LAMPIRAN A
LISTING PROGRAM

```
% Simulasi Convolutional Coding dengan Viterbi Decoding
% Pembangkitan sinyal biner random
Fd = 1;
Fs = 4; % Frekuensi sampling
N = Fs/Fd;
M = 4;
k = log2(M);
numSymb = 100; % Jumlah simbol
numPlot = 30; % Yang diplot
codeRate = 1/2; % Code rate
constlen = 7; % Constraint length
SNRpBitDemo = 3;
SNR = SNRpBitDemo*k;
seed = [654321 123456];
rand('state',seed(1));
randn('state',seed(2));
msg_orig=randsrc(numSymb,1,[0:1]);
figure(1);
stem([0:numPlot-1],msg_orig(1:numPlot),'bx');
axis([0 numPlot -0.2 1.2]);
xlabel('Waktu');
ylabel('Amplituda');
title('Simbol Biner sebelum Convolutionl Encoding');

% Convolutional Encoding terhadap Sinyal Biner
constlen = [7]; % Constraint length
codegen = [171 133]; % Generator polinomial
tblen = 32; % Trace back length
codeRate = 1/2; % Code Rate
trel=poly2trellis(constlen,codegen); % Trellis
[msg_enc_bi]=convenc(msg_orig,trel);
numEncPlot=numPlot./codeRate;
tEnc=[0:numEncPlot-1]*codeRate;
figure(2);
stem(tEnc,msg_enc_bi(1:length(tEnc)),'rx');
axis([min(tEnc) max(tEnc) -0.2 1.2]);
xlabel('Waktu'); ylabel('Amplituda');
title('Simbol Biner setelah Convolutional Encoding');

% Modulasi terhadap simbol hasil Encoding
msg_enc=bi2de(reshape(msg_enc_bi,size(msg_enc_bi,2)*k,size(msg_enc_bi,1)/k
)');
grayencod=bitxor([0:M-1],floor([0:M-1]/2))
msg_gr_enc=grayencod(msg_enc+1);
msg_tx=dmodce(msg_gr_enc,Fd,[Fs,pi/4],'psk',M);
msg_rx=awgn(msg_tx,SNR-10*log10(1/codeRate)-10*log10(N));
numModPlot=numEncPlot*Fs./k;
```

```
tMod=[0:numModPlot-1]./Fs.*k;
figure(3);
plot(tMod,real(msg_tx(1:length(tMod))),'c-
',tMod,imag(msg_tx(1:length(tMod))),'m-');
axis([min(tMod) max(tMod) -1.5 1.5]);
xlabel('Waktu'); ylabel('Amplituda');
title('Simbol Hasil Encoding setelah modulasi Baseband');
legend(Keluaran CPFSK ,0);

% Demodulasi
msg_gr_demod=ddemodce(msg_rx,Fd,[Fs,pi/4],'psk',M);
[dummy graydecod]=sort(grayencod);
graydecod=graydecod-1;
msg_demod=graydecod(msg_gr_demod+1)';
msg_demod_bi=de2bi(msg_demod,k)';
msg_demod_bi=msg_demod_bi(:);
figure(4);
stem(tEnc,msg_enc_bi(1:numEncPlot),'rx');
hold on;
stem(tEnc,msg_demod_bi(1:numEncPlot),'bo');
hold off;
axis([0 numPlot -0.2 1.2]);
xlabel('Waktu'); ylabel('Amplituda');
title('Simbol setelah Demodulasi');

% Deoding dengan Viterbi Decoder
msg_dec=vitdec(msg_demod_bi,trel,tblen,'cont','hard');  % Hard decision
figure(5);
stem([0:numPlot-1],msg_orig(1:numPlot),'rx');
hold on;
stem([0:numPlot-1],msg_dec(1+tblen:numPlot+tblen),'bo');
hold off;
axis([0 numPlot -0.2 1.2]);
xlabel('Waktu'); ylabel('Amplituda');
title('Simbol Hasil Decoding');
```

```
% Program simulasi dengan constraint length (K) = 3
clear;
close all;
clc;
N = 10^4
EbpNo = (2:0.5:7); % Nilai Eb/No
for i=1: length(EbpNo)
   BER(i)=0;
   for j=1:4
      b=sign(randn(1,N)); % Bit data random yang dipancarkan
      orig_bit=b;
      for k=1: length(b)
         if b(k)==1
            b(k)=0;
         else
            b(k)=1;
         end
      end

   % Constraint length (K) = 3, generator polinomial = [7 5];
   trellis=poly2trellis(3,[7 5]);
   code=convenc(b,trellis);
   for l=1: length(code)
      if code(l)==0
         code(l)=1;
      else
         code(l)=-1;
      end
   end

   % data random sebagai AWGN dengan daya noise tertentu
   n=randn(1,2*N)./10^(EbpNo(i)/20)./sqrt(2);
   r=code+n; % Data diterima
   r=sign(r); % hasil recovery oleh decision
   for k=1: length(r)
      if r(k)==1;
         r(k)=0;
      else
         r(k)=1;
      end
   end

   tb=32; % Traceback length
   decoded=vitdec(r,trellis,tb,'trunc','hard');  % Viterbi Decoding
```

```
for k=1: length(decoded)
      if decoded(k)==0;
         decoded(k)=1;
      else
         decoded(k)=-1;
      end
   end

   % Menghitung jumlah bit error
   BER(i)=BER(i)+sum(abs(decoded-orig_bit))/2;
end
BER(i)=BER(i)/5/N;  % BER = Jmlh error rata2/jmlh bit yg dikirim
end
semilogy(EbpNo,BER,'*');
hold on;

% Plot P(e) sebagai fungsi Eb/No secara teoritis
t=(0:0.5:10); % Nilai Eb/No yang ingin dihitung BER-nya
Pe=0.5.*erfc(sqrt(10.^(t/10)));
semilogy(t,Pe,'r');
grid on;
title('BER terhadap Eb/No untuk Kanal AWGN dengan K = 3');
xlabel('EB/No (dB)');
ylabel('BER');
legend('Hasil simulasi BER','Nilai teoritis P(e) tanpa coding',0);
```

Untuk K = 3, 4, dan 5 dengan mengganti nilai K dan generator polinomial pada program sesuai dengan Tabel IV.1.

```
%Program perbandingan nilai BER dengan grafik
clear;
close all;
clc;

Eb_No=[2,2.5,3,3.5,4,4.5,5,5.5,6,6.5,7];
Pe_teoritis=10.^-3.*[37.506 29.655 22.878 17.173 12.501 8.7938...
      5.9539 3.8622 2.3883 1.3998 0.77267];
Pe_tanpa_coding=10.^-3.*[37.566 29.773 22.771 17.168 12.534 8.773...
      5.9574 3.8646 2.3798 1.391 0.7796];
BER_K_3=10.^-3.*[27.421 14.633 8.553 5.6837 3.7853 1.935 0.82...
      0.5501 0.1096 0.0388 0.0108];
BER_K_4=10.^-3.*[21.301 13.26 7.5984 4.2548 2.2452 1.121 0.53...
      0.2074 0.0764 0.0238 0.0084];
BER_K_5=10.^-3.*[20.651 12.511 7.1922 3.958 1.989 1.0234 0.4756...
      0.1842 0.0694 0.02 0.0054];
figure;
plot(Eb_No,Pe_teoritis,'rx-',Eb_No,Pe_tanpa_coding,'bo-',Eb_No,BER_K_3,'k',...
   Eb_No,BER_K_4,'b',Eb_No,BER_K_5,'c');
grid;
legend('teoritis','tanpa coding','K_3','K_4','K_5');
```