

Basic Wireless Communication Systems

Citra Yasin Akbar Fadhlika, S.T.

Center for Advanced Wireless Technologies (AdWiTech),
Telkom University,
Jl. Telekomunikasi No.1, Terusan Buah Batu, Bandung, Indonesia
Email: citrayaf@gmail.com

Presented at WRITE-5G Tutorial
Bandung, 5 April 2020

Outline

- 1 Struktur Dasar Sistem Komunikasi Nirkabel
- 2 *Channel Coding*
 - *Repetition Codes*
 - *LDPC Codes*
- 3 Modulasi
 - BPSK
- 4 *Noise*
- 5 *Bit Error Rate (BER)*
- 6 Latihan Soal
- 7 MATLAB Program

1 Struktur Dasar Sistem Komunikasi Nirkabel

- 2 *Channel Coding*
 - *Repetition Codes*
 - *LDPC Codes*

- 3 *Modulasi*
 - *BPSK*

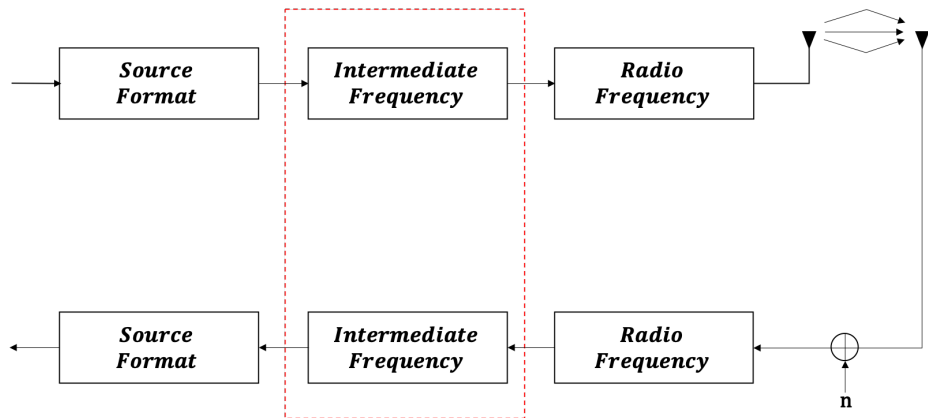
- 4 *Noise*

- 5 *Bit Error Rate (BER)*

- 6 *Latihan Soal*

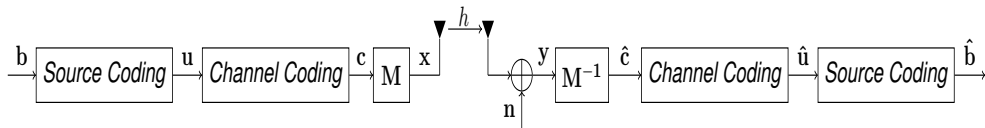
- 7 *MATLAB Program*

Struktur *Transmitter* dan *Receiver*



- Blok sistem komunikasi secara umum terdiri atas tiga bagian yaitu *Source Format*, *Intermediate Frequency*, dan *Radio Frequency*.

Struktur Dasar Sistem Komunikasi



- Pada Blok *Intermediate Frequency* (IF), terdapat 3 komponen utama yaitu: Blok *Source Coding*, Blok *Channel Coding*, dan Blok Modulasi.

1 Struktur Dasar Sistem Komunikasi Nirkabel

2 *Channel Coding*

- *Repetition Codes*
- *LDPC Codes*

3 Modulasi

- BPSK

4 *Noise*

5 *Bit Error Rate (BER)*

6 Latihan Soal

7 MATLAB Program

- *Channel coding*, dikenal juga sebagai *forward error control coding* (FECC), adalah suatu proses pendeteksian dan perbaikan bit yang mengalami kesalahan pada saat transmisi.
- Pada sisi pengirim sistem komunikasi digital bagian *channel coding* yang disebut juga sebagai *encoder*, redundansi (*parity bit*) ditambahkan pada informasi yang dikirimkan sebelum proses modulasi untuk dijadikan sebagai pelindung.
- Pada sisi penerima, bagian *channel coding* yang disebut *decoder* melakukan pendeteksian informasi dan melakukan perbaikan, jika informasi mengalami kerusakan saat terkena *noise*, interferensi, dan *fading*.

1 Struktur Dasar Sistem Komunikasi Nirkabel

2 *Channel Coding*

- *Repetition Codes*

- *LDPC Codes*

3 Modulasi

- BPSK

4 *Noise*

5 *Bit Error Rate (BER)*

6 Latihan Soal

7 MATLAB Program

Repetition Codes



- *Repetition codes* adalah salah satu *channel coding* dasar yang melindungi bit informasi dengan menggandakannya beberapa kali. Contoh untuk *Repetition codes* dengan *code rate* $R = \frac{1}{3}$:

$$b = 1 \text{ maka } c = 111. \quad (1)$$

- *Repetition codes* men-decode *codeword* yang diterima sesuai dengan R , ketika $R = \frac{1}{3}$ maka setiap 3 bit akan di-decode. *Repetition codes* memiliki dua metode *decoding*:
 - ▶ *Hard Decoding*
Repetition codes akan menerjemahkan *codeword* sesuai dengan bit yang paling banyak muncul.
 - ▶ *Soft Decoding*
Repetition codes mengolah LLR yang diterima dengan

$$z = \sum_{i=1}^{\frac{1}{R}} x_i, \quad (2)$$

kemudian z diterjemahkan menjadi bit.

1 Struktur Dasar Sistem Komunikasi Nirkabel

2 *Channel Coding*

- *Repetition Codes*

- *LDPC Codes*

3 Modulasi

- BPSK

4 *Noise*

5 *Bit Error Rate (BER)*

6 Latihan Soal

7 MATLAB Program

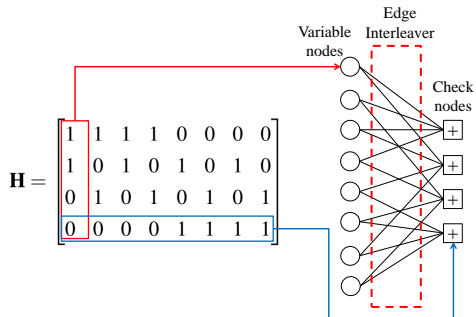
Low Density Parity Check (LDPC) Codes

- LDPC codes adalah *binary linear block codes* dengan matriks *sparse parity-check* \mathbf{H} yang sebagian besar elemennya bernilai "0" dan hanya beberapa yang bernilai "1".
- Matriks \mathbf{H} terdiri dari:
 - Panjang blok n
 - Degree of variable node d_v
 - Degree of check node d_c
- Matriks \mathbf{H} dapat direpresentasikan dengan Tanner *graph* dan *degree distribution*.

$$\Lambda(x) = x^2 \quad \Omega(x) = x^4.$$

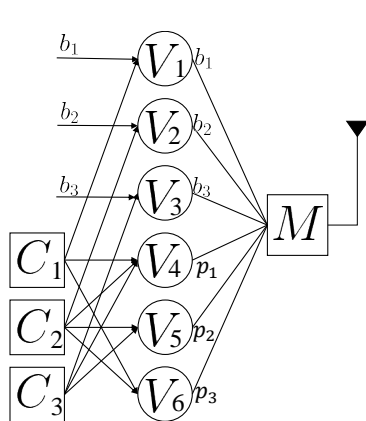
$$\mathbf{G} = [I_k \ P^T], \quad (4)$$

$$\mathbf{G} \cdot \mathbf{H}^T = 0. \quad (5)$$

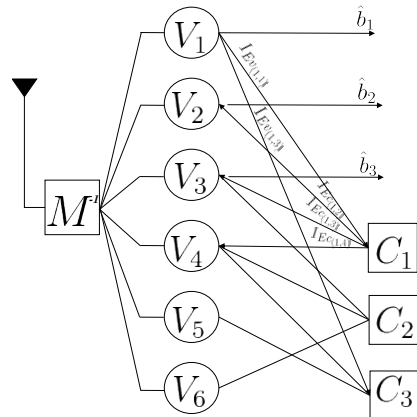


(3)

Encoding dan Decoding pada LDPC Codes



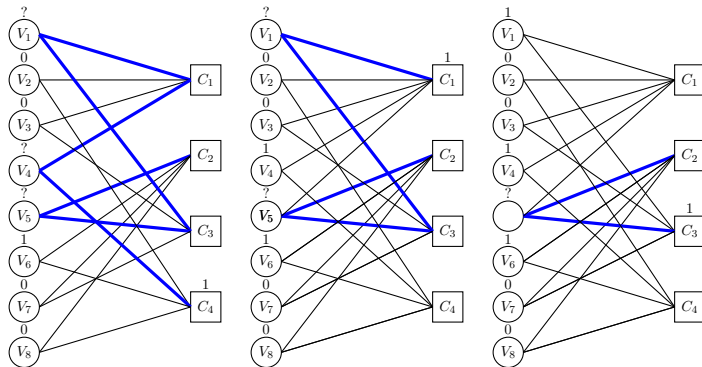
Matriks *Generator* G adalah rangkaian *check node* C , *interleaver* Π_x , dan *variable node* V pada *transmitter*. Maka *codeword* c dapat dituliskan dengan $c = bG$.



Matriks *parity check* H adalah rangkaian *decoder* pada *Receiver*. LDPC decoder melakukan *iterative V – C – V Min Sum Algorithm (MSA)*.

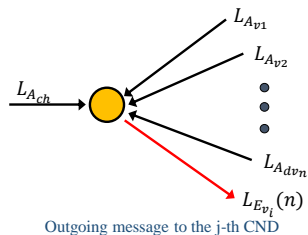
Hard Decoding LDPC Codes

Proses *hard decoding* menggunakan bentuk biner sebagai *input* dengan menerapkan operasi XOR pada CND yang menghasilkan nilai *zero*. Berikut contoh *hard decoding* pada *Binary Erasure Channel* (BEC) dengan sinyal yang diterima adalah $[?00??100]$.



Gambar: Proses *hard decoding* LDPC pada kanal BEC

Soft Decoding LDPC Codes

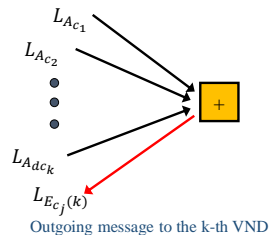


Gambar: Informasi Extrinsic V ke C.

$$L_{E_{v_i}}(n) = L_{A_{ch}} + \sum_{j=1, j \neq i}^{d_{v_n}} L_{A_{v_j}}. \quad (6)$$

Approximasi Box Plus L_1 dan L_2

$$L_1 \boxplus L_2 \approx \text{sign}(L_1 \cdot L_2) \cdot \min\{|L_1|, |L_2|\}. \quad (8)$$



Gambar: Informasi Extrinsic C ke V.

$$L_{E_{c_j}}(k) = \sum_{i=1, i \neq j}^{d_{c_k}} \boxplus L_{A_{c_i}}. \quad (7)$$

1 Struktur Dasar Sistem Komunikasi Nirkabel

2 *Channel Coding*

- *Repetition Codes*
- *LDPC Codes*

3 **Modulasi**

- BPSK

4 *Noise*

5 *Bit Error Rate (BER)*

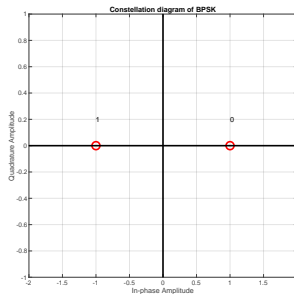
6 Latihan Soal

7 MATLAB Program

- Modulasi adalah suatu teknik untuk membentuk gelombang sinyal, sehingga mencapai jarak yang jauh dan tahan terhadap *error*.
- Dalam modulasi terdapat beberapa jenis modulasi yang digunakan dalam sistem komunikasi digital yaitu *amplitude shift keying* (ASK), *frequency shift keying* (FSK), *phase shift keying* (FSK), *phase shift keying* (PSK), dan *quadrature amplitude modulation* (QAM).

- 1 Struktur Dasar Sistem Komunikasi Nirkabel
- 2 *Channel Coding*
 - *Repetition Codes*
 - *LDPC Codes*
- 3 **Modulasi**
 - **BPSK**
- 4 *Noise*
- 5 *Bit Error Rate (BER)*
- 6 *Latihan Soal*
- 7 *MATLAB Program*

Binary Phase Shift Keying (BPSK)



- Binary Phase Shift Keying (BPSK) adalah salah satu modulasi digital sederhana yang setiap simbolnya hanya terdiri dari satu bit.
- Pada modulasi BPSK, bit c dipetakan ke simbol dengan persamaan

$$x = 1 - 2 \cdot c, \quad (9)$$

dengan x adalah hasil *mapping*.

1 Struktur Dasar Sistem Komunikasi Nirkabel

2 *Channel Coding*

- *Repetition Codes*
- *LDPC Codes*

3 Modulasi

- BPSK

4 **Noise**

5 *Bit Error Rate (BER)*

6 Latihan Soal

7 MATLAB Program

- *Noise* adalah sinyal acak yang tidak diinginkan karena dapat mendistorsi sinyal pada suatu sistem komunikasi.
- *Noise* biasanya disebabkan oleh panas, rangkaian tidak sempurna, dan beberapa hal lain.
- *Noise* akan mempengaruhi sinyal yang diterima pada *receiver*, dengan persamaan matematika sebagai

$$\mathbf{y} = h \cdot \mathbf{x} + \mathbf{n}, \quad (10)$$

dengan \mathbf{y} adalah sinyal yang diterima oleh *receiver*, h adalah kanal, dan \mathbf{n} adalah *noise*.

Additive White Gaussian Noise (AWGN)

- Karakteristik AWGN:
 - ▶ **Additive**, karena AWGN adalah *noise* yang ditambahkan pada semua frekuensi suatu sistem komunikasi.
 - ▶ **White**, AWGN adalah *noise* yang bersifat putih yaitu mengganggu semua frekuensi.
 - ▶ **Gaussian**, karena AWGN terdistribusi secara Gaussian atau normal.
- AWGN sering digunakan sebagai nama *channel* model saat koefisien kanal bernilai 1 atau saat kanal pada kondisi sempurna atau tidak berubah-ubah.
- AWGN dalam perintah MATLAB dapat dimodelkan oleh

$$\mathbf{n} = \sigma \cdot (\text{randn}(1, N) + \text{sqrt}(-1) \times (\text{rand}(1, N))), \quad (11)$$

dengan σ adalah standar deviasi dari *double-sided white noise* dan N adalah jumlah simbol.

1 Struktur Dasar Sistem Komunikasi Nirkabel

2 *Channel Coding*

- *Repetition Codes*
- *LDPC Codes*

3 Modulasi

- BPSK

4 *Noise*

5 *Bit Error Rate (BER)*

6 Latihan Soal

7 MATLAB Program

Bit Error Rate (BER)

- Pada sistem komunikasi digital, informasi direpresentasikan oleh biner, sehingga kualitas sistem ini diukur dengan rata-rata nilai *bit error rate* (BER). Bit dikatakan mengalami *error* saat bit yang dikirim berbeda dengan bit yang diterima. BER dapat dinyatakan dengan

$$BER = \frac{\text{jumlah bit error}}{\text{jumlah bit yang dikirimkan}}. \quad (12)$$

- Teori BER untuk BPSK pada kanal AWGN dinyatakan oleh

$$BER_{BPSK_AWGN} = \frac{1}{2} \text{erfc}(\sqrt{\gamma}), \quad (13)$$

dengan γ adalah SNR.

- 1 Struktur Dasar Sistem Komunikasi Nirkabel
- 2 *Channel Coding*
 - *Repetition Codes*
 - *LDPC Codes*
- 3 *Modulasi*
 - BPSK
- 4 *Noise*
- 5 *Bit Error Rate (BER)*
- 6 Latihan Soal**
- 7 MATLAB Program

Latihan Soal

Diketahui:

- a) Menggunakan modulasi BPSK b) Bit informasi $b = (0, 1, 1)$ c) Melalui kanal sempurna

Ditanya:

- 1 Menggunakan *Repetition codes* dengan $R = \frac{1}{3}$

a. *Codeword*?

b. Estimasi bit informasi \hat{b} ? Untuk

$$L(y) = (-0.911, 3.153, 1.646, -1.594, 1.571, -1.394, -1.571, -1.271, -1.726)$$

- 2 Menggunakan LDPC *codes* dengan matriks

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

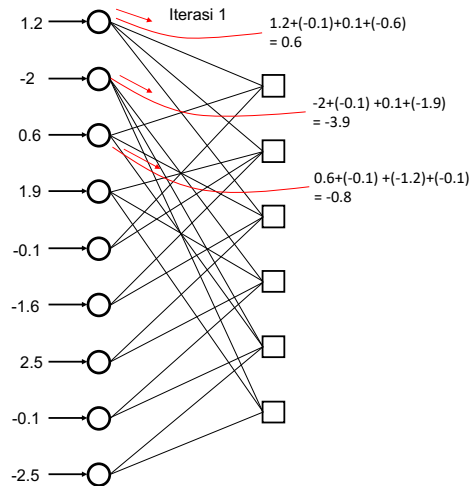
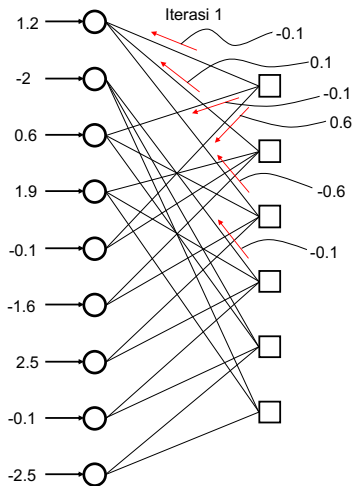
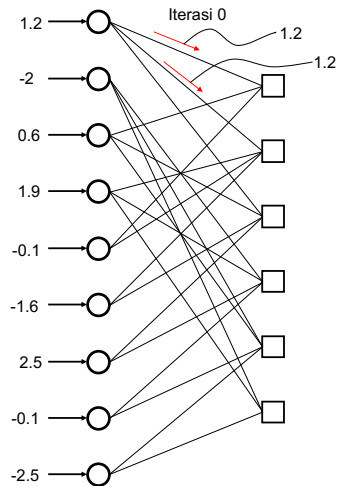
a. Matriks \mathbf{G} ?

b. *Codeword*?

c. Estimasi bit informasi \hat{b} pada iterasi pertama? Untuk

$$L(y) = (1.2, -2, 0.6, 1.9, -0.1, -1.6, 2.5, -0.1, -2.5).$$

Soft Decoding LDPC Codes MSA



- 1 Struktur Dasar Sistem Komunikasi Nirkabel
- 2 *Channel Coding*
 - *Repetition Codes*
 - *LDPC Codes*
- 3 Modulasi
 - BPSK
- 4 *Noise*
- 5 *Bit Error Rate (BER)*
- 6 Latihan Soal
- 7 **MATLAB Program**

MATLAB Program: Teori BER BPSK

```
1. SNR=0:2:30;
%inisialisasi SNR dalam dB
2. numerik_snr=10.^(SNR/10);
%mengubah SNR dari dB ke numerik
3. for i=1:length(SNR)
%melakukan perulangan sebanyak jumlah SNR yang akan diuji
4.     fprintf('SNR: %d \n', SNR(i));
%mencetak di command window SNR yang sedang diproses
5.     BER_bpsk_awgn(i)=0.5*erfc(sqrt(numerik_snr(i)));
%rumus teori BER BPSK
6. end
7. semilogy(SNR,BER_bpsk_awgn,'-r*','linewidth',1)
%mengeplot hasil perhitungan dengan sumbu y SNR dan sumbu x hasil dari perhitungan teori BER
```

MATLAB Program: Uncoded BSPK (1/2)

```
1. info = 120;
%inisialisasi jumlah informasi
2. Modulasi = 2;
%inisialisasi modulasi
3. indeks_modulasi = log2(Modulasi);
%untuk mengetahui indeks modulasi
4. frame = 25;
%inisialisasi jumlah frame
5. SNR = 0;
%inisialisasi SNR dalam dB
6. snr = 10.^(SNR/10);
%merubah SNR dB ke numerik
7. sigma = sqrt(1./(2*snr));
%rumus probabilitas densitasnya
8. jum_error = 0;
%inisialisasi variabel untuk menyimpan jumlah error
9. for i = 1:frame
%perulangan sebanyak frame
10. bit_informasi = randi([0 1], 1, info);
%generate bit informasi sepanjang variabel info
11. x = 1-bit_informasi*2;
%mapper BPSK
12. h = 1;
%inisialisasi kanal sempurna
13. noise = sigma*(randn(1,length(x)) + sqrt(-1)*randn(1,length(x)));
%generate noise dengan distribusi Gaussian
14. y = h*x+noise;
%persamaan sinyal terima
15. u = real(y);
%karena BPSK hanya membagi dua wilayah
```

MATLAB Program: Uncoded BSPK (2/2)

```
16. for j = 1:length(u)
17.     if u(j) > 0
18.         b(j) = 0;
19.     else
20.         b(j) = 1;
21.     end
22. end
%demapper BPSK, proses merubah kembali ke bit
23. error = sum(b ~= bit_informasi);
%menghitung jumlah perbedaan bit informasi awal dengan bit yg diterima
24. jum_error = jum_error+error;
%menjumlahkan error per frame
25. end
26. ber = jum_error / (info*frame);
% rumus untuk menghitung BER
```

MATLAB Program: Repetition Codes

```
1. Modulasi = 2;
2. indeks_modulasi = log2(Modulasi);
3. frame = 1;
4. SNR = -5;
5. rate = 1/3;
%rate dari repetition codes
6. snr = 10.^(SNR/10);
7. sigma = sqrt(1./(2*snr));
8. jum_error = 0;
9. for i = 1:frame
10.  bit_informasi = [0 1 1];
%bit informasi telah di-state karena mengikuti latihan soal
11.  c=[];
%inisialisasi variabel untuk menyimpan codeword dari repetition codes
12.  for r = 1:length(bit_informasi)
13.    for re = 1:1/(rate)
14.      c = [c bit_informasi(r)];
%proses penggandaan
15.    end
16.  end
%proses encoding repetition codes
17.  x = 1-c*2;
%mapping codeword
18.  h = 1;
19.  noise = sigma*(randn(1,length(x)) + sqrt(-1)*randn(1,length(x)));
20.  y = h*x+noise;
21.  u = real(y);
```

MATLAB Program: Repetition Codes Hard Decoding

```
%Soal
22. u = [-0.911 3.153 1.646 -1.594 1.571 -1.394 -1.571 -1.271 -1.726];
% LLR channel yang diterima
32. for j = 1:length(u)
33.     if u(j) > 0
34.         b(j) = 0;
35.     else
36.         b(j) = 1;
37.     end
38. end
39. eb = [];
%inisialisasi variabel untuk hasil proses decoding
40. m = 1;
%counter untuk proses hard decoding
41. for d = 1:1/(rate):length(b)
%perulangan dengan step sebesar 1/rate sampai panjang dari b
42.     eb = [eb mode(b(d:1/(rate)*m))];
%proses mencari bit terbanyak yg muncul per 1/rate
43.     m = m+1;
44. end
%HARD Decoding
45. b = eb;
46. error = sum(b ~= bit_informasi);
47. jum_error = jum_error+error;
48. end
49. ber = jum_error/(length(bit_informasi)*frame);
```


MATLAB Program: Repetition Codes Soft Decoding

```
%Soal
22.  u = [-0.911 3.153 1.646 -1.594 1.571 -1.394 -1.571 -1.271 -1.726];
23.  ub = [];
%inisialisasi variabel untuk menempatkan hasil decoding
24.  for xx = 1:1/(rate):length(u)
%perulangan dengan step sebesar 1/rate sampai panjang dari u
25.  x = 0;
%inisialisasi variabel untuk perhitungan soft decoding
26.  for j = 1:1/(rate)
%perulangan untuk membaca setiap 1/rate Lch
27.  x = x + u(xx+j-1);
%perhitungan soft decoding repetition codes 28.  end
29.  ub = [ub x];
%memasukkan nilai hasil decoding ke ub 30.  end
31.  u = ub;
32.  for j = 1:length(u)
33.  if u(j) > 0
34.  b(j) = 0;
35.  else
36.  b(j) = 1;
37.  end
38.  end
%Soft Decoding
39.  error = sum(b ~= bit_informasi);
40.  jum_error=jum_error+error;
41.  end
42.  ber = jum_error/(length(bit_informasi)*frame);
```

MATLAB Program: LDPC Codes (1/2)

```
1. Modulasi = 2;
2. indeks_modulasi = log2(Modulasi);
3. frame = 1;
4. SNR = -6;
% LDPC
5. iterasi = 1;
%inisialisasi iterasi, menentukan banyaknya proses iterasi pada soft decoding LDPC codes
6. H = [1,0,1,0,1,0,0,0,0;1,0,0,1,1,1,0,0,0;1,0,1,0,0,1,1,0,0;0,1,0,1,0,0,1,1,0;0,1,1,0,0,0,0,1,1;0,1,0,1,0,0,0,0,1];
%matriks parity check LDPC codes
7. rows = size(H, 1);
%menghitung jumlah kolom matriks H
8. cols = size(H, 2);
%menghitung jumlah baris matriks H
9. k = cols-rows;
%menghitung jumlah informasi yang dapat disisipkan
10. G_awal = H;
%Gauss Jordan Elimination, mulai dari sini
11. r = 1;
12. for c = cols - rows + 1:cols
13.     if G_awal(r,c) == 0
14.         for r2 = r + 1:rows
15.             if G_awal(r2,c) ~= 0
16.                 tmp = G_awal(r, :);
17.                 G_awal(r, :) = G_awal(r2, :);
18.                 G_awal(r2, :) = tmp;
19.             end
20.         end
21.         if G_awal(r,c) == 0
22.             error('H is singular');
%apabila muncul error ini, maka matriks H adalah singular sehingga matriks G tidak dapat terbentuk dari matriks H
23.     end
```

MATLAB Program: LDPC Codes (2/2)

```
24. end
25. for r2 = r + 1:rows
26.     if G_awal(r2, c) == 1
27.         G_awal(r2, :) = xor(G_awal(r2, :), G_awal(r, :));
28.     end
29. end
30. for r2 = 1:r - 1
32.     if G_awal(r2, c) == 1
33.         G_awal(r2, :) = xor(G_awal(r2, :), G_awal(r, :));
35.     end
36. end
37. r = r + 1;
38. end
%sampai sini, Gauss Jordan Elimination
39. G = [eye(3) G_awal(:,1:3)'];
%rumus pembentukan matriks G LDPC codes
%sebelum dilanjutkan ke tahap berikutnya, baiknya melakukan pengecekan dengan melakukan operasi mod(G*H',2) di command window
6. snr = 10.^(SNR/10);
7. sigma = sqrt(1./(2*snr));
8. jum_error = 0;
9. for i = 1:frame
10.     bit_informasi = [0 1 1];
%bit informasi di-state sesuai dengan soal
%encode LDPC
11. c = mod(bit_informasi*G,2);
%c adalah codeword dari LDPC codes
17. x = 1-c*2;
18. h = 1;
19. noise = sigma*(randn(1,length(x)) + sqrt(-1)*randn(1,length(x)));
20. y = h*x+noise;
21. u = real(y);
```

MATLAB Program: LDPC Codes Soft Decoding (1/2)

```
%Soal
22. u = [1.2 -2 0.6 1.9 -0.1 -1.6 2.5 -0.1 -2.5];
%Soft Decoding
44. Le_VND = zeros(rows,cols);
45. La_VND = zeros(rows,cols);
46. Le_CND = zeros(rows,cols);
47. La_CND = zeros(rows,cols);
%inisialisasi LLR masuk dan keluar dari check node dan variable node
48. for k = 1:cols
49.     el1 = find(H(:,k));
50.     La_VND(el1,k) = u(k);
51. end
52. Le_VND = La_VND;
%masuk LLR chanel
%iterasi ke-0
%mulai iterative decoding LDPC codes
53. for iter = 1 : iterasi
%perhitungan di check node
54.     for m = 1:rows
55.         el2 = find(H(m,:));
56.         for ec = 1:length(el2)
57.             el2_baru = setdiff(el2,el2(ec));
58.             for ee = 1:length(el2_baru)
59.                 if ee == 1
60.                     operasi_CND = Le_VND(m,el2_baru(ee));
61.                 else
62.                     operasi_CND = sign(operasi_CND)*sign(Le_VND(m,el2_baru(ee)))*min(abs([operasi_CND Le_VND(m,el2_baru(ee))]]));
%rumus di check node
63.                 end
64.             end
65.             La_CND(m,el2(ec)) = operasi_CND;
```

MATLAB Program: LDPC Codes Soft Decoding (2/2)

```
68. Le_CND = La_CND;
%variable node
69. for k = 1:cols
70.     e11 = find(H(:,k));
71.     for ev = 1:length(e11)
72.         La_VND(e11(ev),k) = u(k)+sum(Le_CND(e11,k))-Le_CND(e11(ev),k);
%proses perhitungan di variable node
73.     end
74. end
75. Le_VND = La_VND;
76. end
77. for in = 1:length(bit_informasi)
78.     ub(in) = sum(Le_CND(:,in))+u(in);
% perhitungan LLR akhir dengan mengambil sepanjang bit informasi
79. end
80. u = ub;
81. for j = 1:length(u)
82.     if u(j) > 0
83.         b(j) = 0;
84.     else
85.         b(j) = 1;
86.     end
87. end
88. error = sum(b ~= bit_informasi);
89. jum_error=jum_error+error;
90. end
91. ber = jum_error/(length(bit_informasi)*frame);
```

Referensi

1. Harada, Hiroshi and Prasad, R. Simulation and Software Radio for Mobile Communications. 2002.
2. Martin Bossert. 1999. Channel Coding for Telecommunications (1st. ed.). John Wiley & Sons, Inc., USA.
3. S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," IEEE Transactions on Communications, vol. 49, no. 10, pp. 1727–1737, Oct 2001..
4. Moon, T.K.. (2005). Error Correction Coding: Mathematical Methods and Algorithms. 10.1002/0471739219.ch3.
5. A. F. Molisch, Wireless Communications, 1st ed. IEEE, 2011.
6. R. Tanner, A recursive approach to low complexity codes, in IEEE Transactions on Information Theory, vol. 27, no. 5, pp. 533-547, September 1981.
7. G. H. Golub and C. F. van Loan, Matrix Computations, 4th ed.
8. Gauss-jordan elimination over $gf(2)$,
<https://stackoverflow.com/questions/44471439/gauss-jordan-elimination-over-gf2>,
accessed: 2020- 03-07.