaN;
else
  mse = max(stats.mse,0);
end
gcov = mse * tmp' * tmp;

Xrows = stats.rowbasis';
bb = Xrows \ (x');
xres = Xrows * bb - x';
xres = sum(abs(xres));
cutoff = sqrt(eps(class(xres))) * size(xres,2);
gmeans(xres > cutoff) = NaN;

% Tukey-Kramer
if (isempty(ctype)), ctype = 'tukey-kramer'; end
crit = getcrit(ctype, alpha, stats.dfe, length(gmeans));


ngroups = size(ffdesign,1);
gnames = cell(ngroups,1);
allnames = stats.grpnames;
varnames = stats.varnames;
for j=1:ngroups
  v1 = dim(1);
  vals1 = allnames{v1};
  nm = sprintf('%s=%s',varnames{v1},vals1{ffdesign(j,1)});
  for i=2:size(ffdesign,2)
```

```
      v2 = dim(i);
      vals2 = allnames{v2};
      nm = sprintf('%s,%s=%s',nm,varnames{v2},vals2{ffdesign(j,i)});
    end
    gnames{j} = nm;
  end

case 'aoctool'
  model = stats.model;
  if (model==1 || model==3)
    error('stats:multcompare:NoMultipleParameters',...
        'No multiple comparisons possible from this aoctool fit.');
  end
  gnames = stats.gnames;
  n = stats.n(:);
  ng = length(n);
  df = stats.df;
  if (df < 1)
    error('stats:multcompare:NotEnoughData',...
        'Cannot do multiple comparison with 0 d.f. for error.');
  end

  % Estimasi intercept dan covariance
  if (isempty(estimate))
    if (model == 5)
      estimate = 'slope';
    else
      estimate = 'intercept';
    end
  end

  if (isequal(estimate,'s')), estimate = 'slope'; end
  if (isequal(estimate,'i')), estimate = 'intercept'; end
  if (isequal(estimate,'p')), estimate = 'pmm'; end
  if (~isempty(estimate))
    if ~(   isequal(estimate,'slope') || isequal(estimate,'pmm')...
       || isequal(estimate,'intercept'))
      error('stats:multcompare:BadEstimate',...
          'ESTIMATE must be "slope", "intercept", or "pmm".');
    end
  end
  switch(estimate)
   case 'slope'
    if (~isfield(stats, 'slopes'))
      error('stats:multcompare:BadStats',...
          'No slope estimates available in STATS argument.');
    end
    gmeans = stats.slopes;
```

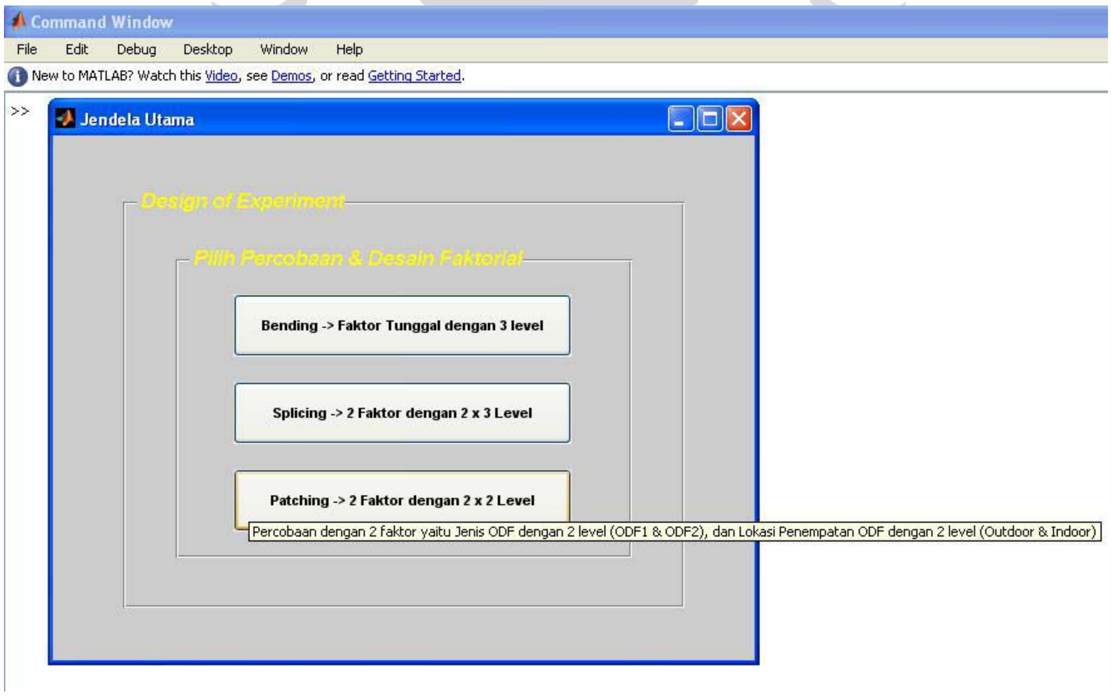
```
    gcov = stats.slopecov;
    mname = 'slopes';
   case 'intercept'
    if (~isfield(stats, 'intercepts'))
      error('stats:multcompare:BadStats',...
           'No intercept estimates available in STATS argument.');
    end
    gmeans = stats.intercepts;
    gcov = stats.intercov;
    mname = 'intercepts';
   case 'pmm'
    gmeans = stats.pmm;
    gcov = stats.pmmcov;
    mname = 'population marginal means';
  end

  if (any(any(isinf(gcov))))
    error('stats:multcompare:InfiniteVariance',...
        'Cannot do comparisons.  Some %s have infinite variance.', mname);
  end

  % Tukey-Kramer critical value
  if (isempty(ctype)), ctype = 'tukey-kramer'; end
  crit = getcrit(ctype, alpha, df, ng);

 case 'kruskalwallis'
  gmeans = stats.meanranks(:);
  gnames = stats.gnames;
  n = stats.n(:);
  sumt = stats.sumt;
  ng = length(n);
  N = sum(n);
  mname = 'mean ranks';

  if (isempty(ctype)), ctype = 'tukey-kramer'; end
  crit = getcrit(ctype, alpha, Inf, ng);

  gcov = diag(((N*(N+1)/12) - (sumt/(12*(N-1)))) ./ n);

  if (dodisp)
    disp(['Note:  Intervals can be used for testing but are not ' ...
        'simultaneous confidence intervals.']);
  end

 case 'friedman'
  gmeans = stats.meanranks(:);
  n = stats.n;
  ng = length(gmeans);
```

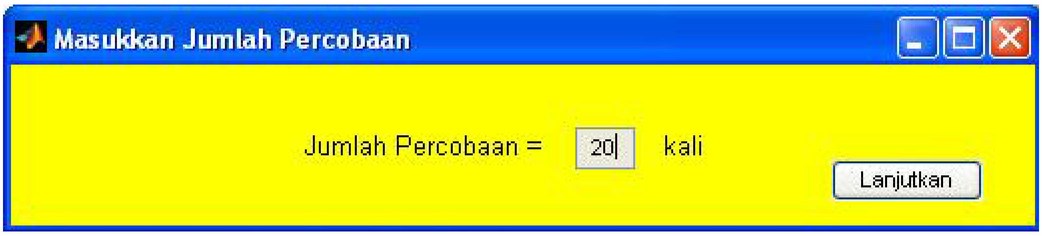
```
  sigma = stats.sigma;
  mname = 'mean column ranks';
  gnames = strjust(num2str((1:ng)'), 'left');

  % Tukey-Kramer
  if (isempty(ctype)), ctype = 'tukey-kramer'; end
  crit = getcrit(ctype, alpha, Inf, ng);

  gcov = ((sigma^2) / n) * eye(ng);

  if (dodisp)
    disp(['Note:  Intervals can be used for testing but are not ' ...
          'simultaneous confidence intervals.']);
  end

 otherwise
  error('stats:multcompare:BadStats','Bad SOURCE field in STATS structure');
end


 [M,MM,hh] = makeM(gmeans, gcov, crit, gnames, mname, dodisp);

comparison = M;
if (nargout>1), means = MM; end
if (nargout>2), h = hh; end

% -----------------------------------------------
function crit = getcrit(ctype, alpha, df, ng)

crit = Inf;
[onetype,ctype] = strtok(ctype);

while(~isempty(onetype))
  if (length(onetype) == 1)
    switch onetype
     case 't', onetype = 'tukey-kramer';
     case 'd', onetype = 'dunn-sidak';
     case 'b', onetype = 'bonferroni';
     case 's', onetype = 'scheffe';
     case 'h', onetype = 'tukey-kramer';
     case 'l', onetype = 'lsd';
    end
  end
  if (isequal(onetype, 'hsd')), onetype = 'tukey-kramer'; end

  switch onetype
   case 'tukey-kramer' % or hsd
    crit1 = stdrinv(1-alpha, df, ng) / sqrt(2);
```

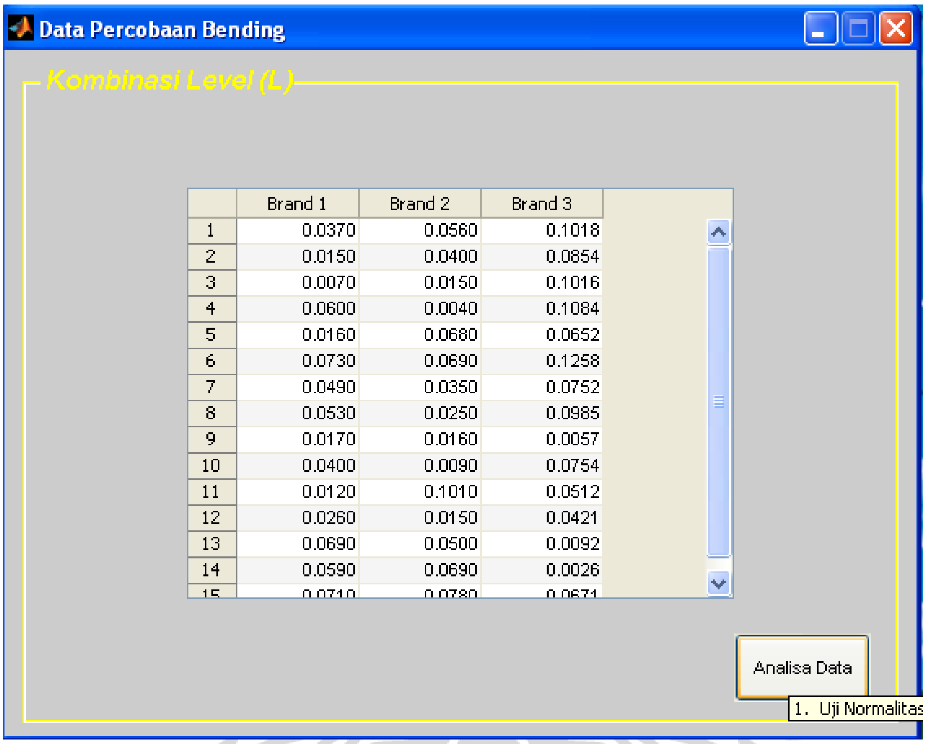
```
    ub = getcrit('dunn-sidak', alpha, df, ng);
    if (crit1 > ub), crit1 = ub; end

   case 'dunn-sidak'
    kstar = nchoosek(ng, 2);
    alf = 1-(1-alpha).^(1/kstar);
    if (isinf(df))
      crit1 = norminv(1-alf/2);
    else
      crit1 = tinv(1-alf/2, df);
    end

   case 'bonferroni'
    kstar = nchoosek(ng, 2);
    if (isinf(df))
      crit1 = norminv(1 - alpha / (2*kstar));
    else
      crit1 = tinv(1 - alpha / (2*kstar), df);
    end

   case 'lsd'
    if (isinf(df))
      crit1 = norminv(1 - alpha / 2);
    else
      crit1 = tinv(1 - alpha / 2, df);
    end

   case 'scheffe'
    if (isinf(df))
      tmp = chi2inv(1-alpha, ng-1) / (ng-1);
    else
      tmp = finv(1-alpha, ng-1, df);
    end
    crit1 = sqrt((ng-1) * tmp);

   otherwise
    error('stats:multcompare:BadCType',...
        'Unknown CTYPE (critical value type) %s.', ctype);
  end

  if (~isnan(crit1)), crit = min(crit, crit1); end
  [onetype,ctype] = strtok(ctype);
end

% -------------------------------------------
function [M,MM,hh] = makeM(gmeans, gcov, crit, gnames, mname, dodisp)
% Membuat matrix rata-rata dan tampilan
```

```
t = isnan(gmeans);
if any(t)
   gcov(t,:) = 0;
   gcov(:,t) = 0;
end

ng = length(gmeans);
MM = zeros(ng,2);
MM(:,1) = gmeans(:);
MM(:,2) = sqrt(diag(gcov));
MM(isnan(MM(:,1)),2) = NaN;

M = nchoosek(1:ng, 2);
M(1,5) = 0;
g1 = M(:,1);
g2 = M(:,2);
M(:,4) = gmeans(g1) - gmeans(g2);
i12 = sub2ind(size(gcov), g1, g2);
gvar = diag(gcov);
d12 = sqrt(gvar(g1) + gvar(g2) - 2 * gcov(i12));
delta = crit * d12;
M(:,3) = M(:,4) - delta;
M(:,5) = M(:,4) + delta;

if (dodisp)
  d = zeros(ng, ng);
  d(i12) = d12;
  sum1 = sum(sum(d));
  d = d + d';
  sum2 = sum(d);
  if (ng > 2)
    w = ((ng-1) * sum2 - sum1) ./ ((ng-1)*(ng-2));
  else
    w = repmat(sum1, 2, 1) / 2;
  end
  halfwidth = crit * w(:);
  hh = meansgraph(gmeans, gmeans-halfwidth, gmeans+halfwidth, ...
             gnames, mname);
  set(hh, 'Name', sprintf('Multiple comparison of %s',mname),'NumberTitle','off');
else
  hh = [];
end
```

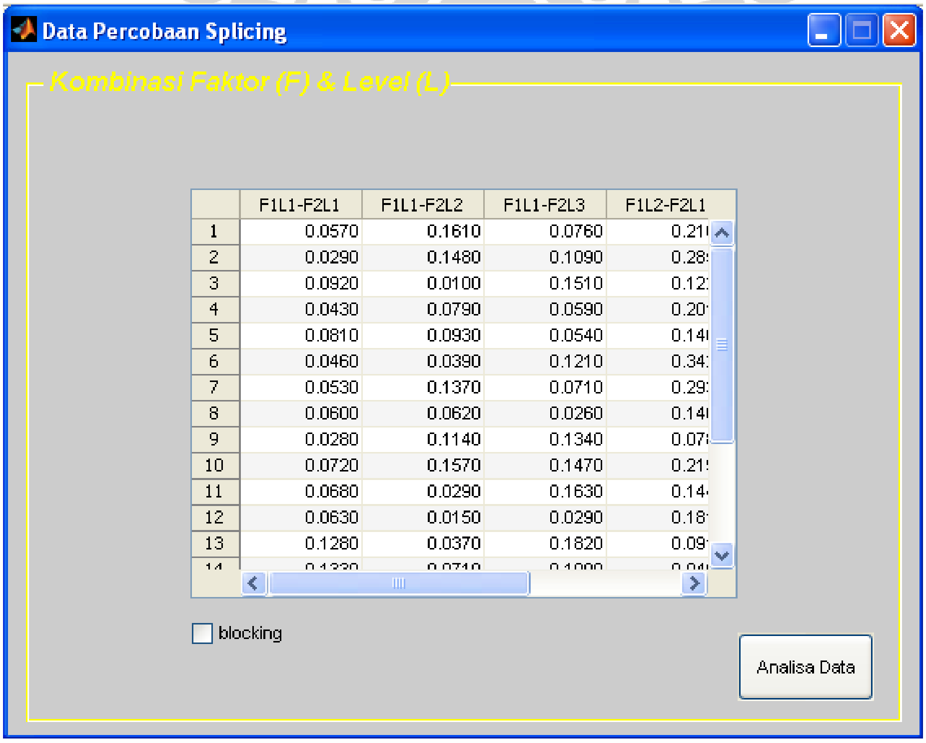
Lampiran 2. Tampilan Program Simulasi



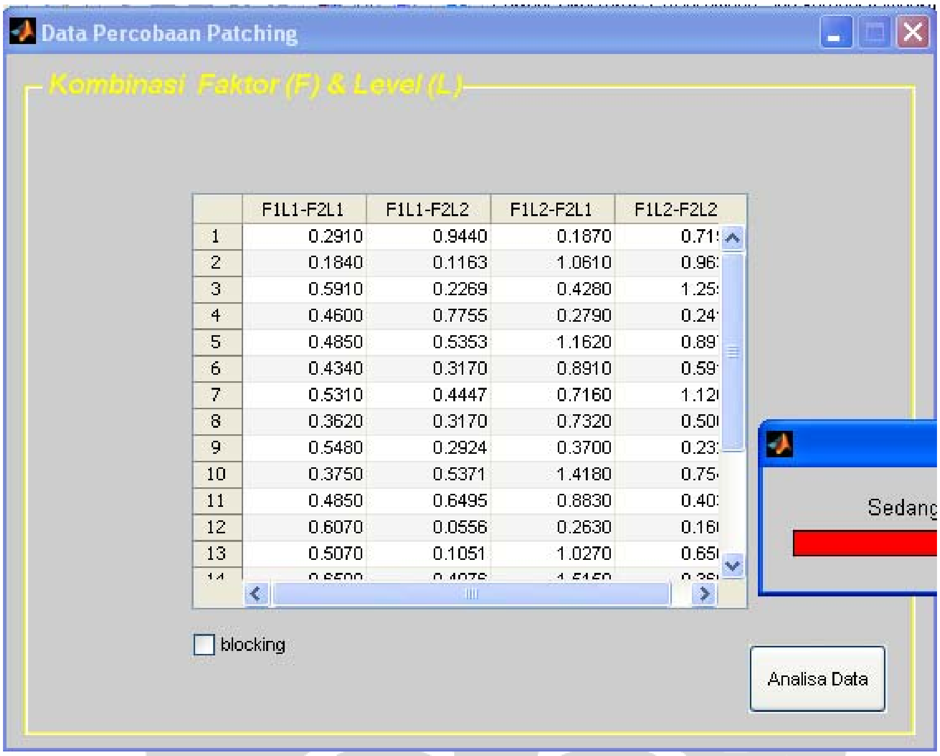




	Brand 1	Brand 2	Brand 3
1	0.0370	0.0560	0.1018
2	0.0150	0.0400	0.0854
3	0.0070	0.0150	0.1016
4	0.0600	0.0040	0.1084
5	0.0160	0.0680	0.0652
6	0.0730	0.0690	0.1258
7	0.0490	0.0350	0.0752
8	0.0530	0.0250	0.0985
9	0.0170	0.0160	0.0057
10	0.0400	0.0090	0.0754
11	0.0120	0.1010	0.0512
12	0.0260	0.0150	0.0421
13	0.0690	0.0500	0.0092
14	0.0590	0.0690	0.0026
15	0.0710	0.0780	0.0671



	F1L1-F2L1	F1L1-F2L2	F1L1-F2L3	F1L2-F2L1
1	0.0570	0.1610	0.0760	0.21
2	0.0290	0.1480	0.1090	0.28
3	0.0920	0.0100	0.1510	0.12
4	0.0430	0.0790	0.0590	0.20
5	0.0810	0.0930	0.0540	0.14
6	0.0460	0.0390	0.1210	0.34
7	0.0530	0.1370	0.0710	0.29
8	0.0600	0.0620	0.0260	0.14
9	0.0280	0.1140	0.1340	0.07
10	0.0720	0.1570	0.1470	0.21
11	0.0680	0.0290	0.1630	0.14
12	0.0630	0.0150	0.0290	0.18
13	0.1280	0.0370	0.1820	0.09
14	0.1330	0.0710	0.1000	0.04


Data Percobaan Patching
Kombinasi Faktor (F) & Level (L)
F1L1-F2L1 F1L1-F2L2 F1L2-F2L1 F1L2-F2L2
1 0.2910 0.9440 0.1870
2 0.1840 0.1163 1.0610
3 0.5910 0.2269 0.4280
4 0.4600 0.7755 0.2790
5 0.4850 0.5353 1.1620
6 0.4340 0.3170 0.8910
7 0.5310 0.4447 0.7160
8 0.3620 0.3170 0.7320
9 0.5480 0.2924 0.3700
10 0.3750 0.5371 1.4180
11 0.4850 0.6495 0.8830
12 0.6070 0.0556 0.2630
13 0.5070 0.1051 1.0270
blocking
Analisa Data
Sedang

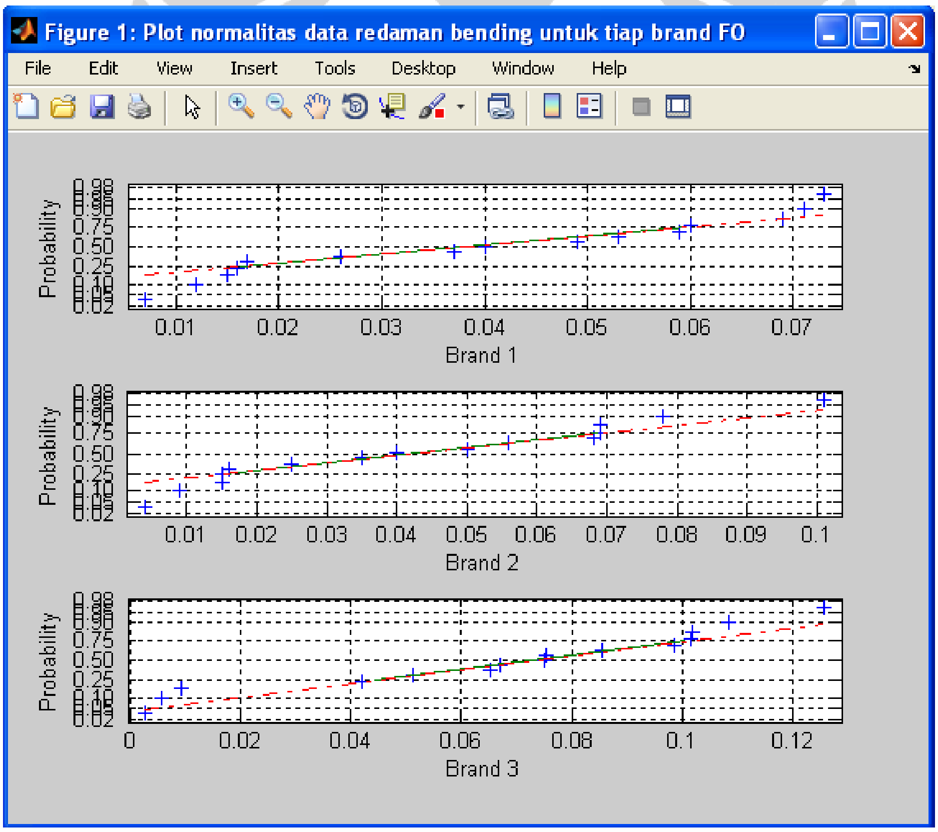


Figure 1: Plot normalitas data redaman bending untuk tiap brand FO
File Edit View Insert Tools Desktop Window Help
Probability
Brand 1
Brand 2
Brand 3

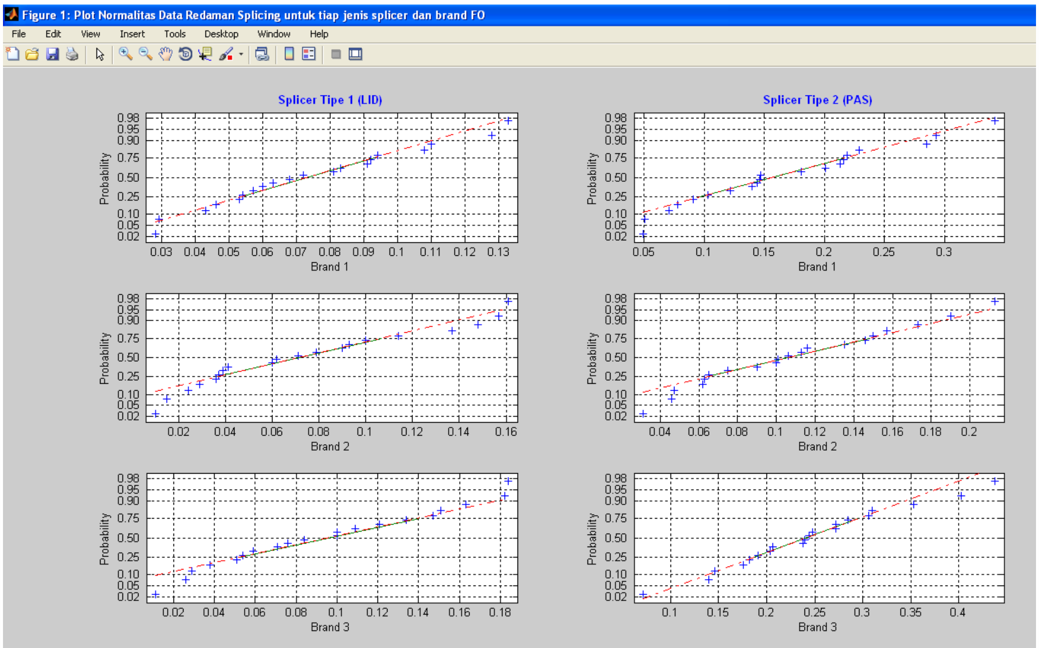


Figure 1: Plot Normalitas Data Redaman Splicing untuk tiap jenis splicer dan brand FO
File Edit View Insert Tools Desktop Window Help
Splicer Tipe 1 (LID)
Splicer Tipe 2 (PAS)
Probability
Brand 1
Brand 2
Brand 3

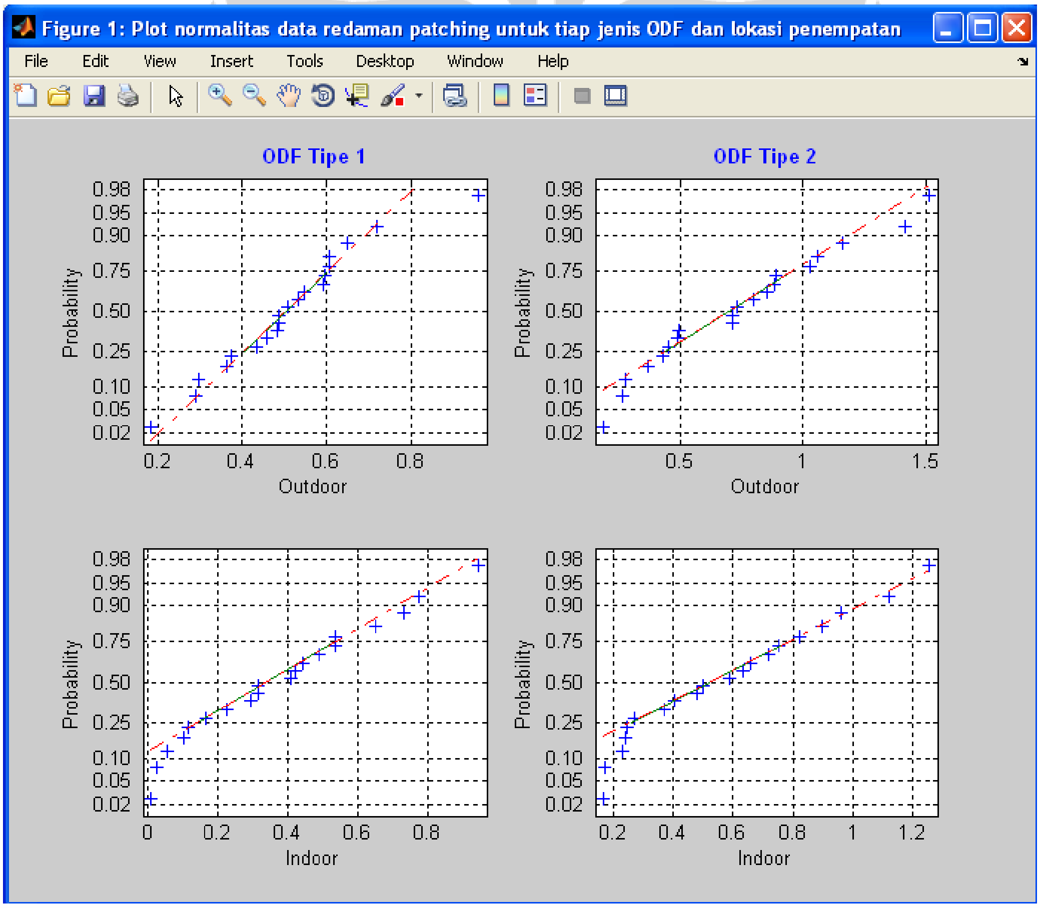


Figure 1: Plot normalitas data redaman patching untuk tiap jenis ODF dan lokasi penempatan
File Edit View Insert Tools Desktop Window Help
ODF Tipe 1
ODF Tipe 2
Probability
Outdoor
Indoor

Lampiran 3. Spesifikasi OTU-8000

Specifications

Base Unit Technical

Mechanical

Height	2 RU
Width	19-, 21- (ETSI), or 23-in
Depth	260 mm (ETSI), 300 mm (19- or 23-in)

Power Supply

DC input	–36 to –60 V
Power consumption	30 W

Environmental

Operating	–10 to 50°C
Storage	–20 to 60°C
Humidity	95% without condensing
EMI/ESD	CE Compliant

Interfaces

1 RJ45 Ethernet 10/100BaseT Port
1 RJ11 if equipped with PSTN modem
GSM if equipped with GSM modem

Storage

Media	Solid state disk
Optical links (max)	512
Alarms storage	512
OTDR Trace storage	1024

Relay Contacts (Option)

3 relays correspond respectively to unit alarm, major optical alarm, and minor optical alarm
Relay is closed in normal conditions

Nominal switching capacity	1 A at 30 VDC, 0.5 A at 125 VAC

OTDR[1]

The OTU-8000 can house two field-interchangeable OTDR modules. A wide range of OTDRs are available, ensuring optimum monitoring of all types of fiber optic networks from short-range multimode to long-haul single-mode. The OTU-8000 monitors active fibers using the 1625 nm OTDR module, which allows for factors such as the Raman effect of the optical amplifier.

Technical

Distance Unit — km, kft, miles

Group Index Range	1.30000 to 1.70000 in 0.00001 steps
No. of Data points	Up to 128 000 data points
Distance Measurement	Automatic or dual cursor
Display span	From 2.6 m up 380 km
Display resolution	1 cm
Cursor resolution	From 1 cm
Sampling resolution	From 4 cm
Accuracy	±1 m ±sampling resolution ±1.10^{-5}x distance (excluding group index uncertainties)

Attenuation Measurement

Automatic, manual, 2-points, 5-points and LSA

Display span	1.25 to 55 dB
Display resolution	0.001 dB
Cursor resolution	From 0.001 dB
Accuracy	±0.05 dB ±0.05 dB/dB

Reflectance/ORL Measurements

Automatic or manual

Display resolution	0.01 dB
Threshold	–11 to –99 dB in 1 dB step

Optical Switch

The OTU-8000 can house a field interchangeable optical switch module with up to 24 ports. If higher ports count is required, 24 ports can be extended to more than 1000 ports by adding chassis of 64 ports each. An OTU-8000 with no OTDR modules fitted forms the base of the Remote Optical Switch controlled by TCP/IP.

Upgrade the Remote Optical Switch base by adding an OTDR module to become a complete OTU-8000 at any time.

General

No. of Ports	2, 4, 8, 12, 16, 24, 36, 48, or 64 with 1 or 2 commons, more than 1000 ports with 1 common
Insertion Loss (excluding connectors)	0.6 dB up to 64 ports, 1.2 dB for higher port counts
Back-Reflection	–60 dB (Singlemode)
Repeatability	±0.01 dB
Wavelength Range	1310, 1550 and 1625 nm
Lifetime	10^7 cycles
Housing	Up to 24 ports: Included in the OTU-8000 For higher port counts: external 4 RU racks

[1]The main specifications of OTDR modules are available on the consoli dated OTDR datasheet.

Lampiran 4. Spesifikasi ITU-T G.652 D

Table 4/G.652 – G.652.D attributes

Fibre attributes		
Attribute	**Detail**	**Value**
Mode field diameter	Wavelength	1310 nm
	Range of nominal values	8.6-9.5 μm
	Tolerance	±0.6 μm
Cladding diameter	Nominal	125.0 μm
	Tolerance	±1 μm
Core concentricity error	Maximum	0.6 μm
Cladding noncircularity	Maximum	1.0%
Cable cut-off wavelength	Maximum	1260 nm
Macrobend loss	Radius	30 mm
	Number of turns	100
	Maximum at 1625 nm	0.1 dB
Proof stress	Minimum	0.69 GPa
Chromatic dispersion coefficient	λ_{0min}	1300 nm
	λ_{0max}	1324 nm
	S_{0max}	0.092 ps/nm^2 × km
Cable attributes		
Attribute	**Detail**	**Value**
Attenuation coefficient	Maximum from 1310 nm to 1625 nm (Note 2)	0.4 dB/km
	Maximum at 1383 nm ±3 nm	(Note 3)
	Maximum at 1550 nm	0.3 dB/km
PMD coefficient	M	20 cables
	Q	0.01%
	Maximum PMD_Q	0.20 ps/$\sqrt{km}$

NOTE 1 – According to 6.2, a maximum PMD_Q value on uncabled fibre is specified in order to support the primary requirement on cable PMD_Q.

NOTE 2 – This wavelength region can be extended to 1260 nm by adding 0.07 dB/km induced Rayleigh scattering loss to the attenuation value at 1310 nm. In this case, the cable cut-off wavelength should not exceed 1250 nm.

NOTE 3 – The sampled attenuation average at this wavelength shall be less than or equal to the maximum value specified for the range, 1310 nm to 1625 nm, after hydrogen ageing according to IEC 60793-2-50 regarding the B1.3 fibre category.

Lampiran 5. Tabel Konversi Daya Redaman

Redaman (dB)	Persentasi Daya yang Diterima di Penerima	Persentasi Daya yang Hilang	Keterangan
1	79%	21%	---
2	63%	37%	---
3	50%	50%	1/2 Daya Pengirim
4	40%	60%	---
5	32%	68%	---
6	25%	75%	1/4 Daya Pengirim
7	20%	80%	1/5 Daya Pengirim
8	16%	84%	1/6 Daya Pengirim
9	12%	88%	1/8 Daya Pengirim
10	10%	90%	1/10 Daya Pengirim
11	8%	92%	1/12 Daya Pengirim
12	6.30%	93.70%	1/16 Daya Pengirim
13	5%	95%	1/20 Daya Pengirim
14	4%	96%	1/25 Daya Pengirim
15	3.20%	96.80%	1/30 Daya Pengirim
16	2.50%	97.50%	1/40 Daya Pengirim
17	2%	98%	1/50 Daya Pengirim
18	1.60%	98.40%	1/60 Daya Pengirim
19	1.30%	98.70%	1/80 Daya Pengirim
20	1%	99%	1/100 Daya Pengirim
25	0.30%	99.70%	1/300 Daya Pengirim
30	0.10%	99.90%	1/1000 Daya Pengirim
40	0.01%	99.99%	1/10,000 Daya Pengirim
50	0.00%	100.00%	1/100,000 Daya Pengirim