Dokumen ini berisi lampiran listing program Matlab tambahan yang diperlukan untuk praktikum modul 1 Transmisi Digital baseband.

Buatlah 5 function tambahan berikut dan simpanlah dalam 1 folder yang sama untuk melakukan step-step percobaan yang telah dijelaskan di modul 1.

Function tambahan yang diperlukan antara lain:

1) **modul.m**

```
function [s]=modul(b,pulsa,rolloff,delay)
%[s] = modul(binary_sequence,'line_code')
global Fd Fs;
Nsamp=Fs/Fd;
if (nargin == 0)
return;
elseif (nargin == 1)
elseif (nargin > 1)
switch lower(pulsa)
    case {'unipolar_nrz'}
        s=rectpulse(b,Nsamp);
    case {'bipolar_nrz'}
        s=rectpulse(b,Nsamp);
        s=s*2-1;
    case {'bipolar_rz'}
        bpolar = 2*b-1;
        if (Nsamp==floor(Nsamp/2)*2)
            bpolarup = upsample(bpolar,2);
            s = rectpulse(bpolarup,Nsamp/2);
        else
            shalf = kron(bpolar,ones(ceil(Nsamp/2),1));
            sfull = [shalf; zeros(floor(Nsamp/2),length(b))];
            s = sfull(:)';
        end
    case {'ami'}
        % ami nrz
        sum_mat = tril(ones(length(b)));
        power_factor = sum_mat * b' + 1;
        bami = (-b).^(power_factor');
        s = rectpulse(bami,Nsamp);
    case {'manchester'}
           b1 = (b == 1);
           b0 = (b == 0);
        if (Nsamp==floor(Nsamp/2)*2)
           if length(b)==1
           b1up = double(upsample(b1,2)');
           b0up_ = upsample(b0,2)';
           else
           b1up = double(upsample(b1,2));
           b0up_ = upsample(b0,2);
           end
           b0up = [0 b0up_(1:length(b0up_)-1)];
           s1 = rectpulse(b1up,Nsamp/2);
           s0 = rectpulse(b0up,Nsamp/2);
           s = 2*(s1 + s0) - 1;
        else
           s1half = kron(b1,ones(ceil(Nsamp/2),1));
           s1full = [s1half; zeros(floor(Nsamp/2),length(b))];
           s1 = s1full(:)';
           s0half = kron(b0,ones(ceil(Nsamp/2),1));
           s0full = [zeros(floor(Nsamp/2),length(b)); s0half];
```

```
            s0 = s0full(:)';
            s = 2*(s1 + s0) - 1;
        end
    case {'unipolar_nyquist'}
        t = -10*Nsamp+1:10*Nsamp-1;
        sincfull = sinc(t);
        bup = upsample(b,Nsamp);
        snoncausal = conv(sincfull,bup);
        s = snoncausal;
        %s = snoncausal(Nsamp*10+1:length(snoncausal));
    case {'bipolar_nyquist'}
        bpolar = 2*b - 1;
        t = -10*Nsamp+1:Nsamp*10-1;
        sincfull = sinc(t);
        bpolarup = upsample(bpolar,Nsamp);
        snoncausal = conv(sincfull,bpolarup);
        s = snoncausal;
        %s = snoncausal(Nsamp*10+1:length(snoncausal));
    case {'duobinary'}
        t = -11*Nsamp+1:Nsamp*11-1;
        sincfull = sinc(t);
        duopulse = sincfull(Nsamp+1:Nsamp*21-1)+sincfull(1:Nsamp-1);
        bup = upsample(b,Nsamp);
        snoncausal = conv(duopulse,bup);
        s = snoncausal;
        %s = snoncausal(Nsamp*10+1:length(snoncausal));
end
elseif (nargin == 3)
    if (lower(pulsa) == 'raised_cosine')
        s = rcosflt(b,Fd,Fs,'sqrt',rolloff,delay);
    end
end
```

2) scope.m

```
function scope(s,os,pj)
global Fd Fs;
Nsamp = Fs/Fd;
switch lower(os)
case {'one-shot'}
    T=1/Fd;
    jmlbit=pj/T;
    ss=s;
    m=length(ss);
    n=0:m-1;nn=n*pj/(Nsamp*jmlbit);
    plot(nn,ss); xi=min(ss); xa=max(ss);
    axis([0 pj xi-abs(xi/10) xa+abs(xa/10)]);
case {'continuous'}
    T=1/Fd;
    MM=pj/T;
    jmlbit=length(s)/(MM*Nsamp);
    ss=s;ss=reshape(ss,(MM*Nsamp),jmlbit);
    n=0:(MM*Nsamp)-1;n=(n/Nsamp)*T;
    plot(n,ss,'b');xi=min(s);
    xa=max(s);
    axis([0 pj xi-abs(xi/10) xa+(xa/10)]);
end
```

3) bbchannel.m

```
function [y]=bbchannel(s,att,npwr,bw)
global Fd Fs;
%ss=s;
%bw=bw;
N=length(s);
M=floor(N*(round(bw)/Fs));
nawgn=(sqrt(npwr))*rand(1,N);
x=att*s + nawgn;
Xf=fft(x);
if N==2*floor(N/2)
   zbw=(N/2)-M-1;
else
   zbw=(N+1)/2-M-1;
end
bww=ones(1,M+1);
bwz=zeros(1,zbw);
sk=[bww bwz bwz bww];
Xflp=Xf.*sk;
y=real(ifft(Xflp));
```

4) isichannel.m

```
function [chnlout] = isichannel(s,gain,npwr,bw,wno)
global Fs Fd;
N = length(s);
M=floor(N*(round(bw)/Fs));
nawgn=(sqrt(npwr))*rand(1,N);
x=gain*s + nawgn;
Xf=fft(x);
if N==2*floor(N/2)
   zbw=(N/2)-M-1;
else
   zbw=(N+1)/2-M-1;
end
bww=ones(1,M+1);
bwz=zeros(1,zbw);
sk=[bww bwz bwz bww];
Xflp=Xf.*sk; % bandlimited signal
wnoM = floor(wno*M);
if M==2*floor(M/2)
  if wnoM==2*floor(wnoM/2)
      Xflp(M/2-wnoM/2+1:M/2+wnoM/2) = zeros(1,wnoM);
      Xflp(N-(M/2+wnoM/2)+2:N-(M/2-wnoM/2+1)+2) = zeros(1,wnoM);
  else
      Xflp(M/2-(wnoM-1)/2+1:M/2+(wnoM-1)/2-1) = zeros(1,wnoM);
      Xflp(N-(M/2+(wnoM-1)/2-1)+2:N-(M/2-(wnoM-1)/2+1)+2) =
zeros(1,wnoM);
  end
else
  if wnoM==2*floor(wnoM/2)
      Xflp((M+1)/2-wnoM/2+1:(M+1)/2+wnoM/2) = zeros(1,wnoM);
      Xflp(N-((M+1)/2+wnoM/2)+2:N-((M+1)/2-wnoM/2+1)+2) = zeros(1,wnoM);
  else
      Xflp((M+1)/2-(wnoM-1)/2+1:(M+1)/2+(wnoM-1)/2-1) = zeros(1,wnoM);
      Xflp(N-((M+1)/2+(wnoM-1)/2-1)+2:N-((M+1)/2-(wnoM-1)/2+1)+2) =
zeros(1,wnoM);
  end
end
chnlout=real(ifft(Xflp));
```

5) computeisi.m

```
function [isi] = computeisi(s,ts,ori)
global Fd Fs;
if (nargin<3)
    return
elseif (nargin==3)
    Nsamp = Fs/Fd;
    orinum = round((ori+1)*Nsamp)+1;
    tsnum = round(ts*Nsamp);
    ysamp = s(orinum+tsnum:Nsamp:length(s));
    isi = sum(ysamp.^2)/length(ysamp);
end
```