FUNGSI M-FILE MATLAB

LAMPIRAN A:

Percobaan untuk kode LDPC

```
function ldpc_dB_test(Go,Ho,max_ite,maxSNR,inc)                % Percobaan dengan bit Flip (BF)

% masukan (input)
%   Go – Matriks Generator
%   Ho – Matriks Parity Check
%   max_ite – nilai maksimum pada pengulangan (iteration)
%   maxSNR – nilai maksimum pada SNR
%   inc - penambahan pada nilai SNR

close all

[Grow,Gcol]=size(Go);
[row,col] = size(Ho);
snr= [0:inc:maxSNR];

max = max_ite;
max_block=10;
% blok maksimum (max_block)=100;

% Untuk perhitungan SNR/BER
block=0;
biterrorsBF=0;                  % untuk perhitungan BER

%%%%%%%%%%  Membuat sebuah file %%%%%%%%%%
serial=fix((datenum(clock))*1e3);
s=sprintf('%d_BF_study_results.txt',serial);
fid = fopen(s,'a');
```

```
fprintf(fid,'\n');
fprintf(fid,'%s %d\n',' Date Stamp= ',serial);
fclose(fid);

for a=1:length(snr)

   block=0;
   biterrorsBF=0;                % untuk perhitungan BER

   while (block < max_block)

      block = block+1;

      fprintf(1,'\\\\\\\t\t Simulasi untuk \t\t SNR=%d \t\t block=%d\n',snr(a),block)

      %%%%%%%% Membangkitkan bit message yang acak %%%%%%%%
      u = (sign(randn(1,size(Go,1)))+1)/2;

      %%%%%%%% Mengkodekan Message %%%%%%%%
      [cw]=ldpc_encode(Go,u);

      % Modulasi BPSK
      % "1" --> 1     "0" --> -1
      cw_bpsk = 2*cw-1;

      %%%%%%%% menambahkan AWGN %%%%%%%%
      % Transmisi AWGN pada soft decision menerima rentetan (sequence)
      y_bpsk = awgn(cw_bpsk,snr(a));  %awgn berasal dari toolbox komunikasi
```

```
%%%%%%%% Meng-dekodekan Message %%%%%%%%
      % Algoritma Bit Flip (BF)
      [synBF,y_reBF] = ldpc_bf(Ho,y_bpsk,max);
%       [synBF,y_reBF] = ldpc_bf_dB(Ho,y_bpsk,max);

      [numBF,ratioBF] = biterr(cw,y_reBF,'secara keseluruhan'); % berasal dari Toolbox Komunikasi
      biterrorsBF = biterrorsBF+numBF;

      if rem(block,50)==0; %simpan statistik setelah setiap 50 blok
         s=sprintf('%d_BF_study_results.txt',serial);
         fid = fopen(s,'a');
         fprintf(fid,'\n');
         fprintf(fid,'%s %2.1E\n',' SNR= ',snr(a));
         fprintf(fid,'%s %d\n',' block= ',block);
         fprintf(fid,'%s %d\n',' BFbiterrors= ',biterrorsBF);
         fclose(fid);
      end

   end

%     fprintf(1,'Simulasi selesai untuk SNR: %d \n',snr(a))

end  % untuk a
```

LAMPIRAN B:

Percobaan dengan algoritma *Bit Flip* (BF) untuk proses dekoding

```
function [u ,ite]=ldpc_bf(H,re,max_ite)        % fungsi aktual

% keluaran (output)
%  u - mengkodekan message
%  ite – nilai pengulangan (iteration)

% input
%   H – matriks parity-check
%   re – menerima kata
%   max_ite – pengulangan maksimum (maximum iteration)

% tic            % start timer

%%%%%%%%%%      Langkah 1: Membangkitkan bit sindrom
%%%%%%%%%%

% inisialisasi matriks dan variabel
[row,col] = size(H);
hard=[ ];
y_re = re;
iteration = 0;

% hard decision dari BPSK
% dimodifikasi dari bitf.c
% ECC Website, http://the-art-of-ecc.com
% y_re > 0 --> 1
% y_re <= 0 --> 0
for i = 1:col
   if y_re(i) > 0.0
```

```
hard(i) = 1;
   else
      hard(i) = 0;
   end % if
end % for

y_re = hard;
syn = mod(y_re*H',2);          % bit-bit sindrom

while (sum(sum(syn)) ~= 0) & (iteration < max_ite)  %pengecekan jika syn=0 atau pengulangan maksimum (maximum iteration) tercapai

   iteration = iteration + 1;
   %%%%%%%%%% Langkah 2: Komputasi untuk S, pengecekan node bit %%%%%%%%%%
   S=zeros(1,1);
   for i = 1:col
      S(i) = syn*H(:,i);
   end % for i

   %%%%%%%%%% Langkah 3: Kembalikan bit-bit menjadi flipped %%%%%%%%%%
   [srow,scol]=size(S);
   bflip=[ 1 ];
   flip_count=1;

   for i = 1:scol-1
      if S(i+1)>=S(bflip)
         bflip(flip_count)=i+1;
```

```
flip_count=flip_count+1;
        end % if S
    end % for i

    if S(1) == S(bflip(1))
        bflip(flip_count)=1;
    end % if S

    %%%%%%%%%% Langkah 4: Flip bit-bit %%%%%%%%%%
    y_re(bflip)=not(y_re(bflip));
    syn = mod(y_re*H',2);  % peroleh/sebelum komputasi bit-bit sindrom

end % while

% kembalikan nilai hasil
if (sum(sum(syn)) == 0)
    disp('BF DECODING BERHASIL')
    u = y_re;
    ite = iteration;
end % if sum

if (sum(sum(syn)) ~= 0)
    u = [0];
    ite = iteration;
    disp('BF DECODING TIDAK BERHASIL')
end % if sum

% toc           % end timer
```

LAMPIRAN C:

Percobaan pada data pengamata dan analisa untuk kode LDPC

```
clc;
clear all;
close all;
clc;

load ga
Go = Ga;
load ha
Ho = Ha;
Hs=full(Ha);
ldpc_dB_test(Go,Ho,50,15,1);
```