

# Perancangan Perangkat Lunak Pembelajaran Algoritma Hamming Code dalam Mencari Bit Error pada Komunikasi Data

Misdem Sembiring<sup>1</sup>, Fauzi Haris Simbolon<sup>2</sup>

<sup>1,2</sup>AMIK Medan Business Polytechnic

Jl. Jamin Ginting No. 285-287, Padang Bulan, Medan Baru, Kota Medan, Sumatera Utara, Indonesia - 20155

<sup>1</sup>misdem@amikmbp.ac.id, <sup>2</sup>farisboys@amikmbp.ac.id

DOI: xx.xxxx/j.ccs.xxxx.xx.xxx

## Abstrak

Dalam ilmu komputer terdapat bermacam-macam algoritma dalam mendeteksi serta mengoreksi error ketika transmisi data digital berlangsung. Salah satu algoritma untuk mendeteksi error adalah dengan menggunakan Hamming Code dengan single error correction. Metode ini sangat cocok digunakan pada situasi dimana terdapat beberapa error yang teracak (randomly occurring errors). Metode Hamming Code menyisipkan  $(n + 1)$  check bit ke dalam  $2n$  data bit. Metode ini menggunakan operasi XOR (Exclusive - OR) dalam proses pendeteksian error, dengan nilai Input dan output data berupa bilangan biner. Peneliti bermaksud untuk merancang perangkat lunak pembelajaran yang mampu menjelaskan teknik pendeteksian error dari algoritma Hamming Code. Panjang data input dan output pada algoritma Hamming Code minimal harus sama dengan 4 bit. Atau dengan perkataan lain, algoritma Hamming Code tidak mendukung data dengan panjang 2 bit. Algoritma Hamming Code tidak mampu melakukan pengecekan terhadap posisi data error (bad bit) yang lebih dari satu buah.

*Kata Kunci:* Hamming Code, Perancangan Perangkat Lunak, Komunikasi Data, Bit Error.

## 1. Pendahuluan

Ketika transmisi data digital berlangsung, terkadang data yang ditransfer dapat mengalami kegagalan (Error). Error yang terjadi dapat mengakibatkan perubahan isi dari data yang ditransfer. Dalam ilmu komputer terdapat bermacam-macam algoritma dalam mendeteksi serta mengoreksi error tersebut. Salah satu algoritma untuk mendeteksi error adalah dengan menggunakan Hamming Code dengan single error correction.

Hamming Code merupakan algoritma pendeteksi error yang mampu mendeteksi beberapa error, namun hanya mampu mengoreksi satu error (single error correction). Algoritma ini sangat cocok digunakan pada situasi dimana terdapat beberapa error yang teracak (randomly occurring errors). Algoritma Hamming Code menyisipkan  $(n + 1)$  check bit ke dalam  $2n$  data bit. Algoritma ini menggunakan operasi XOR (Exclusive - OR) dalam proses pendeteksian error, dengan nilai input dan output data berupa bilangan biner.

Pendeteksian error mulai dari awal hingga akhir dalam algoritma Hamming Code berjalan dengan sangat cepat sehingga sulit untuk dijelaskan. Agar dapat mempelajari proses pendeteksian error algoritma Hamming Code secara tahap demi tahap

peneliti bermaksud untuk merancang perangkat lunak pembelajaran yang mampu menjelaskan teknik pendeteksian error dari algoritma Hamming Code.

## 2. Tinjauan Literatur

Peneliti tidak luput meninjau penelitian-penelitian yang relevan agar dapat menyelesaikan penelitian ini dengan baik.

Dalam peninjauan akan penelitian yang dilakukan oleh Maulana et al (2014) yang berjudul Perancangan Simulasi Pengkodean Hamming (7, 4) untuk Menghitung Bit Error Rate (BER) pada Binary Symmetric Channel. Tujuan peneliti untuk merancang sebuah aplikasi simulasi menggunakan bahasa pemrograman Java untuk menggambarkan proses pengkoreksian error pada pengiriman data berupa angka dalam bentuk integer dan akan diubah menjadi biner untuk memudahkan dalam proses pengecekan error. Algoritma yang digunakan proses pengecekan error pada aplikasi simulasi adalah Hamming Code. Dari hasil pengamatan aplikasi ini, data dikirimkan akan di deteksi jika terjadi kesalahan maka aplikasi akan mengoreksi kesalahan yang telah terdeteksi. Hasil pengamatan menunjukkan bahwa error terjadi pada saat pengiriman data di karenakan kesalahan pada bit-bit yang dikirimkan, maka terjadilah error. Dan juga diharapkan dengan simulasi yang dibuat ini

bisa membantu dalam memahami tentang proses pengiriman data dan bagaimana pengkoreksian error tersebut. Kesimpulan yang didapatkan dari penelitian ini bahwa aplikasi mensimulasikan proses pengkodean dengan Kode Hamming (7,4) dan menghitung bit error rate dengan menggunakan Binary Symetric Channel (BSC) dan pengkodean Hamming (7,4) dapat mendeteksi tepat satu error, dan pengkodean Hamming dapat memperkecil bit error rate dalam sistem komunikasi digital.

Tinjauan pada penelitian Andanal (2018) dengan judul Implementasi Deteksi dan Koreksi Error Pada Komunikasi Serial Arduino Berbasis UART Dengan Metode Hamming Code, peneliti menggunakan metode Hamming Code yang diterapkan pada Arduino dan komunikasi UART. Berdasarkan hasil pengujian, algoritma Hamming Code dapat melakukan proses encode dan decode data, serta dapat melakukan deteksi dan koreksi error pada data yang mengalami error dalam proses pengujian. Rata-rata delay yang didapatkan berjumlah 102,7ms untuk data 5 bit dan 109,5ms untuk data 4 bit pada proses encode. Serta 17,5 ms untuk data 10 bit dan 100,1ms untuk data 11 bit pada proses decode. Faktor pengambilan data suhu serta jumlah bit yang dilakukan proses encode dan decode sangat mempengaruhi proses encode dan decode menggunakan algoritma Hamming Code.

Dari peninjauan penelitian-penelitian yang telah dipaparkan diatas, peneliti akan memaparkan beberapa landasan teori yang digunakan dalam penelitian ini, antara lain:

### 2.1. Error Detection

Pada saat data berada dalam transmission system terdapat kemungkinan data terkorupsi (data error). Data error tersebut akan diperbaiki oleh receiver melalui proses error detection dan error correction. Proses error detection dilakukan oleh transmitter dengan cara menambahkan beberapa bit tambahan ke dalam data yang akan ditransmisikan. Proses error detection dan correction ini sering digunakan pada CD Players, High speed modem, dan telepon selular (cellular phones).

Secara garis besar, metoda pendeteksi error (error detection) dan pengontrol error (error controller) dapat dibagi menjadi 2 bagian besar dengan perincian sebagai berikut:

1. Error Detection and Correction.
2. Error Controller, yaitu: Automatic Repeat Request (ARQ).

### 2.2. Hamming Code

Algoritma Hamming Code ditemukan oleh Richard W. Hamming pada tahun 1940-an. Algoritma Hamming Code merupakan salah satu metoda pendeteksi error (error detection) yang mampu untuk mendeteksi beberapa error, namun hanya mampu mengoreksi satu error (single error correction). Algoritma pendeteksi error ini sangat cocok digunakan pada situasi dimana terdapat beberapa error yang teracak (randomly occurring errors).

Algoritma Hamming Code merupakan salah satu algoritma pendeteksi error (error detection) dan pengoreksi error (error correction) yang paling sederhana. Algoritma ini menggunakan operasi logika XOR (Exclusive – OR) dalam proses pendeteksian error (error detection) maupun proses pengoreksian error (error correction), sedangkan input dan output data dari algoritma Hamming Code berupa bilangan biner.

#### 1. Operasi Logika

Operasi – operasi logika dasar terdiri dari operasi NOT, operasi AND, dan operasi OR. Operasi logika lain merupakan kombinasi dari operasi - operasi logika dasar ini. Salah satu operasi logika hasil kombinasi operasi logika dasar adalah operasi XOR (Exclusive - OR).

#### 2. Cara Kerja Algoritma Hamming Code

Algoritma Hamming Code menyisipkan beberapa buah check bit ke dalam data. Jumlah check bit yang disisipkan tergantung pada panjang data. Rumus perhitungan untuk menghitung jumlah check bit yang harus disisipkan ke dalam data adalah sebagai berikut.

Untuk data  $2n$  bit, jumlah check bit yang disisipkan ada sebanyak  $c = (n + 1)$  bit.

## 3. Metodologi Penelitian

Lokasi dari penelitian ini berada di transmisi komunikasi data.

### 3.1. Pengumpulan Data

Untuk menyelesaikan penelitian ini, peneliti mengumpulkan data-data yang dibutuhkan dari berbagai sumber dengan beberapa metode, antara lain:

1. Studi lapangan  
Degan metode ini peneliti mengamati bagaimana data dapat berpindah dengan dari satu tempat ke tempat lain di perangkat komputer yang terhubung dengan internet. Peneliti juga mengamati data-data

yang berpindah kenapa mengalami corrupt dan harus dilakukan ulang perpindahan data.

2. Studi Kepustakaan

Untuk memperdalam pengetahuan peneliti akan permasalahan yang sedang diteliti. Peneliti melakukan studi pustaka yang berhubungan dengan komunikasi data. Pengetahuan yang didapatkan berupa teknik pendeteksian error dari algoritma-algoritma dalam mentransmisi data khusus Hamming Code serta hal-hal lain yang berhubungan dengan jaringan internet.

3.2. Metode Analisis Data

Dalam menyelesaikan penelitian ini yang bersifat deduktif dimana menganalisa data dengan cara mengambil kesimpulan berdasarkan teori algoritma Hamming Code yang telah diterima sebagai suatu kebenaran umum mengenai fakta yang diamati.

4. Hasil dan Pembahasan

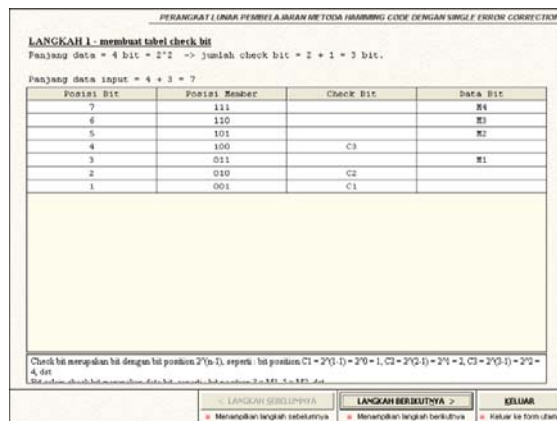
Hasil dari penelitian akan dipaparkan disini dan juga tentang pembahasannya:

4.1. Hasil

Hasil dari penelitian ini berupa perangkat lunak yang dapat menjelaskan teknik pendeteksian error dari algoritma Hamming Code.

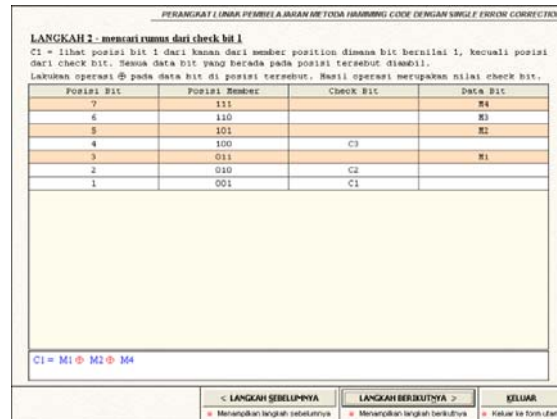
Misalkan panjang data input dan output = 4 bit, data Input = 1001, maka , langkah-langkah pendeteksian single error dengan Hamming Code adalah sebagai berikut:

1. Membuat tabel check bit



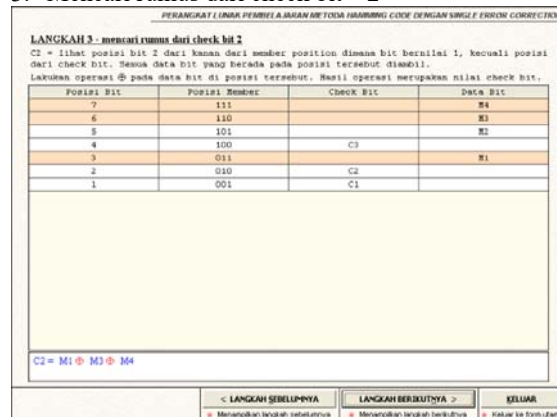
Gbr. 1. Membuat tabel check bit untuk data Input = 1001, data Output = 1000 (panjang data = 4 bit)

2. Mencari rumus dari check bit - 1



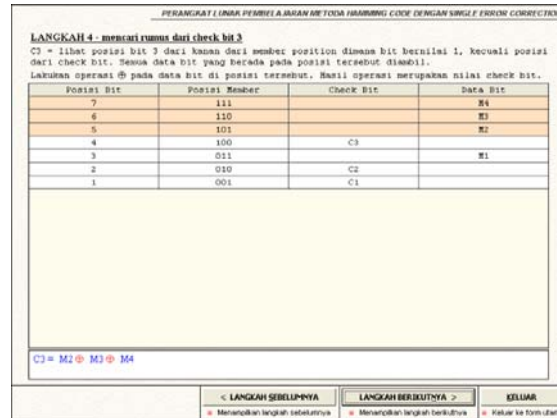
Gbr. 2. Mencari rumus dari check bit - 1 untuk data input = 1001, data output = 1000 (panjang data = 4 bit)

3. Mencari rumus dari check bit - 2



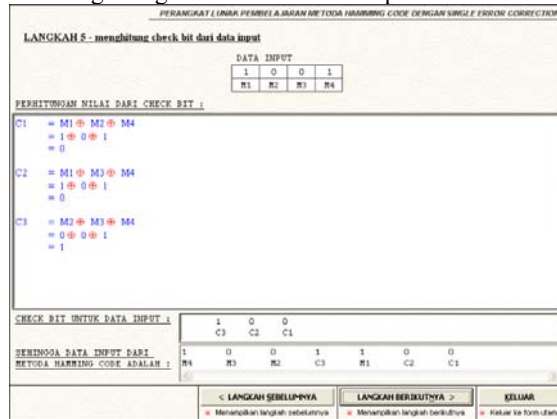
Gbr. 3. Mencari rumus dari check bit - 2 untuk data input = 1001, data output = 1000 (panjang data = 4 bit)

4. Mencari rumus dari check bit-3



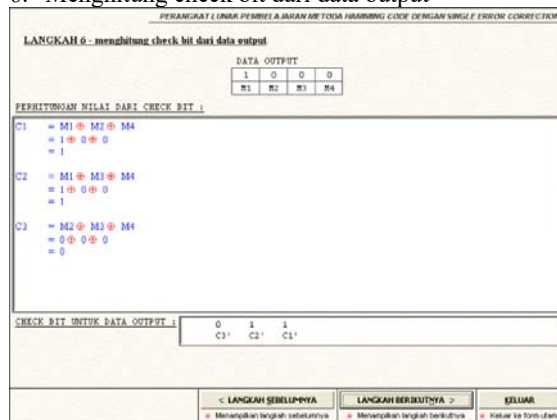
Gbr. 4. Mencari rumus dari check bit - 3 untuk data input = 1001, data output = 1000 (panjang data = 4 bit)

5. Menghitung check bit dari data input



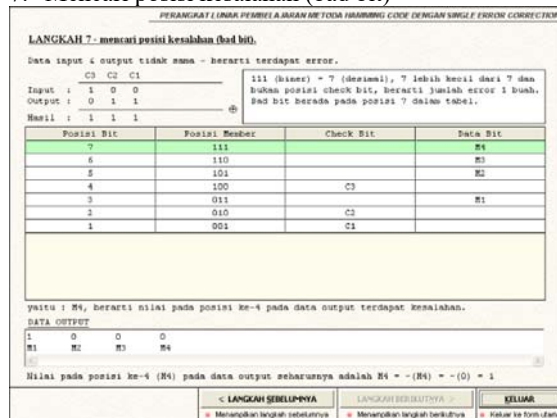
Gbr. 5. Menghitung check bit dari data input untuk data input = 1001, data output = 1000 (panjang data = 4 bit)

6. Menghitung check bit dari data output



Gbr. 6. Menghitung check bit dari data output untuk data input = 1001, data output = 1000 (panjang data = 4 bit)

7. Mencari posisi kesalahan (bad bit)



Gbr. 7. Mencari posisi kesalahan (bad bit) untuk data input = 1001, data output = 1000 (panjang data = 4 bit)

4.2. Pembahasan

Algoritma Hamming Code melakukan proses pengecekan error dengan cara menyisipkan  $n - 1$  check bit untuk  $2n$  bit data (input dan output). Lalu dilakukan perhitungan nilai dari check bit tersebut untuk data input dan data output. Setelah itu, nilai check bit tersebut di-XOR-kan dan dilakukan perbandingan antara nilai check bit input dan nilai check bit output. Apabila nilainya tidak sama, maka terdapat error (kesalahan). Sebaliknya, apabila nilainya sama maka tidak terdapat error dalam data output.

Error tersebut dapat terjadi karena adanya gangguan pada saat transmisi data. Error yang muncul dapat berupa single error (error tunggal) yaitu error yang terjadi hanya pada 1 bit data saja, ataupun double error (error ganda) yaitu error yang terjadi pada beberapa bit data. Metoda Hamming Code hanya dapat digunakan untuk mendeteksi dan mengatasi single error. Sedangkan untuk double error, metoda Hamming Code hanya dapat mendeteksinya, namun tidak mampu mengatasinya, karena tidak dapat mengetahui posisi bit error (bad bit). Keunggulan dari metoda Hamming Code hanya terletak pada cara kerjanya cukup sederhana (simple).

5. Kesimpulan dan Saran

Dari hasil penelitian dan pembahasannya, maka dapat disimpulkan dan diberikan saran sebagai berikut:

5.1. Kesimpulan

Setelah menyelesaikan perancangan perangkat lunak pembelajaran Hamming Code dengan Single Error Correction, maka peneliti dapat menarik kesimpulan sebagai berikut:

1. Algoritma Hamming Code hanya mampu melakukan pengoreksian terhadap satu buah error (single error correction) saja.
2. Data input dan output pada algoritma Hamming Code harus berupa hasil dari perpangkatan  $2n$  dengan  $n$  harus lebih besar dari 1.
3. Panjang data input dan output pada algoritma Hamming Code minimal harus sama dengan 4 bit. Atau dengan perkataan lain, algoritma Hamming Code tidak mendukung data dengan panjang 2 bit.

## 5.2. *Saran*

Peneliti ingin memberikan beberapa saran untuk pengembangan lebih lanjut pada perancangan perangkat lunak pembelajaran metoda pendeteksi dan pengoreksi kesalahan (error detection and correction). Saran-saran tersebut antara lain:

1. Dapat dikembangkan suatu perangkat lunak dengan menggunakan metoda pendeteksi dan pengoreksi error (error detection and correction method) lainnya yang lebih canggih daripada metoda Hamming Code seperti metoda CRC (Cyclic Redundancy Check), gabungan metoda LRC (Longitudinal Redundancy Check) dan VRC (Vertical Redundancy Check) yang mampu melakukan pengecekan terhadap lebih dari satu buah error dan sebagainya.
2. Perangkat lunak dapat ditambahkan beberapa animasi lainnya agar lebih menarik.

Perangkat lunak juga dapat dikembangkan menjadi sebuah perangkat lunak multimedia dengan menambahkan efek suara, efek grafik dan efek-efek lainnya

## Referensi

- [1] Maulana, J., Arini, & Fahrianto, F. (2014). Perancangan Simulasi Pengkodean Hamming (7, 4) untuk Menghitung Bit Error Rate (BER) pada Binary Symmetric Channel. *Jurnal Teknik Informatika*, 7(2), 24-34.
- [2] Andana1, A. F., Akbar, S. R., & Maulana, R. (2018). Implementasi Deteksi Dan Koreksi Error Pada Komunikasi Serial Arduino Berbasis UART Dengan Metode Hamming Code. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(11), 5089-5095.
- [3] Green, D. C. (2000). *Komunikasi Data*. Yogyakarta: Andi.
- [4] Ralph J. S. (1990). *Rangkaian, Piranti dan Sistem*, Edisi Keempat, Jilid 1. Jakarta: Penerbit Erlangga.
- [5] Stallings, W. (2001). *Dasar-dasar Komunikasi Data*. Jakarta: Penerbit Salemba Teknika.
- [6] Malvino, Albert P., & Tjia M, O. (1994). *Elektronika Komputer Digital*, Edisi Kedua. Jakarta: Penerbit Erlangga.
- [7] Pramono, D. (2002). *Mudah menguasai Visual Basic 6*. Jakarta: PT. Elex Media Komputindo.
- [8] Rahadian, H. (2001). *Pemrograman Microsoft Visual Basic 6.0*. Jakarta: PT. Elex Media Komputindo.