

PERCOBAAN 3

ENCODER DAN DECODER KODE SIKLIK

1. Tujuan :

Setelah melakukan praktikum, diharapkan mahasiswa dapat :

- Membangkitkan generator siklik dan bit informasi yang telah ditentukan menggunakan matlab.
 - Menghasilkan *codeword* dari data informasi yang telah ditentukan menggunakan generator polinomial tertentu (pengkodean data).
 - Melakukan proses decoding dari codeword yang diperoleh dari proses sebelumnya.
 - Melakukan pengkodean tanpa dan dengan *error correction*.

2. Dasar Teori :

Pengertian Kode Siklik

Menurut definisinya, siklik berarti putaran. Jadi dikatakan kode siklik karena kode-kode vektornya merupakan versi putaran kode vektor lainnya, misal : codeword $[x_n, x_0, x_1, \dots, x_{n-1}]$, maka jika diputar $[x_0, x_1, \dots, x_{n-1}, x_n]$ juga merupakan codeword dari kode siklik. Telah dijelaskan pada bab yang lain bahwa kode blok disusun dengan menggunakan matrik vektor, sedangkan kode siklik di gambarkan sebagai deretan polinomial beserta koefisiennya. Koefisien polinomial inilah yang sangat membantu dalam pembuatan struktur kode siklik. Kode siklik dibagi menjadi 2 macam, yaitu kode siklik sistematik dan non sistematik. Dan mempunyai struktur (n,k) dengan derajat polinomial tertinggi $(n-k)$.

Kode Siklik Sistematik

Dengan generator polinomial $g(x)$ dari sebuah kode siklik, codeword dapat dibentuk secara matematis dengan rumusan :

$$c = (\gamma_0, \gamma_1, \dots, \gamma_{n-k-1}, d_0, d_1, \dots, d_{k-1}) \dots \quad (1)$$

Dari persamaan diatas, informasi polinomial $d(x)$ dan check polinomial $\gamma(x)$ dapat dinyatakan sebagai berikut :

$$d(x) = d_0 + d_1x + \dots + d_{k-1}x^{k-1} \dots \quad (2)$$

$$\gamma(x) = \gamma_0 + \gamma_1 x + \dots + \gamma_{n-k-1} x^{n-k-1} \dots \quad (3)$$

dimana, $\gamma(x)$ adalah sisa pembagian polinomial berderajat $n-k-1$ atau kurang, yang diperoleh dari pembagian $x^{n-k}d(x)$ dengan $g(x)$.

$$x^{n-k}d(x) : g(x) = q(x) + r(x) \dots \quad (4)$$

dengan, $q(x)$ adalah hasil bagi

$r(x)$ adalah sisa pembagian

Kode siklik non sistematis

Pembentukan kode siklik non sistematik lebih mudah dari pada pembentukan kode siklik sistematik. Dari generator polinomial yang telah didapatkan, dikalikan dengan data informasi untuk mendapatkan codeword non sistematik.

Perbedaan yang mendasar antara kode siklik sistematis dan non sistematis adalah pada codeword yang dihasilkan oleh encodernya. Pada kode sistematis dapat dibedakan antara bit-bit informasi dengan bit-bit paritynya, sedangkan pada kode non sistematis tidak dapat dibedakan. Karena sifatnya tersebut, maka kode siklik yang banyak digunakan adalah kode siklik yang sistematis.

Proses Encoding Kode Siklik.

Secara blok diagram encoder dengan generator polinomial $g(x)=1+x+x^3$ dapat digambarkan pada gambar 1.

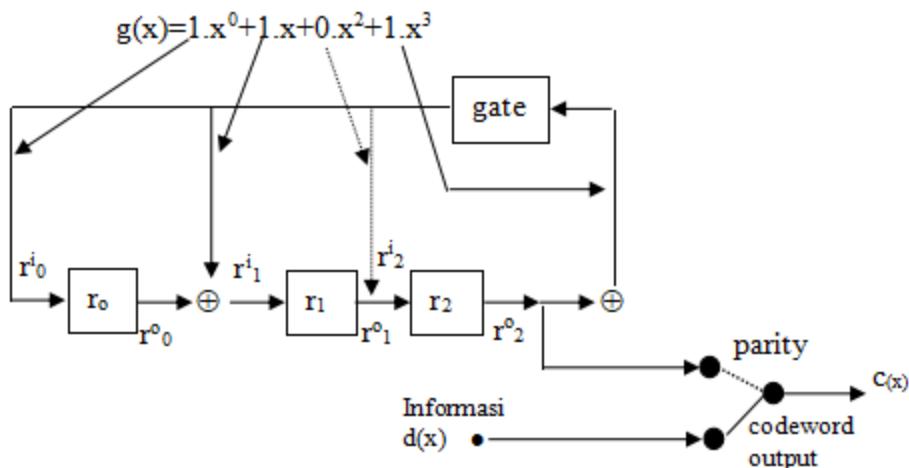
Sebuah kode siklik (7,4) mempunyai :

Generator polinomial $g(x) = 1+x+x^3$

Urutan informasi yang akan dikirim d = (1100)

Informasi polinomial $d(x) = 1+x$

Karena $n-k = 3$, maka didapatkan $x^3 \cdot d(x) = x^3 + x^4$



Gambar 1. Encoder kode siklik dengan $g(x)=1+x+x^3$

Pembagian x^3+x^4 dengan $g(x) = 1+x+x^3$, menghasilkan sisa polinomial :

$$\gamma(x) = 1+x^2.$$

Jadi code polinomial yang didapat adalah :

$$c(x) = 1+x^2+x^3+x^4$$

sehingga codeword (data yang telah dikodekan) $c = (1011100)$

atau dapat dinyatakan bahwa :

$$c(x) = d(x) \cdot g(x) = (1+x)(1+x+x^3) = 1+x^2+x^3+x^4$$

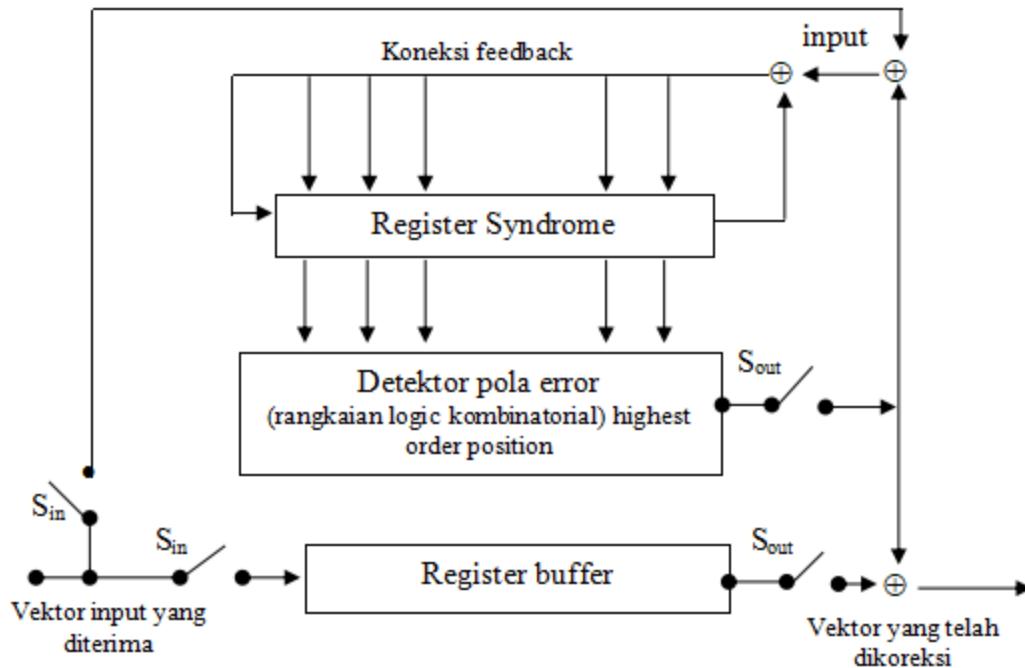
Sehingga codeword (data yang telah dicodekan) $c = (1011100)$

Proses Decoding Kode Siklik

Secara umum decoder kode siklik dapat ditunjukkan dengan blok diagram seperti gambar 2. Secara prinsip prosedur koreksi kesalahan pada kode siklik (sesuai blok diagram diatas) adalah sebagai berikut :

- Step 1. Vektor yang diterima digeser ke register buffer dan register syndrome.
- Step 2. Setelah perhitungan syndrome dan semua bit digeser ke buffer, isi register syndrome dibaca oleh detektor. Bila output detektor =1 maka isi register

syndrome berkaitan dengan pola error yang dapat dikoreksi dengan error terletak di buffer paling kanan.



Gambar 2. Blok Diagram Dekoder Kode Siklik

- c. Step 3. Digit yang diterima pertama digeser keluar, secara bersamaan isi register syndrome juga digeser keluar, bila bit pertama error, output detektor akan 1 yang akan digunakan untuk koreksi error pada bit pertama tersebut. Output detektor juga digeser ke register syndrome untuk memodifikasi syndrome. Isi syndrome berkaitan dengan isi buffer berikutnya.
- d. Step 4. Syndrome yang baru digunakan untuk mengecek apakah ada kesalahan pada bit kedua seperti halnya pada bit pertama, bila ada dilakukan koreksi.
- e. Step 5. Proses diatas berlangsung digit demi digit sampai semua vektor digeser keluar buffer.

3. Perangkat Yang Digunakan :

- a. PC yang berisi program aplikasi matlab
- b. Disket, flash disk, memory card

4. Langkah-langkah Percobaan :

A. POLINOMIAL, MATRIKS GENERATOR, DAN MATRIKS PARITY KODE SIKLIK

A.1. Membangkitkan polinomial kode siklik (7,4) dengan MATLAB

```
>> p=cyclpoly(7,4)  
p =  
    1   0   1   1
```

Menunjukkan bahwa polinomial yang diperoleh adalah $p(x) = 1x^0 + 0x^1 + 1x^2 + 1x^3$

1. Dari langkah A1, untuk Kode siklik (n,k) dibutuhkan generator dengan derajat polinomial tertinggi $(n-k)$. Untuk kode siklik (7,4) berapa berderajat polinomial generator tertinggi yang diperoleh dari percobaan ? apakah sudah sesuai dengan teori ?

A.2. Membangkitkan matrik parity dan matrik generator kode siklik (7,4) dengan MATLAB.

```
>> [h,g]=cyclgen(7, p)  
h =  
    1   0   0   1   1   1   0  
    0   1   0   0   1   1   1  
    0   0   1   1   1   0   1  
g =  
    1   0   1   1   0   0   0  
    1   1   1   0   1   0   0  
    1   1   0   0   0   1   0  
    0   1   1   0   0   0   1
```

2. Cobalah mengulangi langkah A2 dengan menggunakan polinomial p yang lain, misal : $p1=[1 0 1 1 0]$, $p2=[1 1 0 1 1 0]$, $p3=[1 0 1 1 1 0 1]$. Apa yang terjadi ? apakah ada kaitannya antara p , h dan g pada comman program tersebut ? Disebut apakah p , h dan g menurut teori ?

B. ENKODING DAN DEKODING KODE SIKLIK (Tanpa Error)

B1. Mengamati hasil pengkodean (encoding) kode siklik

```
>> p=cyclpoly(7,4);
```

```

p =
1 0 1 1
>> [h,g]=cydgen(7, p);
g =
1 0 1 1 0 0 0
1 1 1 0 1 0 0
1 1 0 0 0 1 0
0 1 1 0 0 0 1
>> pesan=[0 0 1 1]
pesan =
0 0 1 1
>> codeword=encode(pesan,7,4,'cyclic')
codeword =
1 0 1 0 0 1 1

```

3. Bagaimana pola dari hasil pengkodean? Adakah kaitannya dengan struktur kode siklik ?
4. Cobalah mengulangi langkah B1, dengan mengganti pesan menjadi msg = [0 0 1 1]. Apakah pengkodean berhasil dengan baik ? Beri penjelasan secara teoritis.

B2. Mengamati hasil pengkodean kembali (decoding) kode siklik

```

>> pesan_terima=decode(codeword,7,4,'cyclic')
pesan_terima =
0 0 1 1

```

5. Cek hasil pesan yang dikirim dengan pesan yang diterima setelah mengalami proses coding dan decoding, apakah hasilnya sama ? mengapa demikian ?

% Gunakan perintah pengecekan

```

>> cek=[pesan pesan_terima]
cek =
0 0 1 1 0 0 1 1
>> cek=[pesan' pesan_terima']

```

```

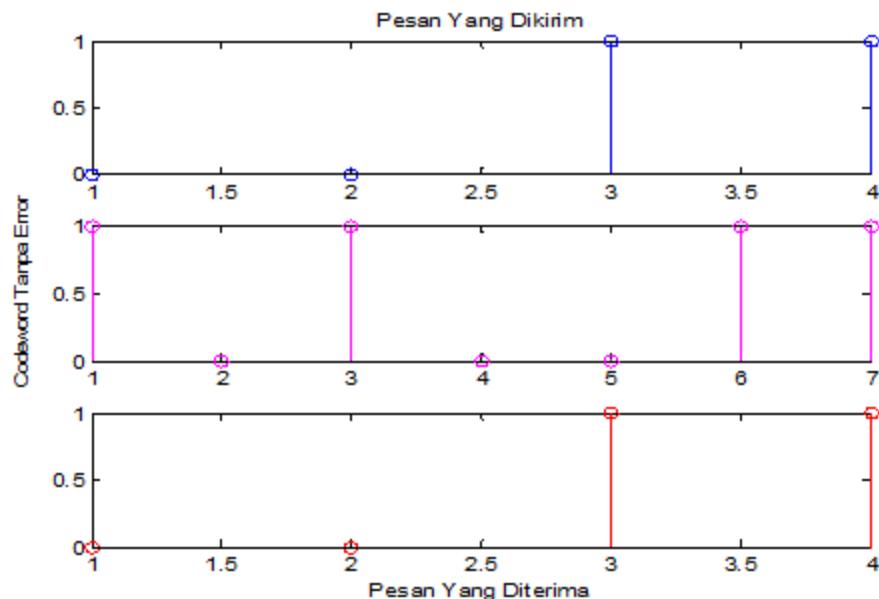
cek =
    0   0
    0   0
    1   1
    1   1

%Mengamati bit error dan rationya
>> [number, ratio] = biterr(pesan',pesan_terima')

number =
    0
ratio =
    0

% Mengeplot pesan tanda noise
>> subplot(3,1,1)
>>stem(pesan'b')
>>title('Pesanan Yang Dikirim')
>>subplot(3,1,2)
>>stem(codeword,'m')
>>ylabel('Codeword Tanpa Error')
>>subplot(3,1,3)
>>stem(pesan_terima,'r')
>>xlabel('Pesanan Yang Diterima')

```



C. ENKODING DAN DEKODING KODE SIKLIK (Ditambah Dengan Error)

```
% Untuk mempermudah pengamatan, codeword tanpa error diambil dari percobaan sebelumnya
```

```
% Tambahkan error pada bit ke 5 pada codeword sebelumnya
```

```
>> error=zeros(7,1);
```

```
>> error([5],1)=1
```

```
error =
```

```
0  
0  
0  
0  
1  
0  
0
```

```
% Penambahan noise pada codeword
```

```
>> code_error=xor(error,codeword')
```

```
code_error =
```

```
1  
0  
1
```

```

0
1
1
1

>> code_noise=fix(code_error)
code_noise =
1
0
1
0
1
1
1

>> cek=[codeword ' code_noise]
cek =
1 1
0 0
1 1
0 0
0 1
1 1
1 1

```

6. Cek hasil code dan code terima setelah ditambah dengan error ? pada bit keberapa codeword menjadi tidak sama ? Beri penjelasan secara ilmiah.

%Proses decoding dari codeword bernoise

```

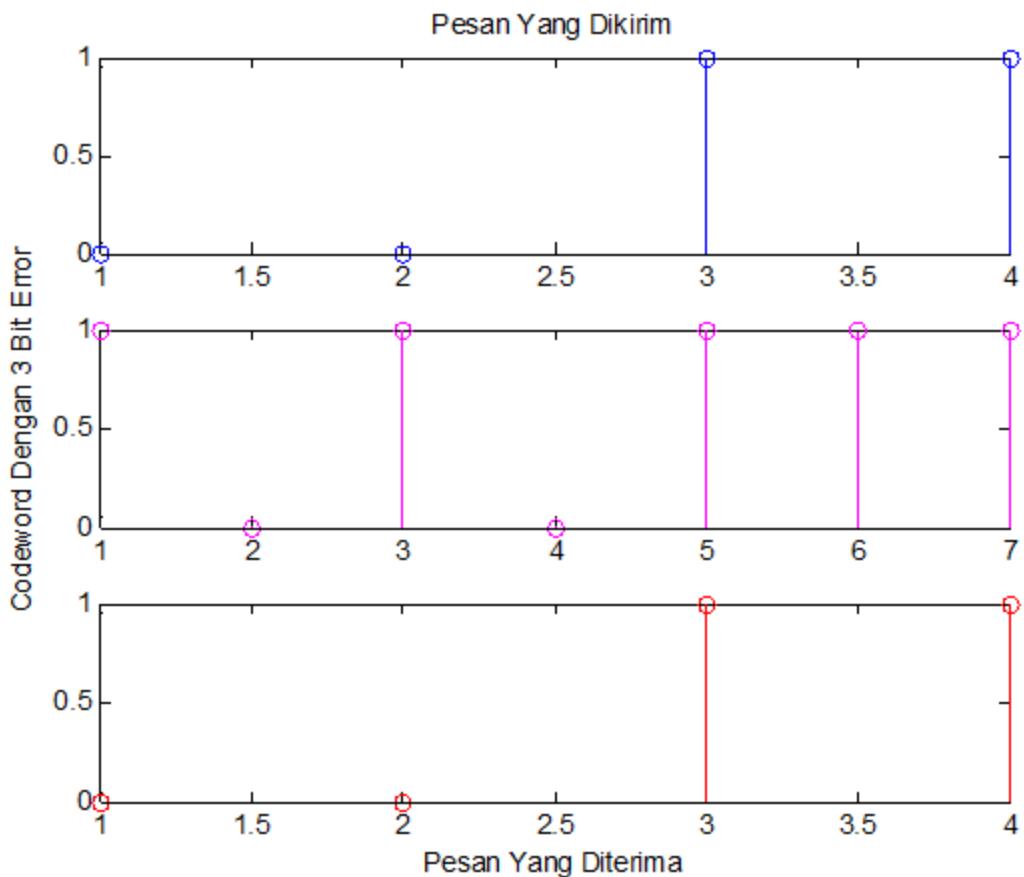
>> pesan_terima=decode(code_noise,7,4,'cyclic')
pesan_terima =
0
0
1
1

>> cek_pesan=[pesan' pesan_terima]
cek_pesan =
0 0
0 0
1 1

```

1 1

```
% Mengeplot pesan_terima dengan noise
>> subplot(3,1,1)
>> stem(pesan,'b')
>> title('Pesan Yang Dikirim')
>> subplot(3,1,2)
>> stem(code_terima,'m')
>> ylabel('Codeword Dengan 3 Bit Error')
>> subplot(3,1,3)
>> stem(pesan_terima_noise,'r')
>> xlabel('Pesanan Yang Diterima')
```



7. Cek pesan sebelum dikodekan maupun setelah dikodekan, bagaimana menurut anda ? apakah bit error yang ditambahkan berpengaruh terhadap informasi yang diterima ? Beri penjelasan secara teoritis.

TUGAS :

1. Buat semua program dalam satu m-file (mulai langkah A1 sampai C6)
2. Buat program tambahan untuk penambahan error dengan jumlah bit error dan posisi bit error diganti-ganti.

LAPORAN RESMI MELIPUTI LANGKAH A, B dan C yang dilengkapi dengan :

1. Laporan Sementara
2. Listing program
3. Hasil simulasi
4. Analisa lengkap

--oo0oo--