

PERCOBAAN 4

METODE DETEKSI DAN KOREKSI PADA KODE SIKLIK

1. Tujuan :

Setelah melakukan praktikum, diharapkan mahasiswa dapat :

- Membangkitkan generator siklik dan bit informasi untuk kode CRC-12 dan CRC-16 menggunakan matlab.
- Menghasilkan *codeword* dari data informasi yang telah ditentukan menggunakan generator polinomial CRC-12 dan CRC-16.
- Melakukan proses deteksi dan koreksi kode siklik dengan cara melakukan proses perhitungan sindrom samapi melakukan proses decoding dari *codeword* yang diperoleh dari proses sebelumnya
- Percobaan dilakukan dengan menambahkan noise sampai batas yang memungkinkan.

2. Teori Penunjang :

2.1.1 Metode Deteksi Error

Secara umum dikenal 2 metode pendeteksian error yang sederhana, yaitu : Vertical Redundancy Check (VCR) dan Longitudinal Redundancy Check (LCR), kedua sistem ini menggunakan bit parity untuk mendeteksi kesalahan.

2.1.2 Metode Koreksi Error

Sistem perbaikan error atau yang dikenal dengan istilah *error control* adalah mengupayakan agar data yang telah berubah dalam transmisi diperbaiki sebelum diproses lebih lanjut. Error dapat dikoreksi dengan 2 prinsip metode, yaitu : Sistem Automatic Repeat Request (ARQ) dan Sistem Forward Error Correction (FEC). Pada kode siklik digunakan sistem FEC dalam mengoreksi kesalahan. Sistem ini perbaikan akan dilakukan oleh decoder tanpa melakukan transmisi balik, selain itu juga akan menentukan lokasi dan perbaikan error yang terjadi. Dengan sistem ini dapat dicapai efisiensi karena tidak terdapat delay seperti pada sistem ARQ. Metode deteksi dan koreksi pada kode siklik,

akan dilakukan dengan cara menghitung sindrom, untuk mengetahui apakah word r (received word) yang diterima merupakan codeword c yang sesuai atau bukan. Bila hasil perhitungan sindrom adalah nol maka word r merupakan codeword yang dikirimkan. Sebaliknya jika hasil perhitungan sindrom bukan nol, berarti word r mengandung error yang membutuhkan proses koreksi.

2.1.3 Perhitungan Sindrom Kode Siklik

Perhitungan sindrom dapat dilakukan dengan menggunakan persamaan-persamaan berikut :

$$r(x) = d(x).g(x) + s(x) \dots \dots \dots (1)$$

dengan,

$r(x)$ adalah word r yang diterima.

$g(x)$ adalah generator matriks

$s(x)$ adalah sisa pembagian (merupaka sindrom s yang dicari)

atau dapat dinyatakan dengan :

$$\begin{aligned} s(x) &= r(x) \bmod g(x) \\ &= s_0 + s_1x + \dots + s_{n-k-1}x_{n-k-1} \dots \dots \dots (2) \end{aligned}$$

Dari persamaan tersebut, nilai $s(x)=0$ apabila nilai $r(x)=c(x)$ dan apabila nilai $s(x)$ tidak sama dengan 0 maka $r(x)$ bukan merupakan codeword $c(x)$ yang diharapkan karena masih terdapat error $e(x)$.

$$r(x) = c(x) + e(x) \dots \dots \dots (3)$$

3. Langkah Percobaan Deteksi dan Koreksi Kesalahan

1. Buat/tentukan struktur kode siklik (sesuaikan dengan jenis CRC yang akan digunakan)
2. Struktur kode siklik (n,k)
 n = panjang bit codeword
 k = panjang bit data
3. Bangkitkan polinomial generator menggunakan matlab.
 $g = \text{cyclpoly}(n,k) \rightarrow$ tuliskan $g(x)$ nya

4. Buat data/pesan
Tuliskan data dalam biner dan polynomialnya
Misal : $d = [1 \ 1 \ 0 \ 0 \ 1]$ dan $d(x) = x + x^1 + x^4$
5. Langkah berikutnya membuat codeword $c(x)$, dengan cara mengalikan data $d(x)$ dengan generator $g(x)$:
 $c(x) = d(x)g(x)$
Sehingga diperoleh codeword $c(x)$
6. Tambahkan noise/error sebanyak 2 bit (1), sehingga $c(x)$ menjadi $r(x)$
misal : $\text{noise}(x)$
7. Jadi codeword yang diterima : $r(x)$ tidak sama dengan $c(x)$,
sehingga $r(x) = c(x) + \text{noise}(x)$
8. Bobot error (wt) = adalah jumlah bit yang salah pada $r(x)$ karena adanya penambahan noise tersebut.
9. Hitung sindrom, sindrom adalah semua bit yang salah pada $r(x)$ selama proses transmisi. Oleh karena itu ada kemungkinan jumlah sindrom lebih banyak dari pada bobot error. Sindrom dihitung menggunakan rumus :
 $s(x) = r(x)(\text{mod } g(x)) \rightarrow$ sindrom merupakan sisa pembagiannya
10. Kemudian hitung : $s_i(x) = x^i s(x)(\text{mod } g(x))$ sampai sindrom s_i ditemukan
11. S_i diperoleh, jika $wt(s_i) \leq t$, dimana t = jumlah sindrom yang dapat dikoreksi oleh kode, dimana $t = \frac{n-k-1}{2}$.

Proses Deteksi, disinilah posisi noise/error pada $r(x)$ dapat diketahui dengan cara menghitung $e(x)$ dengan rumus :

$$e(x) = x^{n-i} s_i(x)$$

Proses koreksi dengan cara menghilangkan error, sehingga mendapatkan $c(x)$ seperti sebelum diberi noise.

$$c(x) = r(x) - e(x)$$

Lakukan proses decoding untuk mendapatkan $d(x)$ kembali

$$d(x) = c(x) \bmod g(x), \text{ dengan sindrom } s(x) = 0$$

3. CRC (Cyclic Redundancy Ceck)

Code	Generator polynomial $g(x)$	Parity check bits
CRC-12	$1+x+x^2+x^3+x^{11}+x^{12}$	12
CRC-16	$1+x^2+x^{15}+x^{16}$	16
CRC-CCITT	$1+x^5+x^{15}+x^{16}$	16

4. TUGAS DALAM PRAKTIKUM :

Buatlah program (m-file) kode siklik untuk CRC-12 dan CRC-16

1. Tentukan struktur kode siklik(n,k) sesuai dengan ketentuan CRC-12 dan CRC-16 (minimum pesan 8 bit)
2. Cek apakah polinomial generator $g(x)$ dari struktur yang akan digunakan dapat men-generate matrik generator \rightarrow cek menggunakan matlab.
3. Bangkitkan pesan $d(x)$ secara fix agar tidak kesulitan saat validasi (banyak bit pesan sesuai dengan struktur yang digunakan baik untuk CRC-12 maupun CRC-16).
4. Kodekan pesan $d(x)$ tersebut menggunakan kode siklik dengan struktur sesuai dengan CRC-12 dan CRC-16, sehingga menghasilkan codeword $c(x)$.
5. Tambahkan noise sebanyak 3 bit (1) error pada codeword $c(x)$, sehingga codeword menjadi $r(x)$
6. Kodekan kembali codeword yang sudah ditambah dengan noise $r(x)$ menjadi $d(x)$ dengan cara mencari $e(x)$ terlebih dahulu.

--oo0oo--