

Implementasi Cyclic Redudancy Check dalam Mendeteksi Bit Error pada Transmisi Data

Sony Bahagia Sinaga ¹, Satria Yudha Prayogi ², Frans Ikorasaki ³

¹ AMIK STIEKOM Sumatera Utara, Rantauprapat, Indonesia

² Universitas Islam Sumatera Utara, Medan, Indonesia

³ Universitas Potensi Utama, Medan, Indonesia

INFORMASI ARTIKEL

Diterima Redaksi: 8 Juni 2022
Revisi Akhir: 20 Juni 2022
Diterbitkan Online: 29 Juni 2022

KATA KUNCI

CRC, Transmisi, Data.

KORESPONDENSI

Phone: -

E-mail: sonybahagia@gmail.com

A B S T R A K

Pengiriman informasi di dalam dunia telekomunikasi sering kali terjadi kesalahan terhadap data yang dikirim. Kesalahan tersebut disebabkan adanya gangguan pada level fisik, yaitu gangguan pada media saluran transmisi, seperti gangguan radiasi elektromagnetik, cakap silang, petir atau adanya gangguan akibat noise. Gangguan ini menyebabkan informasi yang diterima tidak sesuai dengan informasi yang dikirimkan. CRC adalah salah satu metode yang dapat digunakan untuk mendeteksi kesalahan pada sistem tersebut. Pada dasarnya CRC menggunakan perhitungan matematika terhadap sebuah bilangan yang disebut *checksum* (disebut juga nilai CRC), yang dibuat berdasarkan total bit yang akan ditransmisikan. Untuk mempermudah proses perhitungan nilai CEC maka dibuatlah suatu langkah atau algoritma yang dapat diimplementasikan kedalam bentuk *software* (bahasa pemrograman). Dalam penelitian ini dibuat suatu algoritma yang menggunakan CRC, dan mengimplementasikannya ke dalam bahasa pemrograman *Visual Basic*.

PENDAHULUAN

Pada dunia telekomunikasi proses pengiriman informasi sangat penting. Adakalanya informasi yang diterima tidak sesuai dengan informasi yang dikirimkan. Penyebabnya adalah terdapat gangguan yang terjadi pada level fisik, yaitu pada media atau saluran komunikasi yang digunakan adanya gangguan radiasi elektromagnetik, menjaga cakap silang, petir atau karena adanya gangguan akibat noise.

Kesalahan adalah proses alami yang dapat terjadi pada tiap bagian dari sistem komunikasi data. Namun demikian perlu adanya langkah-langkah bagi perbaikan melalui evaluasi terhadap penyebab terjadinya kesalahan dan menyelidiki kemungkinan-kemungkinan terjadinya kesalahan dalam proses transmisi maupun data terminal. Saluran komunikasi yang secara fisik menghubungkan dua mesin secara konseptual bekerja seperti halnya kabel. Sifat penting sebuah saluran yang menyerupai kabel adalah bahwa *bit-bit* diteruskan dalam urutan yang sama dengan sewaktu *bit-bit* itu dikirimkan. Data *link layer* mempunyai fungsi yang spesifik, fungsi-fungsi ini meliputi penyediaan interface bagi layanan yang baik bagi *network layer*. Rangkaian komunikasi sering membuat kesalahan, memiliki laju data yang terbatas dan dapat *delay* propagasi yang tidak nol antara saat bit dikirimkan dengan saat bit diterima. Keterbatasan ini mempunyai implikasi penting bagi efisiensi perpindahan data. [1]

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*). [2]

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

Banyaknya *Hamming Code* yang dibutuhkan bertambah sesuai dengan banyaknya data dan posisi yang harus diisi dengan data tersebut [3].

METODOLOGI

Pada tahap ini dilakukan dengan mempelajari teori dasar yang mendukung penelitian, pencarian dan pengumpulan data-data yang dibutuhkan. Untuk mengumpulkan data yang dibutuhkan, maka penulis memakai teknik :

Penelitian Kelapangan

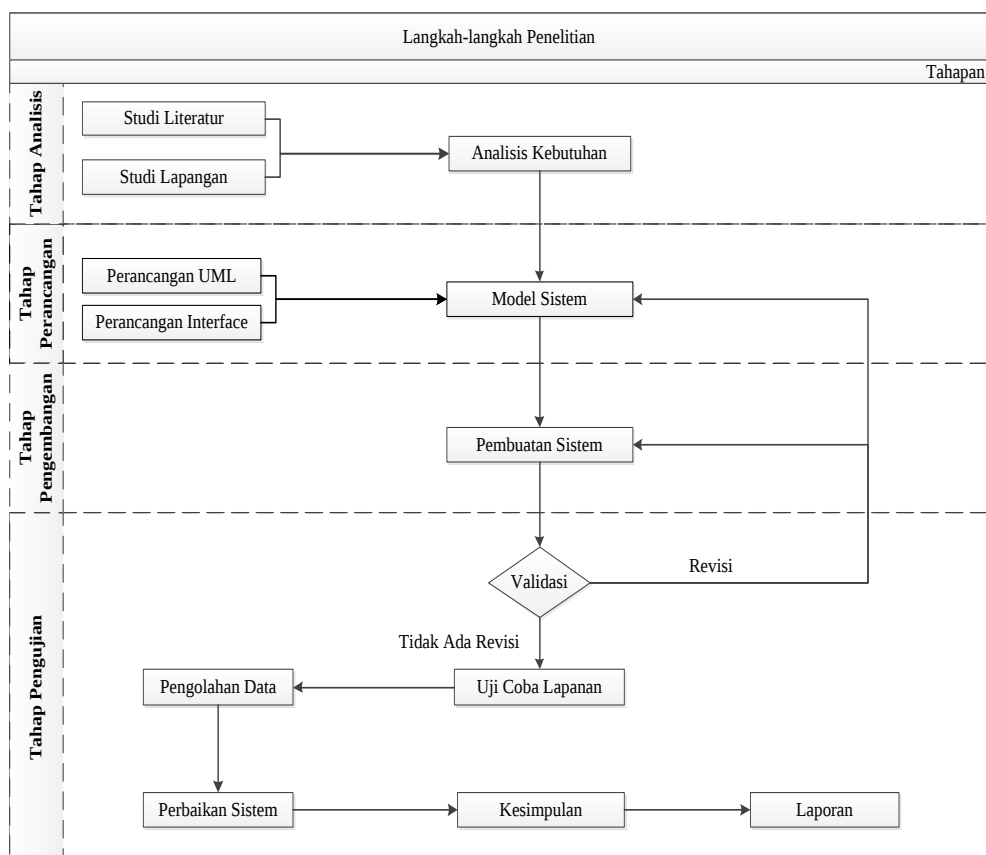
1. Pengamatan Langsung (*Observation*)
2. Penelitian perpustakaan (*Library Research*)

Pada metode ini penulis mengutip dari beberapa bacaan yang berkaitan dengan pelaksanaan skripsi yang dikutip dapat berupa teori yaitu jurnal.

Metodologi penelitian yang digunakan dalam pendeteksian bit error dalam transmisi data antara lain :

Bagan Alur Penelitian

Bagan alur dari penelitian ini dapat dilihat seperti gambar 1 berikut ini:



Gambar 1. Bagan Alur Penelitian

Tahap Analisis

Pada tahap ini dilakukan proses analisis kebutuhan terhadap sistem yang akan dirancang dan dibangun agar dapat dipahami oleh perangkat lunak seperti apa yang dibutuhkan oleh user berdasarkan studi literature dan studi lapangan. Pada tahap ini juga di analisa kebutuhan perangkat lunak dan perangkat keras dari sistem yang akan dirancang. Spesifikasi kebutuhan perangkat lunak dan perangkat keras pada tahap ini perlu untuk didokumentasikan. Suatu *Software Requirements Spesification* harus mencantumkan tentang deskripsi dengan lingkungannya. Adapun spesifikasi kebutuhan dari sistem yang akan dibangun adalah sebagai berikut :

Spesifikasi perangkat keras yang dibutuhkan adalah:

1. Laptop/PC
2. Ram 4GB
3. Hardisk 500 GB
4. Lolipop 5.0
5. Dual Core
6. RAM 1G (Android)

Spesifikasi perangkat Lunak yang dibutuhkan adalah

1. Sistem operasi Windows 8.1
2. Web Server XAMPP-PHP-MySQL
3. *Android Studio*
4. Prosesor Core I3

Tahap analisis kebutuhan terdiri dari:

Studi Literatur

Studi Literatur Mengumpulkan sumber-sumber yang dapat dijadikan sebagai bahan studi literatur dari jurnal ilmiah, dan hasil hasil penelitian kemudian melakukan identifikasi masalah, rumusan masalah, tujuan dan manfaat penelitian dengan melakukan observasi penelitian ini.

Tahap Perancangan

Tahap perancangan merupakan tahap permodelan dari sistem yang akan dibangun. Permodelan Sistem merupakan suatu bentuk penyederhanaan dari sebuah elemen dan komponen yang sangat kompleks untuk memudahkan pemahaman dari informasi yang dibutuhkan. Tahap perancangan ini terdiri atas :

Perancangan UML

Pada tahap ini dilakukan perancangan dari modelan sistem yang akan dibangun menggunakan model UML. Setelah dirancang model sistem yang akan dibangun selanjutnya dilakukan perancangan interface dari sistem yang akan dibangun.

Perancangan Interface

Pada tahap ini dilakukan proses perancangan *interface* yakni dengan merancang bentuk tampilan dari sistem yang akan dibangun baik itu tampilan halaman input, halaman proses dan halaman output sehingga dapat memberikan kemudahan.

Tahap Pengujian (Validasi)

Validasi merupakan proses untuk menunjukkan seberapa besar nilai keakuratan program terhadap kondisi-kondisi saat pemakaian sebenarnya. Proses ini menjalankan skenario berdasarkan data dan lingkungan yang merepresentasikan dunia nyata dengan menggunakan mesin percobaan. Pada tahap ini akan terjadi beberapa perbaikan, baik itu dari segi perancangan sistem maupun pengembangan sistem sampai didapat sistem yang sesuai dan laporan yang dihasilkan sesuai dengan kebutuhan atau perlu dilakukan perbaikan.

HASIL DAN PEMBAHASAN

Untuk mendapat nilai CRC melakukan perhitungan sebagai berikut:

Bit data yang akan dikirim oleh *transmitter*.

Pesan akan ditambahkan dengan bit nol sebanyak lebar bit polinomial.

Pesan = 110101

Polinomial = 101

Dalam hal ini, lebar bit polinomial adalah 2, maka pesan akan ditambah dengan 00 menjadi 11010100, dan akan dibagi dengan polinomial. Pembagian tersebut sama dengan melakukan XOR pada semua bit yang dibagi dengan bit pembagi.

Perhitungannya [p(x)] adalah sebagai berikut:

11010100 / 101

$$\begin{array}{r}
 11010100 \\
 \underline{101} \\
 111 \\
 \underline{101} \\
 100 \\
 \underline{101} \\
 010 \\
 \underline{101} \\
 110 \\
 \underline{101} \\
 11 \quad \text{sisasisa} = \text{CRC checksum}
 \end{array}$$

Pesan yang didapat dengan CRC [px(x)] = 11010111

Perhitungan bit data yang diterima oleh *receiver*, dapat dilakukan dengan dua cara:

Cara I

1. *Receiver* akan memisahkan pesan dan *checksum*, kemudian akan menghitung *checksum* untuk pesan setelah menambahkan bit nol sebanyak lebar bit polinomial.
2. *Receiver* akan membandingkan kedua *checksum* tersebut, termasuk yang diterima dan yang dihitung.
3. Jika *checksum* tersebut sama besar, maka tidak terjadi *error*. Sebaliknya, bila kedua *checksum* tidak sama maka telah terjadi *error*.

Pesan = 11010111

Checksum pesan yang diterima = 11

Polinomial = 101

11010100 / 101

$$\begin{array}{r}
 11010100 \\
 \underline{101} \\
 111 \\
 \underline{101} \\
 100 \\
 \underline{101} \\
 010 \\
 \underline{101} \\
 110 \\
 \underline{101} \\
 11 \quad \text{sisasisa} = \text{CRC checksum}
 \end{array}$$

Cara II

1. *Receiver* akan menghitung *checksum* untuk keseluruhan pesan.
2. Kemudian memeriksa apakah hasilnya sama dengan nol (berarti tidak terjadi *error*).

Pesan = 11010111

Checksum pesan yang diterima = 11

Polinomial = 101

11010111 / 101

11010111

101

111

101

100

101

111

101

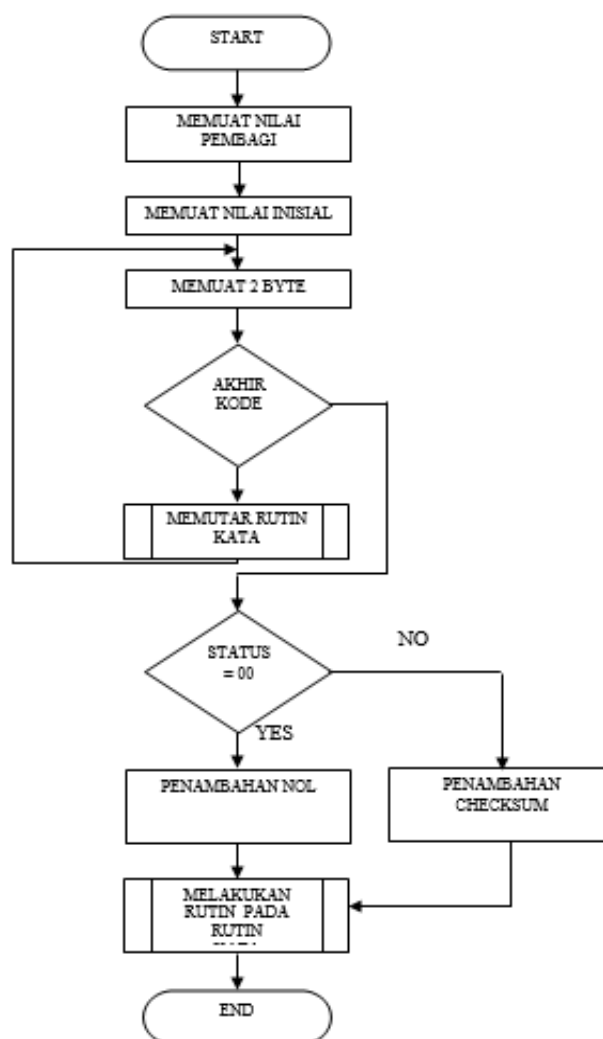
101

101

00

checksum = 0 (tidak terdapat *error*)**Flowchart**

Dalam penyusunan suatu program diperlukan suatu model data yang berbentuk diagram yang dapat menjelaskan suatu alur proses sistem yang akan di bangun. Dalam penulisan penelitian ini penulis menggunakan metode *flowchart* dapat dilihat pada gambar 2 berikut:

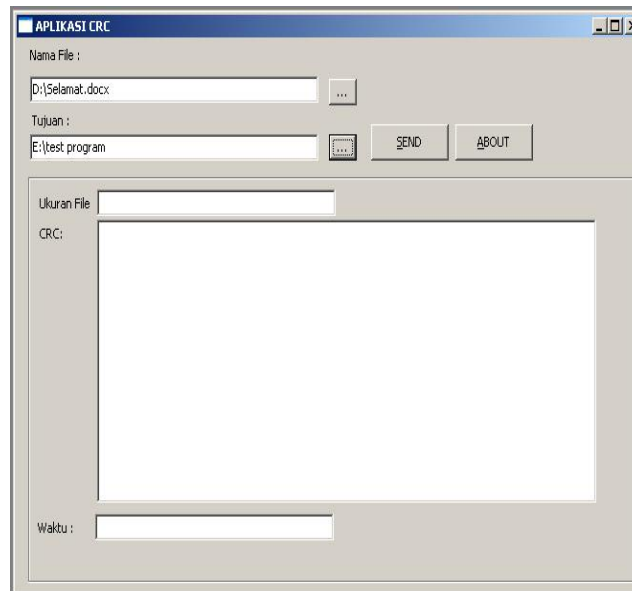


Gambar 2. Flowchart Diagram

Tampilan-tampilan dalam mendeteksi *bit error* pada transmisi data antara lain :

Mencari File

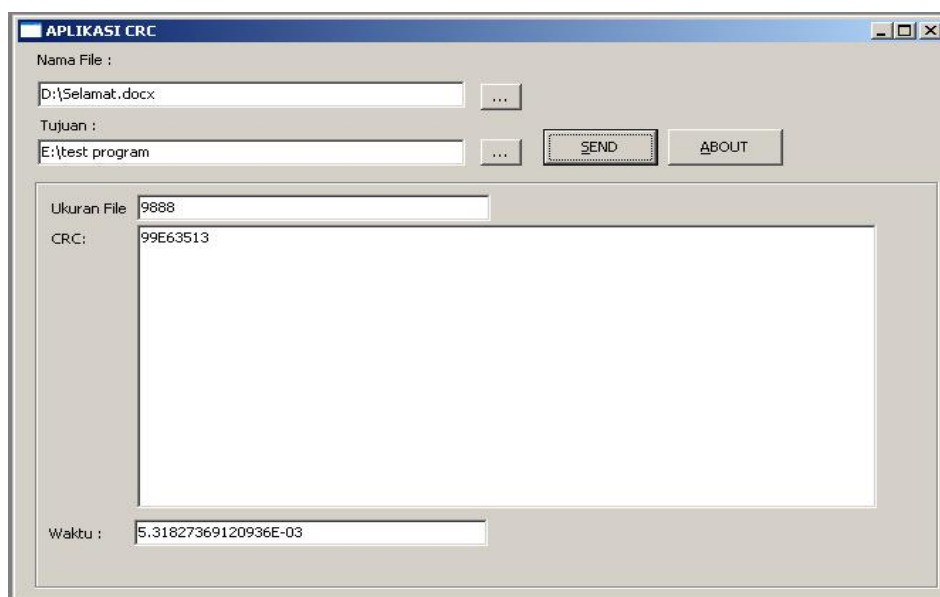
Untuk pencarian file yang akan dilakukan pendeteksian *bit error* dapat dilihat seperti gambar 3 berikut ini.



Gambar 3. Pencarian *file*

Proses *bit Error*

Untuk proses pendeteksian kesalahan *bit error*) dapat terlihat seperti gambar 4 berikut ini.



Gambar 4. Pendeteksian *bit error*

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan selama membuat aplikasi ini, maka dapat ditarik kesimpulan bahwa metoda *Cyclic Redundancy Check* hanya mampu mendeteksi namun tidak mampu mengoreksi *bit error*. Namun kelebihan dari *Cyclic Redundancy Check* adalah bahwa CRC mampu mendeteksi kesalahan/ *bit error* dalam kapasitas besar. Data *input* dan *output* pada metode *Cyclic Redundancy Check* harus berupa bilangan 0 dan 1. Terciptanya sistem Informasi pemesanan Room SPA berbasis android dan mempermudah pelanggan dalam melakukan pemesanan room SPA melalui VIA ATM. Sistem yang dirancang oleh penulis dapat mempermudah receptionist dalam menerima pemesanan room SPA dan proses pembayaran yang efektif dengan cara memvalidasi proses pembayaran. Terciptanya sistem aplikasi pemesanan room SPA dengan menggunakan aplikasi *Android Studio* dan *database Mysql*.

DAFTAR PUSTAKA

- [1] Hidayat, dkk, 2017, *Jurnal : “Perancangan Sistem Informasi Penjualan Barang Handmade Berbasis Website Dengan Metode Waterfall”*, AMIK BSI, Bekasi, Jurnal SIMNASIPTEK, Vol. 1, No. 1.
- [2] Yusfrizal. 2018. *“Kode Autentikasi Hash Pada Pesan Teks Berbasis Android”*. Universitas Potensi Utama, Medan Jl. K.L.Yos Sudarso Km.6,5 No.3-A Medan, Vol. 8, No. 1, September 2018. 6-14.
- [3] Helmi Kurniawan, 2015. *“Perancangan Sistem Aplikasi Ujian Seleksi Pegawai Berbasis Online”*. Seminar Nasional Informatika 2015. 312-317
- [4] Budi Triandi, 2015, Pengembangan Aplikasi Kriptografi Berbasis *Bit Inserted Carrier (Bic)*
- [5] Roni Ameldi (2018) dengan judul Sistem Informasi Reservasi Lapangan Futsal Berbasis *Android* Pada Lapangan Futsal”
- [6] Arta Ulina Sitorus (2017) dengan judul “Sistem Informasi Reservasi Hotel Pada GM. Marsaringar Balige Berbasis Android”.
- [7] Debbi Defrina (2017) dengan judul “Aplikasi Pemesanan Makanan Dan Minuman *Online* Berbasis *Mobile Browser* Pada Restoran Tiga Saudara”.