

PERANCANGAN SIMULASI PENGKODEAN HAMMING (7,4) UNTUK MENGHITUNG BIT ERROR RATE (BER) PADA BINARY SYMETRIC CHANNEL

Janwar Maulana¹, Arini², Feri Fahrianto³

^{1,2,3} Prodi Teknik Informatika, Fakultas Sains dan Teknologi,
Universitas Islam Negeri Syarif Hidayatullah Jakarta

ABSTRAK

Seiring perkembangan zaman, komputer semakin dibutuhkan dalam kehidupan sehari-hari, baik pada bidang akademis maupun bidang non-akademis. Komunikasi bisa juga mempengaruhi proses informasi itu bisa disampaikan dengan baik dan tepat. Komunikasi yang dilakukan dengan melakukan proses pengiriman data dari komputer ke komputer lain dalam suatu jaringan. Dan pada saat ini sistem pengiriman data masih kurang maksimal sehingga sering terjadi kesalahan dalam proses pengirimannya. Sebagian besar sistem pengiriman data sekarang ini belum dapat mengurangi kesalahan pada pengiriman, dan juga masih kurangnya pengetahuan tentang bagaimana proses pengiriman dan pengkoreksian error. Pada penelitian ini akan dirancang sebuah aplikasi simulasi yang bertujuan untuk menggambarkan bagaimana proses pengkoreksian error pada proses pengiriman data yang berupa angka dalam bentuk integer dan akan diubah menjadi biner untuk memudahkan dalam proses pengecekan error. Metode yang digunakan dalam perancangan simulasi ini adalah menggunakan Hamming Code. Perancangan simulasi menggunakan Java. Dari hasil pengamatan aplikasi ini data dikirimkan akan di deteksi jika terjadi kesalahan maka aplikasi akan mengkoreksi kesalahan yang telah terdeteksi. Hasil pengamatan menunjukkan bahawa 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.

I. PENDAHULUAN

A. Latar Belakang

Pada era teknologi saat ini, komunikasi data menjadi aspek yang sangat penting dikarenakan penggunaan teknologi khususnya komputer dalam setiap aspek kehidupan. Komunikasi data dibutuhkan sebagai suatu cara untuk menyampaikan atau menyebarkan data, informasi, berita, ataupun pendapat dengan menggunakan media transmisi data. Pentingnya komunikasi data terletak pada kelancaran penyampaian dan penyebaran data ini akan membawa dampak kepada kelancaran suatu proses bisnis yang bergantung pada data dan informasi.

Pengetahuan tentang adanya metode pengkoreksian error juga sangatlah penting. Ada beberapa metode yang digunakan untuk mendeteksi dan mengkoreksi *error* yang terjadi pada komunikasi data. Salah satunya metode yang digunakan untuk mengecek *error* dan mengkoreksi error adalah metode Hamming Code. Metode ini merupakan salah satu jenis linier *error correction code* yang sederhana. Memiliki keuntungan yaitu cara kerjanya yang cukup sederhana dan tidak membutuhkan alokasi memori yang banyak. Selain itu dengan digunakannya konsep *error correcting code* pada metode ini maka jika ditemukan *error* pada saat pendeteksian, data tidak perlu ditransmisikan ulang tetapi langsung dikoreksi di *receiver*.

Menurut quisoner yang sudah disebar dikalangan mahasiswa Teknik Informatika UIN semester 4 keatas dengan kriteria tertentu, hampir 95% dari total mahasiswa yang tidak mengetahui tentang pengkoreksian error dengan metode Hamming tetapi hampir 18% mahasiswa UIN mengetahui pengkoreksian error selain metode Hamming.

Dengan metode Hamming yang disimulasikan dapat memberi pengetahuan tentang pengkoreksian *error* dengan metode Hamming dan mahasiswa bisa melihat bagaimana proses pada *Binary Symetric Channel* itu bekerja pada pengkoreksian Hamming yang dibangun. Untuk BER (*Bit Error Rate*) mahasiswa bisa mengetahui apakah dengan menggunakan Hamming ini bisa menurunkan kemungkinan error yang terjadi.

Oleh karena itu penulis bertujuan mengambil konsep dan mengimplementasikan dalam bentuk simulasi tentang Hamming Code ini untuk memberi pengetahuan tentang bagaimana proses pendeteksian *error* dan cara pengkoreksian error pada komunikasi data pada bahasa pemrograman berbasis Java.

B. Pokok Permasalahan

1.1. Rumusan Masalah

Dari latar belakang yang telah penulis paparkan diatas, maka penulis dapat merumuskan beberapa permasalahan antara lain, sebagai berikut :

1. Bagaimana pembuatan dan penerapan simulasi pengkodean Hamming dengan menggunakan *Binary Symetric Channel (BSC)*?
2. Dapat atau tidaknya membantu mahasiswa dalam mengetahui bagaimana proses dan pengkoreksian *error* pada pengkodean Hamming?

1.2. Batasan Masalah

Untuk menghindari pembahasan menjadi terlalu luas maka penulis perlu membatasinya. Adapun batasan masalah dalam penelitian ini adalah, sebagai berikut:

1. Hanya membahas tentang Kode Hamming (7,4)
2. Data dan noise dibangkitkan dengan menggunakan Bilangan Acak.
3. Program dibuat dengan bantuan Netbeans IDE 7.0.
4. Noise kanal dimodelkan dengan probabilitas *error* kanal yang diinput oleh si pengguna simulasi.
5. Tidak membahas tentang teknik modulasi dan filter.
6. Hanya membahas tentang BER pada Pengkodean Hamming dengan simulasi.
7. Tidak membahas tentang Analisa *bit error rate* Pengkodean Hamming secara sistematis.
8. Tidak membahas lebih mendalam tentang source code yang dibuat melainkan hanya sebatas penggunaan algoritma yang diterapkan pada simulasi yang dibuat.
9. Tidak membahas tentang include atau tidak terhadap jaringan hanya simulasi saja.
10. FEC dan BEC hanya digunakan untuk mempermudah dalam penggunaan dan penerapan algoritma Pengkodean Hamming.
11. Simulasi yang penulis buat ini berfokus pada Mahasiswa pada tingkat empat keatas dan yang sudah mengambil mata kuliah komunikasi digital, dan jika Dosen bersedia menggunakan *tools* yg di buat oleh penulis ini hanya sekedar jadi bahan ajar untuk Mahasiswa.

C. Tujuan Penelitian

Adapun tujuan dari skripsi ini adalah, sebagai berikut :

1. Membuat suatu program simulasi yang mampu mensimulasikan proses pengkodean Hamming sebagai bukti fisik dari penerapan ilmu yang dipelajari selama berada di perguruan tinggi.
2. Mencari *Bit Error Rate* pada pengkodean Hamming.
3. Menggunakan simulasi untuk menganalisa *Bit Error Rate* pada pengkodean Hamming.
4. Mengetahui logika pada proses perancangan simulasi pengkodean Hamming.
5. Memberi pengetahuan kepada masyarakat (mahasiswa) sebagai pengguna komunikasi digital khususnya pengguna komputer bagaimana proses

pengecekan atau pengkoreksian *error* pada komunikasi digital menggunakan pengkodean Hamming.

D. Metodologi Penelitian

1. Metode Pengumpulan Data

1) Studi Pustaka

Penulis melakukan studi kepustakaan melalui literatur-literatur atau referensi-referensi yang ada di Perpustakaan Fakultas Sains dan Teknologi maupun di Perpustakaan Utama UIN Syarif Hidayatullah Jakarta dan Perpustakaan pada Universitas lain, serta literatur dari internet untuk membantu penyusunan penelitian ini

2) Studi Lapangan

Pada studi lapangan penulis melakukan observasi dan kuisioner tentang seberapa besar mahasiswa mengetahui pengkodean Hamming dan bisa tidak memahami pengkodean Hamming. Serta memahami tentang pengkodean Hamming yang pernah penulis lain buat. Dan tidak lupa melakukan perhitungan setelah melakukan kuisioner sehingga dapat menjadi tolak ukur diperlukannya pembuatan simulasi ini.

2. Metode Pengembangan Simulasi

Metode pengembangan sistem yang penulis lakukan adalah dengan menggunakan metode *Prototyping*. Alasan menggunakan metode *prototyping* adalah karena metode ini dapat bekerja lebih baik dalam menentukan pengembangan yang dibutuhkan, penerapan pada simulasi ini lebih mudah dilakukan karena mengetahui apa saja yang dibutuhkan dan lebih menghemat waktu dalam pengembangan sistem. Berikut langkah-langkah dalam metode *Prototyping* :

1. Pengumpulan Kebutuhan

Pada tahapan ini akan dilakukan analisa simulasi yang pernah penulis lain buat, analisa data, spesifikasi aplikasi, dan kebutuhan-kebutuhan lainnya.

2. Membangun *Prototyping*

Melakukan perancangan proses yang didalamnya terdapat proses-proses yang kemungkinan akan terjadi di aplikasi, dan melakukan perancangan interface.

3. Evaluasi *Prototyping*

Evaluasi ini dilakukan oleh *pemakai* apakah *Prototyping* yang sudah dibangun sudah sesuai dengan keinginan *pemakai*.

4. Mengkodekan *System*

Pada tahapan ini system atau simulasi yang akan penulis buat sudah disepakati dan diterjemahkan dalam bahasa pemrograman yang sesuai atau

dengan kata lain sudah masuk dalam tahapan Konstruksi atau pembangunan simulasi atau *system*. Dan dibantu dengan menggunakan *tools* bantuan.

5. Menguji *System*
Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus diuji dahulu sebelum digunakan.
6. Evaluasi Aplikasi
Setelah semua rancangan sudah menjadi suatu aplikasi yang siap pakai. Kemudian Mengevaluasi dari hasil pengujian apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika sudah, maka langkah ke-enam dilakukan, jika belum maka mengulangi langkah 4 dan 5
7. Menggunakan System
Perangkat lunak yang telah diuji dan siap untuk digunakan.

II. LANDASAN TEORI

A. Pengertian Perancangan

Menurut Lonnie D. Bentley dan Jeffrey L. Whitten (2007,p160), perancangan adalah suatu teknik menggabungkan kembali bagian-bagian informasi yang telah dipisahkan oleh analisis sistem.

Perancangan adalah penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi Perancangan sistem dapat dirancang dalam bentuk bagan alir sistem (*system flowchart*), yang merupakan alat bentuk grafik yang dapat digunakan untuk menunjukkan urutan-urutan proses dari sistem (Syifaun Nafisah, 2003 : 2).

Pengertian Simulasi

Menurut Law dan Kelton(1991), Simulasi merupakan suatu teknik meniru operasi-operasi atau proses-proses yang terjadi dalam suatu sistem dengan bantuan perangkat komputer dan dilandasi oleh beberapa asumsi tertentu sehingga sistem tersebut bisa dipelajari secara ilmiah.

Menurut Pusat Bahasa Depdiknas (2005), simulasi adalah suatu metode pelatihan yang memperagakan sesuatu dalam bentuk tiruan yang mirip dengan keadaan yang sesungguhnya.

Menurut Udin Syaefudin Sa'ud(2005: 129), simulasi adalah sebuah replikasi atau visualisasi dari perilaku sebuah sistem, misalnya sebuah perencanaan pendidikan, yang berjalan pada kurun waktu yang tertentu. Jadi dapat dikatakan bahwa simulasi itu adalah sebuah model yang berisi perangkat memungkinkan keputusan-

keputusan yang menentukan bagaimana ciri-ciri utama itu bisa dimodifikasi secara nyata.

Definisi Pengkodean

Pengkodean adalah salah satu alasan mengapa kita menyatakan suatu data (atribut) dalam bentuk lain adalah untuk efisiensi ruang penyimpanan, menurut Fathansyah dalam bukunya yang berjudul Basis Data(2007:105). Dan cara yang ditempuh untuk menyatakan suatu data dalam bentuk lain itu adalah melalui pengkodean (data coding).

Definisi *Binari Symmetric Channel*

Binari Symmetric Channel (atau BSC) adalah model saluran komunikasi yang umum digunakan dalam pengkodean teori dan teori informasi. Dalam model ini, *transmitter* ingin mengirim sedikit (nol atau satu), dan *receiver* menerima sedikit. Hal ini diasumsikan bahwa bit biasanya transmisikan dengan benar, tetapi itu akan "membalik" dengan probabilitas kecil ("probabilitas pindah silang"). Saluran ini sering digunakan dalam teori informasi karena merupakan salah satu saluran yang paling sederhana untuk menganalisis.

Definisi *Bit Error Rate (BER)*

Bit Error Rate atau BER merupakan sejumlah bit digital bernilai tinggi pada jaringan transmisi yang ditafsirkan sebagai keadaan rendah atau sebaliknya, kemudian dibagi dengan sejumlah bit yang diterima atau dikirim atau diproses selama beberapa periode yang telah diterapkan.

Menurut Budiharja Kusuma (2012), Bit Error Rate adalah jumlah kesalahan bit yang dibagi dengan jumlah bit yang ditransfer selama interval waktu belajar. Ber mengukur kinerja *unitless*, sering dinyatakan sebagai nomor persentase

Sistem Komunikasi Digital

Dalam sistem telekomunikasi terdapat dua jenis sistem komunikasi yaitu sistem komunikasi analog dan sistem komunikasi digital. Perbedaan keduanya adalah pada sinyal yang digunakan untuk melakukan komunikasi. Pada sistem komunikasi analog, sinyal yang dikirim berupa sinyal yang bervariasi dan tidak tetap, sedangkan pada sistem komunikasi digital, sinyal yang di kirimkan tertentu yang sudah tetap bentuknya.

Pada sistem telekomunikasi, meskipun sistem komunikasi analog masih digunakan, akan tetapi perkembangan dan penggunaan sistem komunikasi digital lebih banyak bila dibandingkan dengan sistem komunikasi digital.

Hal ini karena sistem komunikasi digital mempunyai beberapa keuntungan sebagai berikut:

1. Sinyal digital lebih mudah untuk diregenerasikan menjadi bentuknya yang semula dibandingkan dengan sinyal analog.
2. Rangkaian digital memiliki tingkat distorsi dan interferensi yang lebih rendah dibandingkan dengan rangkaian analog.
3. Sistem komunikasi digital memiliki teknik kesalahan dan koreksi, sehingga dapat mengurangi tingkat kesalahan, sedangkan pada sistem komunikasi analog tidak.
4. Biaya produksi dari rangkaian digital lebih rendah bila dibandingkan dengan rangkaian analog.
5. Perangkat dari sistem komunikasi digital lebih mudah untuk dikombinasikan dengan perangkat lainnya bila dibandingkan dengan perangkat dari sistem analog.

Deteksi dan Koreksi Kesalahan

Selama pengiriman data baik berupa sinyal digital maupun sinyal analog, data tersebut mengalami perubahan dan kesalahan. Salah satu tujuan dari komunikasi data adalah untuk mengirim data secara utuh dari sumber data hingga sampai ke penerima atau tujuan dari pengirim data tersebut.

Ada beberapa kemungkinan terjadinya kesalahan dalam pengiriman frame-frame data, yaitu:

- P_0 = probabilitas kesalahan bit tunggal yang disebut dengan bit error rate
- P_1 = probabilitas frame yang diterima tanpa adanya kesalahan
- P_2 = probabilitas frame diterima dengan kesalahan tetapi tidak terdeteksi
- P_4 = probabilitas frame diterima dengan kesalahan tetapi terdeteksi

Konsep Dasar Sistem Pengkodean

Kesalahan (error) merupakan masalah dalam sistem komunikasi, sebab dapat mengurangi kinerja dari sistem. Untuk mengatasi masalah tersebut diperlukan suatu sistem yang dapat mengoreksi error. Sehingga dicari solusi metode penanganan error dengan memeriksa bit. Beberapa contoh penanganan error dengan memeriksa bit, seperti (Irsan, 2009):

1. Backward Error Control

Pada backward error control, apabila pada data yang diterima terjadi error, maka penerima akan mengirimkan sinyal kepada pengirim untuk melakukan pengiriman ulang.

2. Forward Error Control

Pada Forward Error Control, sebelum data dikirimkan data akan dikodekan dengan suatu pembangkit kode (enkoder), dan kemudian dikirimkan ke penerima. Pada penerima akan terdapat sebuah penerjemah kode (dekoder) yang mengkodekan data tersebut, dan apabila terjadi error pada data akan dilakukan pengkoreksian data.

Menurut Andi dan Rum Andi, Kontrol kesalahan arus balik ini dibagi menjadi 2 bagian, yaitu:

1. Teknik yang digunakan untuk deteksi kesalahan
2. Kontrol algoritma yang telah disediakan untuk mengontrol transmisi ulang.

Kode Hamming (Enkripsi)

Kode hamming merupakan kode non-trivial untuk koreksi kesalahan yang pertama kali diperkenalkan. Kode ini dan variasinya telah lama digunakan untuk kontrol kesalahan pada sistem komunikasi digital. Kode hamming terbagi menjadi 2 macam, yaitu biner dan non-biner. Kode hamming biner dapat direpresentasikan dalam bentuk persamaan sebagai berikut:

$$(n, k) = (2^m - 1, 2^m - 1 - m) \quad (1.7)$$

Di mana k adalah jumlah bit informasi yang membentuk n bit kata sandi, dan m adalah bilangan bulat positif. Jumlah paritas bitnya adalah sejumlah $m = n - k$ bit.

Bit stream dari sumber data yang masuk ke enkoder dikodekan dengan menggunakan suatu generator. Oleh karena itu dalam proses pengkodean Kode Hamming diperlukan suatu generator matriks.

Kode Hamming diperoleh dari hasil perkalian antara bit stream dengan generator matriks kode Hamming. Generator matriks kode Hamming yang dipilih adalah generator Kode Hamming yang sistematis. Kode hamming ini disimpan dalam array 2 dimensi. Sebagai contoh kode hamming(7,4) yang mengkodekan 4 bit stream menjadi 7 bit kode yang akan dikirimkan yaitu sbb:

$$G = \begin{bmatrix} 0111000 \\ 1010100 \\ 1100010 \\ 1110001 \end{bmatrix}$$

Pengkodean Hamming (Dekripsi)

Pengkodean Kode Hamming dilakukan dengan cara menghitung sindrom yang dihasilkan dengan cara mengalikan bit Kode Hamming yang

diterima dengan matriks cek pariti yang disesuaikan dengan generator kode Hamming yang digunakan pada sisi penerima (Irsan;2009). Sebagai contoh, matriks cek pariti yang sesuai dengan contoh generator matriks untuk Kode Hamming (7,4) diatas adalah sbb:

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

Matriks untuk mengecek pariti ini dapat diperoleh dari rumus sebagai berikut :

$$H = [I | P^T] \dots\dots\dots (3.2)$$

Dimana H merupakan matriks cek pariti, I adalah matriks Identitas, dan P^T merupakan hasil dari transpose dari pariti P

Dari matriks Pariti cek diatas dapat dihitung sindrom dengan rumus sebagai berikut :

$$S = r \cdot H^T \dots\dots\dots$$

(3.3)

Dimana :

- S = Sindrom
- R = bit Kode Hamming yang diterima
- H^T = transposisi dari matriks cek parity

Setelah didapat sindromnya maka dapat diketahui apakah kose yang diterima ada error atau tidak dan dimana letak error-nya bila ada. Jika sindrom yang dihasilkan adalah 0, maka berarti tidak terjadi error selain itu berarti ada terjadi error. Untuk mengetahui letak error-nya, maka sindrom yang sedah diperoleh harus disesuaikan dengan matriks H^T , berarti pada posisi tersebut telah terjadi error. Lalu ubahlah posisi yang error tersebut dengan menginvertkan kode yang diterima. Kemudian ambillah 4 bit kode yang terakhir sebagai bit stream data.

Operasi XOR

Operasi XOR merupakan kombinasi dari operasi-operasi dasar logika dasar. Operasi XOR mempunyai dua input dan satu output. Operasi XOR dari dua input A dan B hanya akan bernilai logika “0” apabila keadaan kedua input A dan B bernilai logika sama, yaitu “0” atau “1”. Atau dengan kata lain output dari operasi XOR akan memiliki nilai logika “1” apabila kedua inputnya bernilai logika berbeda. Dalam sistem digital operasi logika XOR dan input A dan B akan dilambangkan dengan $A \oplus B$. Operasi XOR dapat dinyatakan seperti berikut,

$$A \oplus B = AB +$$

Tabel kebenaran dari operasi XOR dapat dinyatakan sebagai berikut,

Tabel 2.1 Tabel kebenaran fungsi XOR

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Keterangan

- A, B = input operasi logika XOR
- $A \oplus B$ = output operasi logika XOR
- 1 = nilai logika bernilai benar (*true*)
- 0 = nilai logika bernilai salah (*false*)

Operasi AND

Operasi AND adalah salah satu dari kombinasi logika-logika dasar. Operasi AND memiliki dua input dan satu output. Operasi AND akan bernilai output “0” apabila jika kedua input A dan B logika inputnya berbeda. Atau sebaliknya operasi AND akan bernilai “1” jika kedua inputnya mempunyai nilai logika yang sama yaitu “0” dan “1”. Dalam sistem digital operasi logika AND dan input A dan B akan dilambangkan dengan $A \wedge B$. Seperti yang terlihat pada tabel 2.2 adalah tabel kebenaran untuk logika AND.

Tabel 2.2 Tabel Kebenaran Untuk Logik Penghitungan AND

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

III. METODOLOGI PENELITIAN

A. Metode Pengumpulan Data

Pengumpulan data merupakan langkah yang penting untuk metode ilmiah, karena pada umumnya data yang dikumpulkan digunakan untuk menguji hipotesis yang telah dirumuskan. Teknik pengumpulan data yang sering digunakan antara lain:

Studi Pustaka

Studi pustaka yang dilakukan oleh penulis dengan mencari dan mempelajari teori-teori yang dapat dijadikan landasan atau kerangka berpikir bagi penelitian yang akan dilakukan sehingga penyusunan laporan skripsi ini dapat lebih terarah. Teori-teori tersebut diperoleh dari literatu

tertulis, dimana literatur tersebut perlu dicari sebanyak mungkin agar dapat dibandingkan satu dengan yang lainnya.

Studi Lapangan

Untuk studi lapangan ini penulis menggunakan metode observasi dan kuisioner, yaitu dengan melakukan pengamatan dan pencatatan secara sistematis fenomena-fenomena yang akan diteliti. Hal ini bertujuan untuk memperoleh penjelasan-penjelasan langsung mengenai data-data dan informasi yang dibutuhkan dalam percobaan yang akan dilakukan pada penelitian ini. Dalam penelitian ini, observasi yang dilakukan adalah mencari file-file yang akan dibutuhkan dari internet untuk melakukan pengujian diantaranya susunan bit data yang harus diuji serta bagaimana logika algoritma yang dipakai itu bekerja. Untuk penjelasan lebih lengkap dilihat pada sub bab empat.

Kuisioner yang dilakukan untuk mengetahui seberapa besar pengetahuan mahasiswa tentang pengoreksian *error* dan metode-metode apa saja yang ada. Dan 97,5% dari mahasiswa yang penulis berikan kuisioner tidak mengetahui bagaimana pengoreksian tentang menggunakan pengkodean hamming, dan 17,5% mahasiswa mengetahui pengoreksian *error* tetapi tidak menggunakan metode hamming ini. Penulis melakukan kuisioner ini dengan mengajukan ke 50 mahasiswa yang mempunyai tingkatan semester yaitu semester 4 keatas dan tidak sedikit pula lulusan dari Teknik Informatika.

B. Metode Pengembangan Simulasi

Pada metode pengembangan simulasi ini penulis menggunakan metode prototype. Menurut Raymond McLeod, prototype didefinisikan sebagai alat yang memberikan ide bagi pembuat maupun pemakai potensial tentang cara sistem berfungsi dalam bentuk lengkapnya, dan proses untuk menghasilkan sebuaha prototype disebut prototyping. Adapun alasan kenapa penulis menggunakan metode pengembangan prototype ini, sebagai berikut:

1. Perubahan fungsional sering berubah, maksudnya disini kebutuhan akan simulasi yang sering berubah-ubah sesuai dengan keinginan pengguna dan fungsi-fungsi yang ada di dalam simulasi ataupun interface yang telah dibuat.
2. User atau pengguna tidak telalu memiliki pengetahuan yang memadai, maksudnya disini adalah fokus user yang dilakukan penulis adalah para masyarakat yang hanya

menggunakan komunikasi digital ini tanpa dia tau bagaimana proses pengoreksian dan penanganan *error* itu bisa terjadi.

3. Mengijinkan ada banyak inovasi, yang dimaksud adalah setelah menganalisa program simulasi yang telah dibuat oleh penulis lain, penulis tidak hanya berpatokan pada simulasi yang pernah dibuat tetapi menambahkan ataupun menghilangkan fungsi-fungsi yang menurut penulis tidak terlalu penting.
4. Membutuhkan banyak input dari user, yang dimaksud adalah masukan-masukan dari dosen dan teman-teman penulis untuk pembuatan simulasi yang diinginkan baik oleh penulis maupun masyarakat yang menggunakan komunikasi digital ini.
5. Ada tekanan untuk implementasi secepatnya, disini karena penulis menginginkan program simulasi ini bisa jadi secepatnya sehingga tidak terlalu menghabiskan banyak dana dan waktu.

Adapun langkah-langkah yang digunakan dalam pengembangan simulasi yang akan dibuat, sebagai berikut:

Pengumpulan Kebutuhan

Analisis Simulasi Yang Pernah Dibuat

Pada simulasi yang pernah dibuat penulis menganalisa pada bagian fitur-fitur yang ada di simulasi tersebut, serta menganalisa alur dari programnya dan menganalisa cara kerja dari algoritma dari pengkodean Hamming tersebut.

Analisis Data

Data yang digunakan adalah data dalam bentuk random dikarenakan pada Pengkodean Hamming tersebut awal pengiriman akan dilakukan penrandoman data terlebih dahulu lalu baru dibuat menjadi data dalam bentuk biner kemudian dikirimkan. Dalam pembuatan simulasi pada pengolahan data berpatokan pada kondisi real untuk Pengkodean Hamming. Setelah diterapkan pada kondisi real selanjutnya data tersebut dilakukan pengoreksian dan penanganan *error* serta pencatatan waktu pengiriman pada jumlah pengiriman yang telah ditentukan.

Spesifikasi Aplikasi

Pada tahapan ini penulis membuat spesifikasi dari aplikasi simulasi yang penulis buat dari kebutuhan hardware yang penulis gunakan sampai kebutuhan dari software yang penulis gunakan. Pembuatan spesifikasi ini

bertujuan untuk memudahkan dalam melakukan analisa kemampuan yang sesuai dengan kebutuhan.

Setelah konstruksi telah dibuat sesuai dengan yang diharapkan maka sistem siap untuk digunakan.

IV. PERANCANGAN APLIKASI

Pada perancangan aplikasi simulasi ini penulis melakukan perancangan dengan menggunakan metode pengembangan simulasi Prototyping. *Prototyping* adalah proses pembuatan model sederhana software yang mengijinkan pengguna memiliki gambaran dasar tentang program serta melakukan pengujian awal. *Prototyping* memberikan fasilitas bagi pengembang dan pemakai untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan perangkat lunak yang akan dibuat. *Prototyping* merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan. Berikut ini adalah langkah-langkah yang penulis lakukan dalam melakukan prancangan dengan menggunakan metode *prototyping*, sebagai berikut :

Pengumpulan Kebutuhan

Pada tahapan pengumpulan kebutuhan penulis melakukan dengan beberapa tahapan seperti berikut ini:

Analisis Simulasi Yang Pernah Dibuat

Pada bagian ini penulis melakukan analisa pada aplikasi simulasi yang pernah dibuat penulis lain untuk mengetahui bagaimana penerapan algoritma pengkodean Hamming dalam aplikasi simulasi tersebut, selain itu juga analisis fitur-fiturnya dan apa saja yang kurang dan akan ditambahkan dan apa saja yang bisa dihilangkan. Berikut ini adalah fitur-fitur yang penulis analisa dan yang akan penulis buat yang dapat dilihat pada **tabel 4.1** berikut ini:

Tabel 4.1. Tabel Perbandingan Analisa dan Simulasi

Yang Dianalisa	Aplikasi yang pernah dibuat	Aplikasi yang akan dibuat
Fitur-fitur dan algoritma yang dianalisa	<ol style="list-style-type: none"> 1. Terdapat pengatur kecepatan untuk simulasi. 2. Terdapat tiga tombol untuk menjalankan simulasi yaitu simulasi, hentikan, dan keluar. 3. Alur algoritma sumber data (<i>random</i>), pembangkit bilangan acak, pembangkit <i>noise</i>, <i>output</i>, pengecekan <i>bit error rate</i>. 4. Untuk output dapat dilihat dengan menghentikan dan menjalankannya kembali. 5. Menggunakan bahasa pemrograman VB (<i>Visual Basic</i>). 	<ol style="list-style-type: none"> 1. Terdapat pengatur kecepatan dengan tiga kecepatan yaitu <i>low</i>, normal, dan <i>fast</i>. 2. Hanya terdapat dua tombol untuk menjalankan simulasi dan meresetnya. 3. Untuk alur algoritma sama dengan aplikasi yang pernah dibuat tetapi ditambahkan untuk melakukan pengecekan <i>bit error</i> dan letak <i>error</i> nya pada posisi berapa. 4. Output supaya lebih dapat dibaca dengan setiap simulasi akan tertulis pada tabel yang sudah dibuat. 5. Menggunakan bahasa pemrograman Java berbasis dekstop.

Analisa Data

Yang dimaksud dengan analisa data disini adalah data yang akan dikoreksi sudah terinclude pada program simulasi. Data yang akan dianalisis berupa angka yang akan di *random* pada simulasi yang dibuat dan akan dirubah menjadi biner supaya terlihat seperti pada bentuk *real* atau sesuai dengan kenyataan yang ada.

Spesifikasi Aplikasi

Berikut ini adalah spesifikasi dari perangkat keras atau hardware dan perangkat lunak atau software, sebagai berikut :

Kebutuhan Perangkat Keras dan Perangkat Lunak

Perangkat Keras yang digunakan adalah sebagai berikut:

Tabel 4.2. Tabel Spesifikasi Kebutuhan Perangkat Keras

Jenis Laptop atau Komputer	Spesifikasi
Laptop ACER ASPIRE 4745G	Intel i5-450M(2,4GHz, 3MB L3 cache), 2GB RAM, 640 GB HDD, ATI Mobility Radeon 5470, keyboard, Mouse USB.

Kebutuhan untuk perangkat lunak yang penulis gunakan adalah seperti pada tabel 4.3.

Tabel 4.3. Perangkat Lunak Yang Digunakan

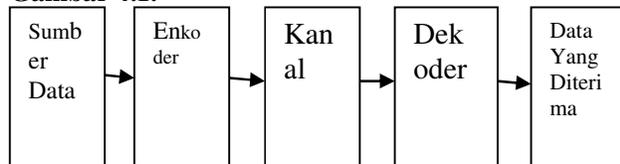
Nama Perangkat Lunak	Fungsi
Sistem Operasi Windows Home Premium 64bit	Sebagai <i>interface</i> antara pengguna dan komputer.
Java 2 SDK 1.6	Sebagai alat pembantu dalam pengekseskuan program yang berbasis java
Netbeans IDE 7.0	Sebagai <i>tools</i> untuk membuat serta menjalankan program simulasi yang dibuat.

4.1. Membangun Prototyping

Pada tahapan ini akan dilakukan perancangan prototyping dengan beberapa tahapan yaitu tahapan rancangan proses dan rancangan interface. Berikut ini adalah penjelasan tentang rancangan yang penulis lakukan, sebagai berikut :

4.1.1. Rancangan Proses

Pada tahapan ini akan dilakukan rancangan proses yang akan terjadi pada aplikasi yang akan dibuat. Perancangan proses dibuat menjadi flowchart agar mudah dimengerti alur prosesnya. Berikut ini adalah rancangan umum serta penjelasan sederhana untuk mempermudah dalam menganalisa dan membangun aplikasi yang akan dibuat, seperti pada Gambar 4.1.



Gambar 4.1 Model Perancangan Umum Simulasi

Dari Gambar 4.1 dapat dilihat ada beberapa proses yang dilakukan dalam proses Hamming yang berjalan pada kondisi real atau sesuai dengan kenyataannya. Dan berikut ini adalah gambaran semua proses yang telah penulis rancang seperti pada gambar 4.2 berikut ini:

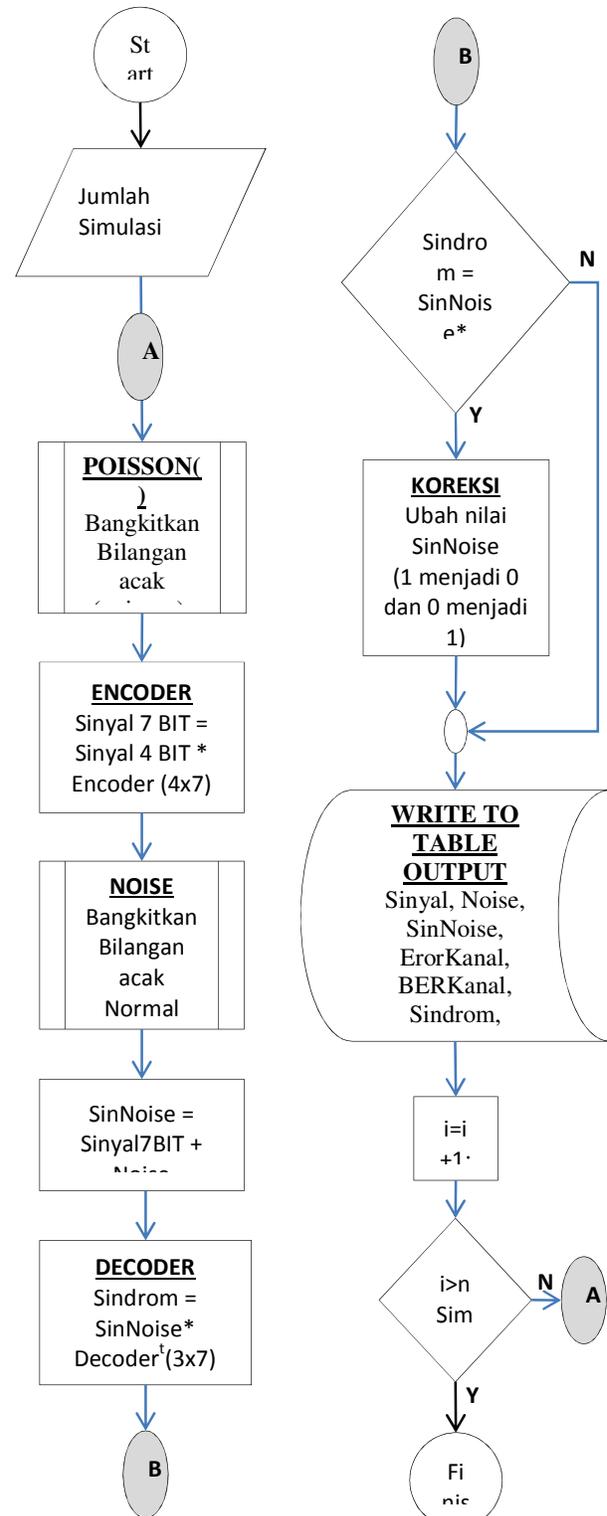
Dari rancangan tersebut dan mengetahui bagaimana proses atau alur kerja dari pengkodean Hamming tersebut maka akan dilakukan penerapan pengkodean java yang mengacu pada rancangan tersebut, berikut ini adalah rancangan yang penulis lakukan berdasarkan Gambar 4.1 dan Gambar 4.2.

1. Sumber Data

Sumber data dimodelkan dalam bentuk bilangan acak dan akan dibangkitkan menurut metode pembangkitan bilangan acak, kemudian diubah dalam bentuk biner (4 Bit), dengan asumsi data berupa sinyal digital yang

siap dienkoderkan. Sinyal digital yang dihasilkan berupa 4 bit stream data.

Bilangan acak yang akan dibangkitkan dengan menggunakan algoritma pengkodean Hamming. Sehingga bilangan acak yang dihasilkan memiliki rata-rata nilai 8.



Gambar 4.2. Alur Perancangan Pengkodean Hamming

2. Enkoder

Pada bagian ini sinyal digital yang dibandingkan akan diencode dengan matriks generator Kode Hamming (7,4) sebagai berikut:

$$G = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Bit-bit data digital dalam bentuk biner yang sudah dibangkitkan pada bagian sumber data akan dibentuk menjadi matriks 1x4. Matriks tersebut akan dikalikan dengan matriks generator yang telah tersedia dan dihasilkan matriks 1x7, sehingga dihasilkan bit-bit data yang telah terencode sebanyak 7 bit.

3. Kanal

Pada bagian ini akan dibangkitkan dengan menggunakan pembangkit bilangan acak Normal. Kemudian bilangan acak Normal ini akan disubstitusikan ke dalam persamaan (3,2) untuk menghasilkan probabilitasnya. Bilangan acak ini akan dibangkitkan sebanyak 7 buah, sesuai dengan jumlah bit dari sinyal yang dikirimkan.

Setiap bilangan acak yang telah dibangkitkan tersebut akan dibandingkan dengan probabilitas error kanal. Jika bilangan acak tersebut lebih kecil dari probabilitas error kanal, maka akan dibangkitkan bit noise 1, apabila bilangan acak tersebut lebih besar dari probabilitas error kanal, maka akan dibangkitkan bit noise 0. Sehingga terbentuk deretan noise sebanyak 7 digit dalam bentuk biner.

Setelah itu sinyal yang sudah diencode tersebut akan ditambahkan dengan noise menurut aturan gerbang XOR sebagai berikut:

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

Apabila pada noise yang dibangkitkan terdapat bit bernilai 1, maka sinyal yang terencode akan berubah nilainya akibat penjumlahan dari sinyal tersebut dengan noise.

4. Dekoder

Pada bagian ini sinyal yang akan diterima akan dihitung sindromnya dan dilakukan pengecekan dan pengkoreksian kesalahan bila ada. Perhitungan Sindrom dilakukan dengan menggunakan rumus $S = r \cdot H^T$, dimana r merupakan sinyal data yang diterima dan H^T adalah transpose dari matriks pengecek error. Matriks H yang digunakan adalah matriks H untuk Kode Hamming (7,4) yaitu :

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

Sinyal yang berasal dari kanal yang sudah ditambah dengan noise yang terdiri dari 14 bit akan dikelompokkan menjadi 2 bagian, lalu disimpan kedalam 2 buah matriks yang berukuran 1x7. Kemudian masing-masing matriks tersebut akan dihitung sindromnya menggunakan rumus untuk menghitung sindrom yang sudah dibahas pada BAB II. Apabila sindromnya tidak sama dengan 0, maka telah terjadi error, jika sama dengan 0 maka tidak terjadi error.

Bila terjadi error, maka dilakukan pencocokan sindrom dengan matriks H^T , dimana posisi data pada matriks H^T yang sama dengan sindrom adalah lokasi kesalahan, kemudian dilakukan koreksi dengan menginversikan bit data, yaitu data yang bernilai 0 menjadi 1 dan sebaliknya data yang bernilai 1 menjadi 0. Setelah selesai dilakukan pengkoreksian, diambil 4 bit terakhir dari bit data sebagai data informasi yang dikirimkan, lalu kedua data tersebut digabungkan dan dihasilkan bit stream data yang terdecode.

5. Data yang diterima

Data yang diterima ini akan dibandingkan dengan sumber data, apabila terdapat bit yang berbeda, maka jumlah error akan bertambah. Setelah semua bit dibandingkan, maka akan dihitung bit error rate (BER) dengan membagikan jumlah error dengan jumlah bit yang dikirimkan seluruhnya

$$BER = \frac{\text{Jumlah Error}}{\text{Jumlah bit yang dikirimkan}}$$

Setelah mengetahui penjelasan diatas maka penulis dapat menentukan apa saja yang harus dibangun sehingga dapat membuat flowchart sesuai dengan apa yang penulis buat, pertama yang penulis buat adalah proses keseluruhan program yang akan dibuat dan bagaimana penerapan algoritma pengkodean hamming ini bekerja.

4.1.2. Pengujian Interface Aplikasi Simulasi

Pengujian fungsi interface aplikasi simulasi ini bertujuan untuk mengetahui fungsionalitas dari elemen-elemen interface yang terdapat di dalam aplikasi yang dibuat. Elemen-elemen interface yang diujikan terutama adalah elemen button.

4.2. Evaluasi Sistem

Pada tahapan evaluasi sistem ini menjelaskan apakah sistem yang sudah dibuat ini sesuai dengan apa yang diharapkan atau tidak. Jika sudah atau sesuai dengan yang diharapkan maka langkah

selanjutnya adalah menggunakan sistem atau implementasi sistem. Jika tidak maka akan dilakukan lagi pengkodean dan pengujian kembali sampai sesuai dengan apa yang diharapkan.

4.3. Menggunakan System

Pada tahapan ini adalah tahapan terakhir yaitu Implementasi. Penulis melakukan implementasi dengan 2 tahap yaitu implementasi pada desain interface dan Aplikasi. Berikut ini adalah hasil dari implementasi tersebut :

4.3.1. Implementasi Desain Interface

Hasil dari rancangan yang sudah dilakukan maka dibuat interface untuk mempermudah pembacaan dari aplikasi simulasi ini. Desain interface yang sesuai dengan langkah pertama dapat dilihat pada lampiran dengan *point* ke 2.

4.3.2. Implementasi Aplikasi

Hasil dari rancangan yang dibuat sudah diterapkan pada bahasa java dan berjalan sesuai dengan apa yang diinginkan. Untuk melihat source code secara penuh dapat dilihat pada lampiran *point* 3. Contoh penghitungan simulasi

1. Penghitungan matriks generator menggunakan XOR dan AND

$$= [1010] \begin{bmatrix} 0111000 \\ 1010100 \\ 1100010 \\ 1110001 \end{bmatrix}$$

$$\text{Untuk bit 1 : } (1x0) \oplus (0x1) \oplus (1x1) \oplus (0x1) = 0 \oplus 0 \oplus 1 \oplus 0 = 0 \oplus 1 = 1$$

$$\text{Untuk bit 2 : } (1x1) \oplus (0x0) \oplus (1x1) \oplus (0x1) = 0 \oplus 0 \oplus 0 \oplus 0 = 0 \oplus 0 = 0$$

$$\text{Untuk bit 3 : } (1x1) \oplus (0x1) \oplus (1x0) \oplus (0x1) = 1 \oplus 0 \oplus 0 \oplus 0 = 1 \oplus 0 = 1$$

$$\text{Untuk bit 4 : } (1x1) \oplus (0x0) \oplus (1x0) \oplus (0x0) = 1 \oplus 0 \oplus 0 \oplus 0 = 1 \oplus 0 = 1$$

$$\text{Untuk bit 5 : } (1x0) \oplus (0x1) \oplus (1x0) \oplus (0x0) = 0 \oplus 0 \oplus 0 \oplus 0 = 0 \oplus 0 = 0$$

$$\text{Untuk bit 6 : } (1x0) \oplus (0x0) \oplus (1x1) \oplus (0x0) = 0 \oplus 0 \oplus 1 \oplus 0 = 0 \oplus 1 = 1$$

$$\text{Untuk bit 7 : } (1x0) \oplus (0x0) \oplus (1x0) \oplus (0x1) = 0 \oplus 0 \oplus 0 \oplus 0 = 0 \oplus 0 = 0$$

Jadi bit untuk 1010 adalah 1011010

2. Penghitungan untuk penjumlahan Noise + Sinyal Menggunakan XOR

$$= [1011010] + [0011000]$$

$$= (1 \oplus 0)(0 \oplus 0)(1 \oplus 1)(1 \oplus 1)(0 \oplus 0)(1 \oplus 0)(0 \oplus 0)$$

$$= 1000010$$
3. Penghitungan untuk perkalian noise+sinyal dengan matriks generator transport

$$= [1000010]$$

1	0	0
0	1	0
0	0	1
0	1	1
1	0	1
1	1	0
1	1	1

Untuk bit 1 :

$$(1x1) \oplus (0x0) \oplus (0x0) \oplus (0x0) \oplus (0x1) \oplus (1x1) \oplus (0x1) = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 1 \oplus 0 \oplus 1 \oplus 1 = 1 \oplus 0 = 1$$

Untuk bit 2 :

$$(1x0) \oplus (0x1) \oplus (0x0) \oplus (0x1) \oplus (0x0) \oplus (1x1) \oplus (0x1) = 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0 \oplus 0 \oplus 1 \oplus 0 = 0 \oplus 1 = 1$$

Untuk bit 3 :

$$(1x0) \oplus (0x0) \oplus (0x1) \oplus (0x1) \oplus (0x1) \oplus (1x0) \oplus (0x1) = 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 0 \oplus 0 \oplus 0 \oplus 0 = 0 \oplus 0 = 0$$

Maka sindrome dan letak error yg salah = 110 & posisi 6 yg error

V. KESIMPULAN DAN SARAN

Dari hasil analisa dan perancangan simulasi Pengkodean Hamming yang telah penulis lakukan dapat diambil kesimpulan, bahwa :

1. Program ini dapat digunakan untuk 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.
2. Pengetahuan mahasiswa bertambah dan dapat mengetahui bagaimana proses Hamming tersebut.

DAFTAR PUSTAKA

- [1] http://www.batan.go.id/ppin/lokakarya/LKSTN_17/SLAMET.pdf
- [2] http://e-riset.darmajaya.ac.id/jurnal-ik/wp-content/uploads/2009/10/@TMZaini_NaskahPublikasi_Deteksi_error.pdf
- [3] <http://sutanto.staff.uns.ac.id/files/2008/09/hamming.pdf>
- [4] <http://lenterakecil.com/pengertian-metode-simulasi/>
- [5] <http://michael.dipperstein.com/hamming/index.html>
- [6] [http://en.wikipedia.org/wiki/Hamming\(7,4\)](http://en.wikipedia.org/wiki/Hamming(7,4))
- [7] <http://www.mathworks.com/matlabcentral/fileexchange/38290-74hamming-code-ber-simulation>

- [8] <http://www.azizi.ca/school/ece1501/hamming.pdf>
- [9] <http://docs.oracle.com/javase/tutorial/>
- [10] http://en.wikipedia.org/wiki/Binary_symmetric_channel
- [11] Todd K. Moon, "Error Correction Coding Mathematical Methods And Algorithms", 2005, Wiley-Interscience.
- [12] Dony Ariyus dan Rum Andri K.R "komunikasi data" 2008 yogyakarta, ANDI
- [13] Sutopo, Ariesto Hadi dan Masya, Fajar, "Pembrograman Berorientasi Objek dengan Java", 2005, Yogyakarta, Graha Ilmu.
- [14] Siallagan, Sariadin, "Pemrograman Java", 2009, Yogyakarta, ANDI OFFSET.
- [15] Forouzan, Behrouz A. "Data Communications and Networking Fourth Edition", 2007, New York, McGraw-Hill.
- [16] Aini, Aurora Nur, "Konstruksi Lexicographic Untuk Membangun Kode Hamming (7,4,3)"