

Perancangan Error Detection System And Error Correction System Menggunakan Metode Hamming Code Pada Pengiriman Data Text

Ahmad Alfi Albar Lubis¹, Dr. Poltak Sihombing, M.Kom², Ir.Arman Sani M.T³

Program Studi SI Ilmu Komputer, Universitas Sumatera Utara

Jalan Universitas No. 9 Kampus USU Medan 20155

¹ ahmad.alfialbar@students.usu.ac.id

² poltak@usu.ac.id

³ arman_sani@yahoo.com

Abstrak— Sistem pengiriman data pada saat ini masih kurang maksimal dan sering terjadi kesalahan dalam proses pengirimannya. Sebagian besar sistem pengiriman data sekarang ini belum dapat mengurangi kesalahan pada pengiriman. Pada penelitian ini akan dirancang sebuah aplikasi *Error Detection System and Error Correction System* Pada Pengiriman Data Text. Metode yang penulis gunakan dalam perancangan sistem ini adalah metode *Hamming Code*. Sistem ini dapat menampilkan pesan *error* yang sudah dideteksi kesalahannya dan kemudian dikoreksi kesalahannya. Aplikasi ini dirancang dengan menggunakan Visual Basic (VB). Dari hasil pengujian sistem ini data yang akan dikirimkan akan dideteksi terlebih dahulu, setelah terdeteksi jika ada kesalahan maka aplikasi akan mengoreksi kesalahan yang telah terdeteksi. Hasil pengujian menunjukkan bahwa *error* yang terjadi pada saat pengiriman data, dikarenakan kesalahan pada bit-bit yang dikirim, maka terjadilah *error*.

Kata Kunci— Error Detection, Error Correction, Hamming Code.

I. PENDAHULUAN

Pada era teknologi saat ini, komunikasi data menjadi aspek yang sangat penting. Menurut Ariyus dan Andri (2008), Komunikasi data didefinisikan sebagai suatu cara untuk menyampaikan atau menyebarkan data, informasi, berita, ataupun pendapat dengan menggunakan media transmisi data. Dilihat dari definisi tersebut, kepentingan komunikasi data terletak pada kelancaran penyampaian dan penyebaran datanya. Kelancaran penyampaian dan penyebaran data ini akan membawa dampak kepada kelancaran suatu proses bisnis yang bergantung pada data dan informasi.

Ada beberapa metode yang digunakan untuk mendeteksi *error* dan mengoreksi *error* yang terjadi. Salah satunya adalah dengan menggunakan metode *Hamming Code*. Metode ini merupakan salah satu jenis *linier error correction code* yang sederhana. Keuntungan yang didapatkan dengan menggunakan metode ini adalah 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* saat pendeteksian, data tidak perlu ditransmisikan ulang

tetapi langsung dikoreksi di simpul tujuan. Maka dari itu, penulis termotivasi untuk mengangkat konsep dan mengimplementasikan metode *Hamming Code* tersebut untuk melakukan pemeriksaan (*detection*) dan pemulihan (*correction*) *error* pada data berbentuk teks yang ditransmisikan dari komputer sumber ke komputer tujuan dengan menggunakan media transmisi *wireless*. Penulis mengharapkan, dengan digunakannya metode ini, maka dapat meminimalisir terjadinya kerusakan (*error*) pada saat proses transmisi data dilakukan dengan media *wireless*.

II. TINJAUAN PUSTAKA

A. Komunikasi Data

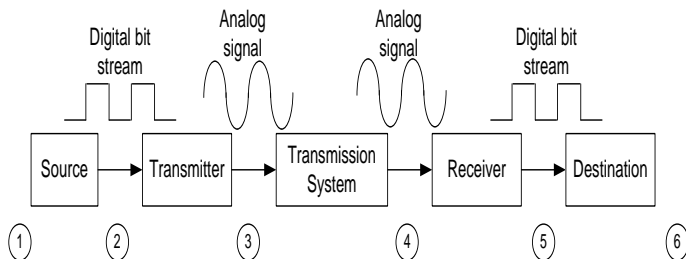
Komunikasi data didefinisikan sebagai suatu cara untuk menyampaikan atau menyebarkan data, informasi, atau berita dengan menggunakan media transmisi data (Ariyus dan Andi, 2008). Kerangka/model komunikasi data digital dasar yang sederhana ditunjukkan pada Gambar 1.



Gambar 1 Kerangka Komunikasi Dasar

Source (sumber) merupakan suatu alat untuk membangkitkan data sehingga dapat ditransmisikan. Contoh yang sering dijumpai untuk sebuah source adalah telepon dan Personal Computer. Transmitter (pengirim) berfungsi untuk melakukan konversi data ke bentuk signal analog untuk ditransmisikan. Biasanya data dibangkitkan dari sistem source (sumber) tidak ditransmisikan secara langsung dalam bentuk aslinya. Transmitter bertugas memindah dan menandai informasi dengan cara yang sama seperti menghasilkan signal elektromagnetik yang dapat ditransmisikan melewati beberapa sistem transmisi berurutan. Contoh transmitter yang sering dijumpai adalah modem yang berfungsi mentransformasikan digital bit stream yang diterima dari personal computer menjadi signal analog yang melintasi jaringan telepon. Source dan transmitter merupakan bagian dari source system.

Transmission system (sistem transmisi) berupa jalur transmisi tunggal (*single transmission line*) atau jaringan kompleks (*complex network*) yang menghubungkan antara sumber dengan tujuan. *Receiver* (penerima) berfungsi untuk mengkonversi kembali signal analog yang diterima dari sistem transmisi menjadi data digital. *Destination* (tujuan) berfungsi menangkap data yang dihasilkan oleh *receiver*. *Receiver* dan *destination* merupakan bagian dari *destination system*. Proses transmisi data secara terperinci diperlihatkan pada Gambar 2.



Gambar 2 Proses Transmisi Data

Pada proses 1, informasi dimasukkan. Pada proses 2 data masukan dikirimkan ke *transmitter*. Pada *transmitter* (proses 3), kemudian dikonversi ke dalam bentuk signal analog. Proses 4 menerima signal analog pada receiver yang kemudian akan mengubah kembali signal analog tersebut menjadi signal digital (proses 5). Informasi dapat dibaca di *destination*.

B. Error Detection And Error Correction

Pada saat data berada dalam transmission system terdapat kemungkinan 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 dijumpai pada telepon selular (Green, D. C, 2000).

Secara garis besar, metode pendeteksi *error* (*error detection*) dan pengontrol *error* (*error controller*) dapat dibagi menjadi dua bagian besar dengan perincian:

Error detection and correction terdiri dari dua metoda, yaitu:

- Manual, dengan menggunakan operator.
- Automatic

C. Metode Hamming Code

Metode hamming code merupakan salah satu metode pendeteksi *error* dan pengkoreksi *error* (*error detection and error correction*) yang paling sederhana. Metode ini menggunakan operasi logika XOR (Exclusive-OR) dalam proses pendeteksian *error* maupun pengkoreksian *error*. Input dan output dari metode ini berupa bilangan biner. Hamming code merupakan salah satu jenis linier *error correcting code*

yang sederhana dan banyak dipergunakan pada peralatan elektronik (Satra, 2007).

Metode *hamming code* bekerja dengan menyisipkan beberapa buah *check bit* ke data. Jumlah *check bit* yang disisipkan tergantung pada panjang data. Rumus untuk menghitung jumlah *check bit* yang akan disisipkan ke dalam data. Data 2^n bit, $c = (n+1)$ bit, dimana c adalah jumlah *check bit* yang disisipkan.

Tabel kenaikan data bit dan check bit

Data Bit	Check Bit
2	2
4	3
8	4
16	5
32	6
64	7
128	8
256	9

Gambar 3 tabel kenaikan data bit dan check bit

Check bit kemudian disisipkan pada data pada posisi yang dihitung menggunakan rumus perhitungan posisi *check bit*. Rumus perhitungan posisi *Check Bit* $C^i = 2^{i-1}$. Sehingga dengan rumus posisi tersebut, didapat posisi *check bit* yang akan diletakkan pada data diperlihatkan pada gambar.

Check Bit	Posisi
C_1	1
C_2	2
C_3	4
C_4	8
C_5	16
C_6	32
C_7	64
C_8	128
C_9	256

Gambar 4 tabel posisi check bit

D. Proses Pendeteksian Error

- Hitung panjang data masukan dari metode *hamming code* yang merupakan hasil penjumlahan dari panjang data masukan dengan panjang *check bit*. Panjang data keluaran dari metode *hamming code* sama dengan panjang data masukan dari metode *hamming code*.
- Tandai posisi bit yang merupakan posisi dari *check bit*. Posisi selain posisi *check bit* merupakan posisi data bit.
- Tentukan rumus perhitungan dari masing-masing *check bit*. untuk $n = 1$ hingga jumlah dari *check bit*, lakukan hal berikut:
 - Catat semua posisi dimana bit n dari member position bernilai 1, kecuali posisi bit itu sendiri. Member position merupakan bentuk biner dari posisi bit. Rumus dari *check bit* n sama dengan operasi XOR dari posisi-posisi yang dicatat.

E. Proses Pendeteksian Error

- Hitung berapa panjang bit yang diterima dan original.
- Cek tabel posisi *check bit* dan ekstrak *check bit* nya.
- Hitung kembali *check bit* nya dengan bit yang didapat.
- Konversikan operasi XOR ke bentuk decimal.

III. ANALISIS DAN PERANCANGAN SISTEM

A. Analisis Permasalahan

Masalah utama yang mendapat perhatian adalah pendeteksian dan pengkoreksian pengiriman data teks. Untuk pengirim data, penyebab masalah yang mungkin muncul adalah pesan yang dikirimkan oleh pengguna mungkin bersifat rahasia, akibatnya bila pesan ini dicuri oleh orang yang tidak bertanggung jawab kemungkinan akan merugikan pengguna atau pengirim.

B. Analisis Kebutuhan Fungsional

Kebutuhan fungsional yang harus dimiliki oleh sistem pendeteksi dan pengkoreksi kesalahan adalah:

- Sistem harus menyediakan sumber daya yang melakukan proses pendeteksi dan pengkoreksi.
- Sistem pendeteksi dilakukan dengan operasi XOR.
- Sistem dapat menghasilkan jaringan syaraf tiruan dengan kinerja maksimal.
- Sistem memberikan keluaran file text yang disisipkan *check bit*.
- Sistem menyediakan fungsi pengkoreksi setelah data itu dikirim.

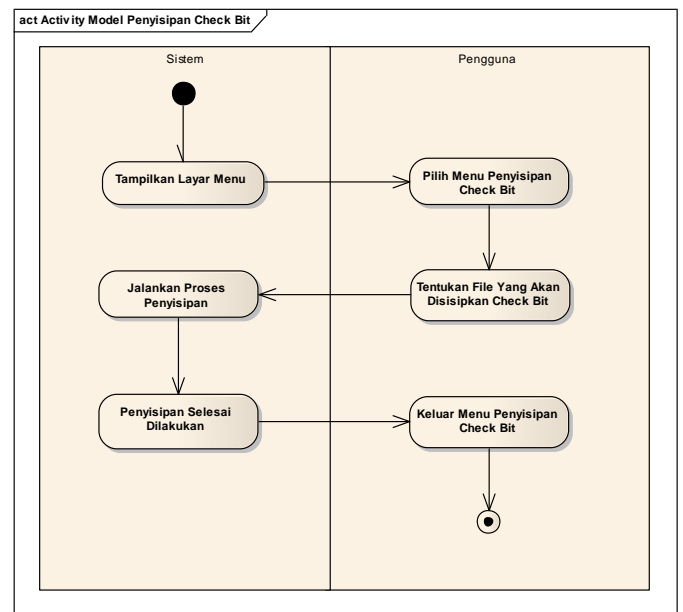
C. Kebutuhan Non-Fungsional Sistem

Untuk mendukung kinerja sistem, sistem sebaiknya dapat berfungsi sebagai berikut:

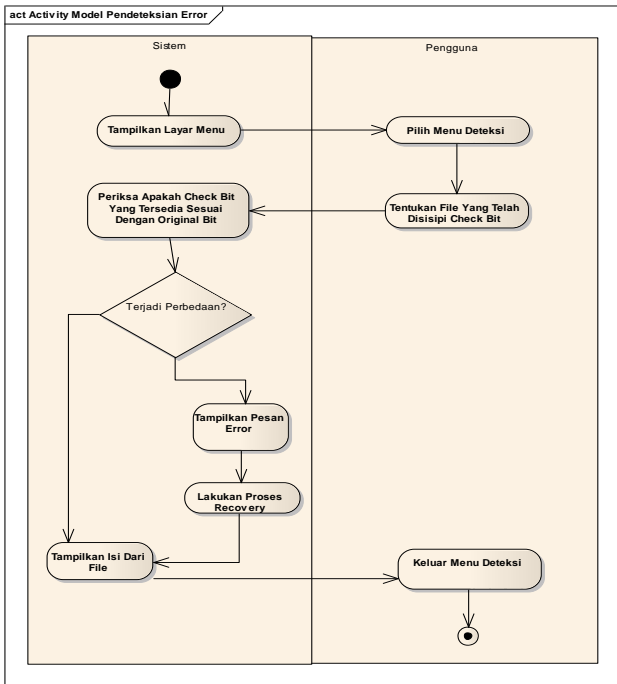
- Penggunaan perangkat lunak tidak akan mengganggu kinerja komputer
- Tampilan antarmuka sistem dapat dengan mudah dimengerti oleh pengguna sistem.
- Pengguna sistem harus menghindari munculnya biaya tambahan.

D. Pemodelan

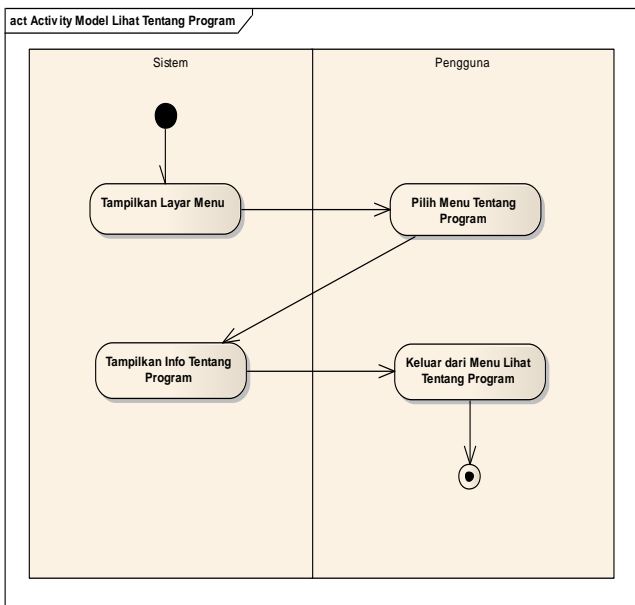
Pemodelan menggunakan Unified Modeling Language (UML) dalam mendesain dan merancang aplikasi Error Detection System and Error Correction System pada pengiriman data teks. UML yang akan digunakan ialah use case diagram.



Gambar 5 Activity Diagram Proses Penyisipan Check Bit

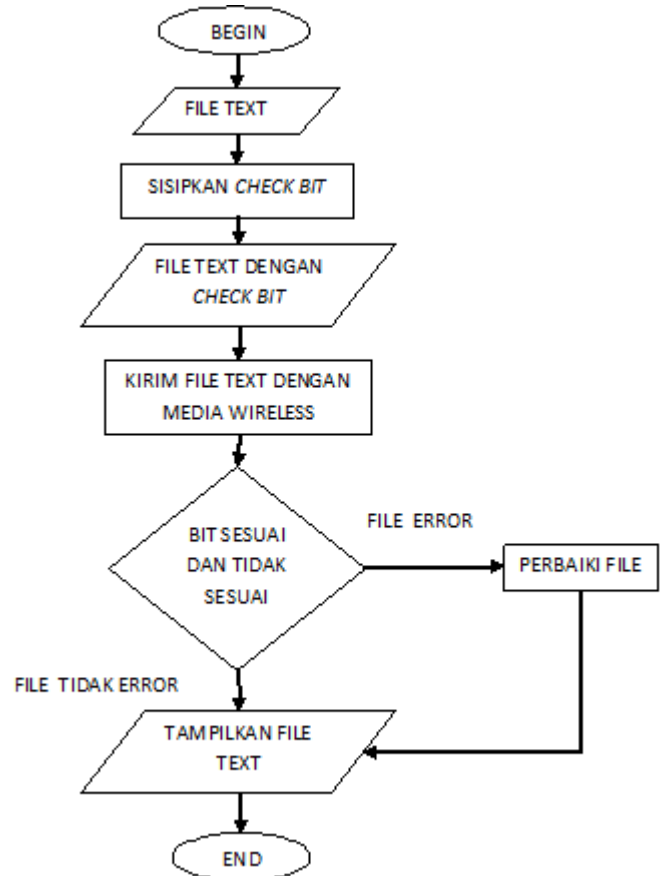


Gambar 6 Activity Diagram Pendeteksian Error



Gambar 7 Activity Diagram Proses Lihat Tentang Program

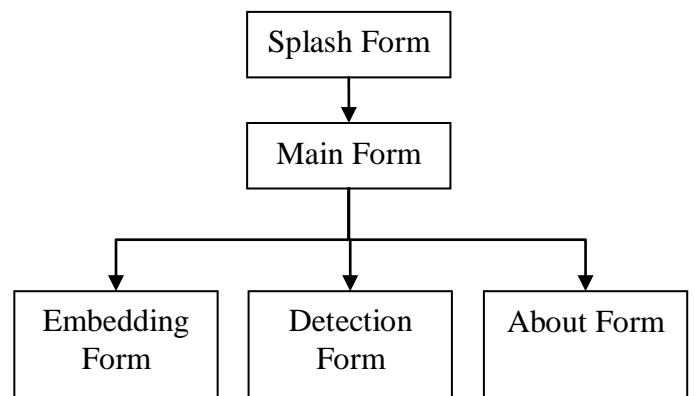
E. Flowchart



Gambar 8 Diagram Alir Kerja Sistem

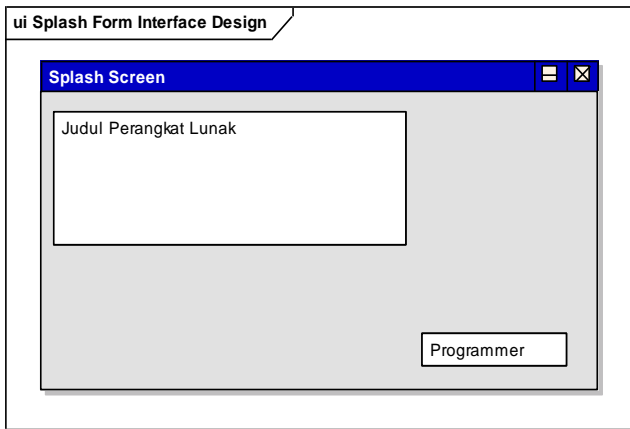
F. Perancangan Antarmuka

Antarmuka dirancang untuk memudahkan dalam membangun tampilan program yang akan dibuat. Sistem yang akan dibangun pada penelitian ini terdiri dari empat form yaitu Main Form, Embedding Form, Detection Form, Show Form. Struktur masing-masing form diperlihatkan pada Gambar 9.



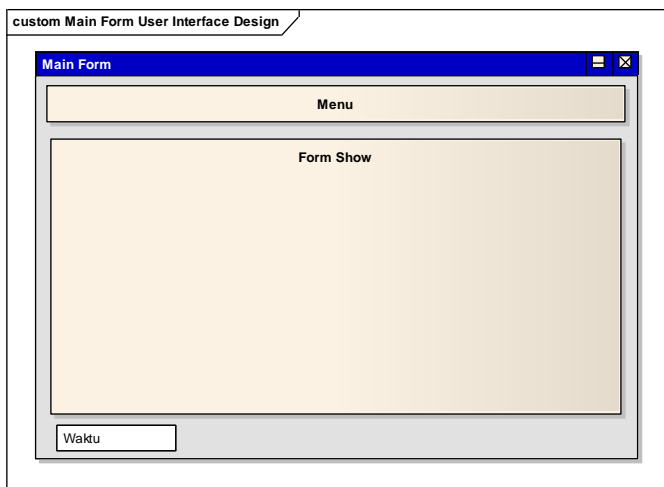
Gambar 9 Struktur pada form Sistem

1) *Antarmuka Splash Form*: Splash Form merupakan form yang tampil selama lima detik saat aplikasi dijalankan. Setelah lima detik tersebut, form ini akan memanggil form utama.



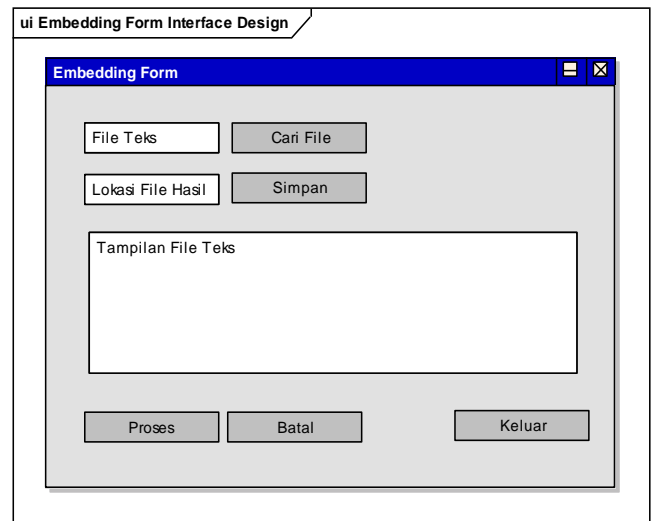
Gambar 10 Antarmuka Splash Form

2) *Antarmuka Main Form*: Main Form merupakan form induk dari sistem yang akan dibangun. Form ini berfungsi untuk memanggil form-form anaknya dan menampilkannya. Form anakan dari Main Form adalah Embedding Form dan Detection Form.



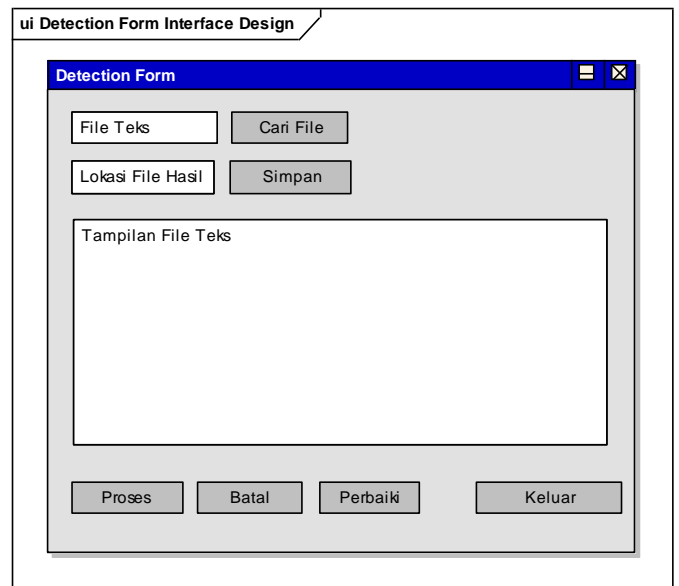
Gambar 11 Rancangan Antarmuka Main Form

3) *Antarmuka Embedding Form*: Embedding Form merupakan form untuk menampung proses pembacaan file teks dan penyisipan check bit pada file teks yang dibaca. Form ini akan meminta masukan file berformat .txt untuk diproses penyisipan check bit. Keluarannya berupa file teks yang telah disisipkan check bit nya.



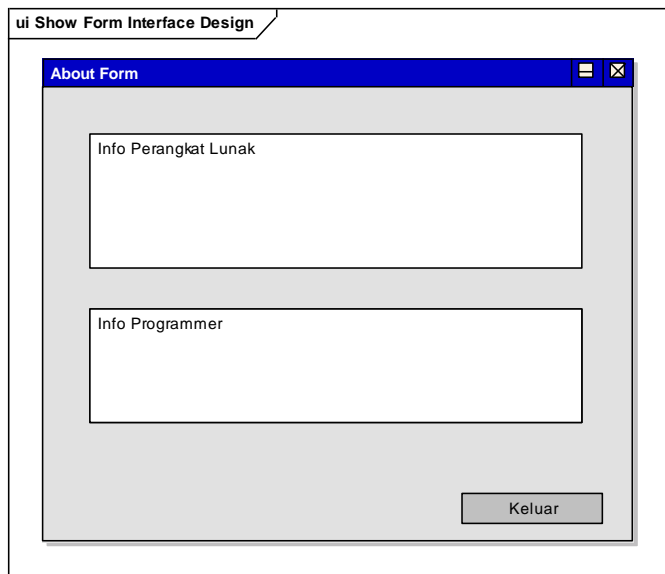
Gambar 12 Antarmuka Embedding Form

4) *Antarmuka Detection Form*: Detection Form merupakan form yang digunakan oleh penerima file teks yang telah disisipi oleh check bit. Berfungsi untuk melakukan deteksi pada file teks apakah file tersebut mengalami error atau tidak. Form tersebut akan memberikan pesan jika tidak terdapat error pada file teks tersebut. Jika file teks mengalami error, maka form ini akan mengeksekusi proses perbaikan error dan memberikan pesan file teks error dan telah diperbaiki.



Gambar 13 Antarmuka Detection Form

5) *Antarmuka About Form* :About Form merupakan form yang berfungsi untuk menampilkan info mengenai perangkat lunak yang dibangun. Menampilkan versi aplikasi dan informasi pembuat program.



Gambar 14 Antarmuka About Form

IV. IMPLEMENTASI DAN PENGUJIAN SISTEM

A. Implementasi Sistem

Seluruh proses pada fase perancangan akan ditransformasikan pada fase pemrograman ke dalam produk perangkat lunak. Pada fase ini, perangkat lunak dibangun dengan menggunakan bahasa pemrograman Visual Basic 6. Berikut adalah kebutuhan sumber daya yang diperlukan untuk menjalankan perangkat lunak dan tampilan ketika perangkat lunak tersebut dijalankan. Penerapan sistem dilakukan dengan menggunakan komputer dengan sistem operasi Windows Seven Ultimate.

B. Implementasi Hamming Code

Contoh proses HAMMING CODE dengan menggunakan kata "HALO"
 A: PENYISIPAN CHECK BIT
 Hitung panjang data masukan : HALO
 "HALO" = 4 Karakter
 1 Karakter = 1 Byte = 8 bit
 "HALO" 32 bit = **0100 1000 0100 0001 0100 1100 0100 1111**
 32 bit = 2^5 bit
 Maka Check Bit = 5 + 1 = 6 bit
 Sehingga panjang bit yang akan dikirim = 32 + 6 bit = 38 bit
 Hitung nilai check bit

$$\begin{aligned}
 C1 &= M1 \oplus M2 \oplus M4 \oplus M5 \oplus M7 \oplus M9 \oplus M11 \oplus M12 \oplus M14 \oplus M16 \oplus M18 \oplus M20 \oplus M22 \oplus M24 \oplus M26 \\
 &\oplus M27 \oplus M29 \oplus M31 \\
 &= 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \\
 &= 0 \\
 \\
 C2 &= M1 \oplus M3 \oplus M4 \oplus M6 \oplus M7 \oplus M10 \oplus M11 \oplus M13 \oplus M14 \oplus M17 \oplus M18 \oplus M21 \oplus M22 \oplus M25 \oplus \\
 &M26 \oplus M28 \oplus M29 \oplus M32 \\
 &= 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \\
 &= 1 \\
 \\
 C3 &= M2 \oplus M3 \oplus M4 \oplus M8 \oplus M9 \oplus M10 \oplus M11 \oplus M15 \oplus M16 \oplus M17 \oplus M18 \oplus M23 \oplus M24 \oplus M25 \oplus \\
 &M26 \oplus M30 \oplus M31 \oplus M32 \\
 &= 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \\
 &= 0 \\
 \\
 C4 &= M5 \oplus M6 \oplus M7 \oplus M8 \oplus M9 \oplus M10 \oplus M11 \oplus M19 \oplus M20 \oplus M21 \oplus M22 \oplus M23 \oplus M24 \oplus M25 \oplus \\
 &M26 \\
 &= 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \\
 &= 1 \\
 \\
 C5 &= M12 \oplus M13 \oplus M14 \oplus M15 \oplus M16 \oplus M17 \oplus M18 \oplus M19 \oplus M20 \oplus M21 \oplus M22 \oplus M23 \oplus M24 \oplus M25 \\
 &\oplus M26 \\
 &= 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \\
 &= 1 \\
 \\
 C6 &= M27 \oplus M28 \oplus M29 \oplus M30 \oplus M31 \oplus M32 \\
 &= 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \\
 &= 0
 \end{aligned}$$

Hasilnya :

$$\begin{aligned}
 &0100 1101 1000 0110 0000 1011 0110 0011 1011 \\
 &10
 \end{aligned}$$

B: PENDETEKSI ERROR

Panjang bit yang diterima = 38,
 Original bit = 32
 Check bit = 6

Hasilnya:

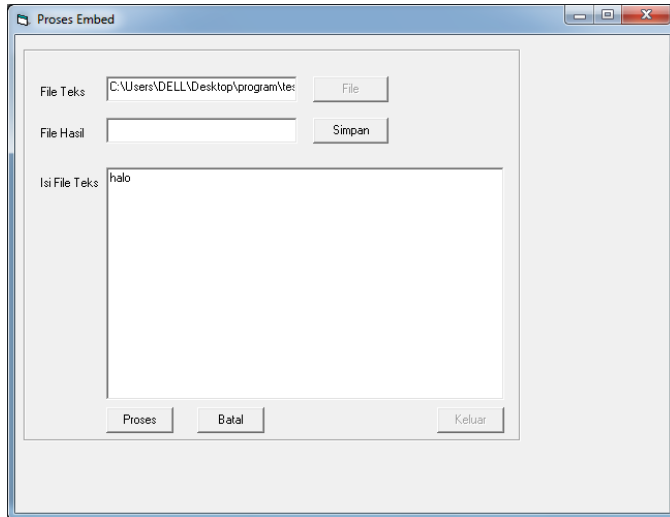
Check Bit yang didapat = 0 1 0 1 1 0
 Check Bit yang di ekstrak = 0 1 0 1 1 0
 Sama berarti tidak terdapat kesalahan pada pengiriman data.
Contoh terdapat kesalahan:
 Check Bit yang didapat = 0 1 1 1 1 0
 Check Bit yang di ekstrak = 0 1 0 1 1 0
 Tidak sama berarti terdapat kesalahan pada pengiriman data.
 Check Bit yang didapat = 0 1 1 1 1 0
 Check Bitt yang di ekstrak = 0 1 0 1 1 0
 Hasil X = 0 0 1 0 0 0

$001000 = 8_{10}$
 Hanya terdapat satu kesalahan karena $8 <$ dari panjang data = 32 bit, dan samadengan posisi Check Bit.
 Kesalahan hanya pada satu Posisi Yaitu Bit Ke 8 Dari Inputan
 Contoh bit yang salah : **0100 1100 1000 0110 0000 1011 0110 0011 1011 10**.
 Bit ke8 = 0 Dinegasikan = 1
 Barisan bit yang benar = **0100 1101 1000 0110 0000 1011 0110 0011 1011**

C) Pengujian Aplikasi

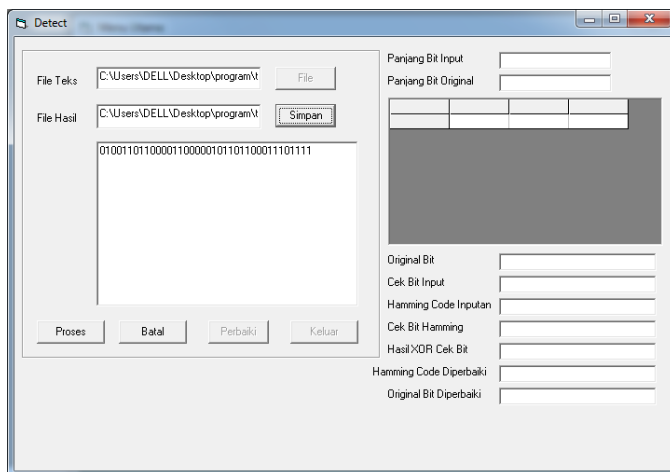
Pengujian adalah proses pemeriksaan atau evaluasi sistem atau komponen sistem secara manual atau otomatis untuk memverifikasi apakah sistem memenuhi kebutuhan yang dispesifikasikan atau mengidentifikasi perbedaan antara hasil yang diharapkan dengan hasil yang terjadi.

1) Proses Penyisipan Check Bit



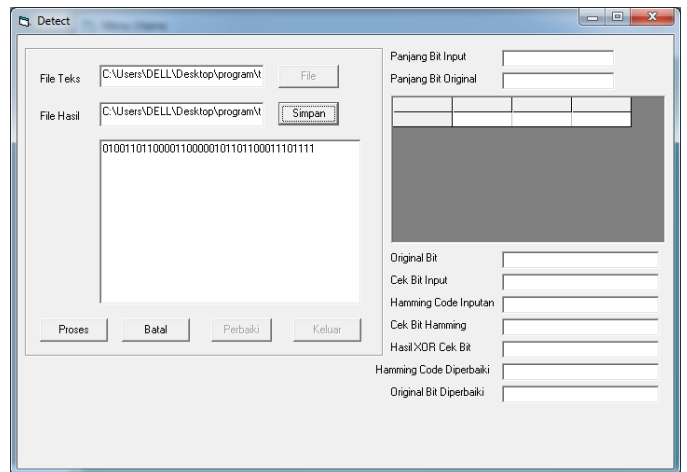
Gambar 15 Proses Penyisipan Check Bit

2) Proses Pendeteksi Error

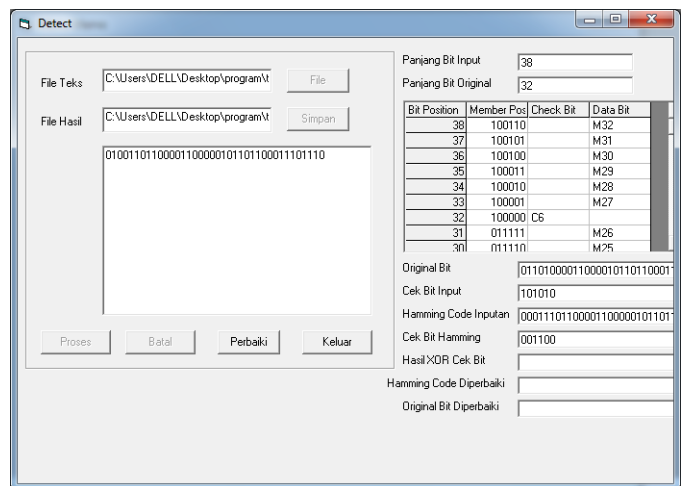


Gambar 16 Proses Pendeteksi Error

3) Proses Pesan Terdapat Kesalahan

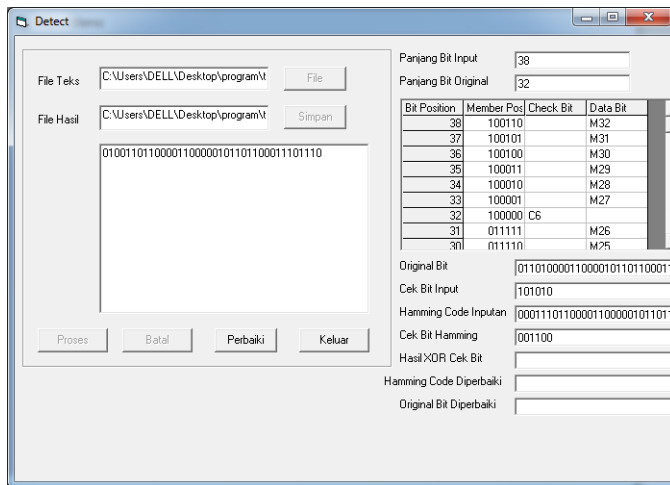


Gambar 17 Proses Pesan Terdapat Kesalahan



Gambar 18 Proses Pesan Terdapat Kesalahan Lanjutan

4) Proses Pendeteksi dan Pengkoreksi Kesalahan



Gambar 19 Proses Pendeteksi dan Pengkoreksi Kesalahan

V. KESIMPULAN DAN SARAN

Sesuai dengan hasil uji coba dan pembahasan program yang dilakukan, maka dapat ditarik kesimpulan sebagai berikut:

1. Metode hamming code mampu mendeteksi kesalahan bit pada file yang diterima oleh penerima file. Pengecekan dilakukan dengan membandingkan nilai *check* bit dengan nilai *original* bit yang ada pada file tersebut.
2. Metode hamming code dapat melakukan koreksi kerusakan bit (*bit error*) sebanyak satu bit atau disebut dengan *single bit error detection*. Sebaliknya metode hamming code tidak dapat mendeteksi jika terdapat lebih dari satu *error* bit. Jika terdapat kasus tersebut *error* bit yang terdeteksi hanya satu saja.

Adapun saran yang berkembang dalam pengerjaan aplikasi ini, dituangkan kedalam saran berikut:

1. Aplikasi di tonjolkan pada pengiriman jarak jauh.
2. Aplikasi dikuatkan pada enkripsi datanya agar tidak terjadi pencurian.
3. Mengembangkan aplikasi agar bisa dikirim melalui media wireless.

VI. REFERENSI

- [1] Dewobroto, Wiryanto.2003. Pemrograman Ms.Visual Basic 6.0, Elemedia Komputindo Jakarta.
- [2] Gupta, Brajesh K. and Rajeshwar Lal dua. 2011. 30 BIT Hamming Code For Error Detection and Correction with even parity and odd parity Check Method by using VHDL.
- [3] Gupta Shiv k. and Rajiv Kumar. 2011 Reducing to fault Errors in Communication Channel System.International Journal of advances in engineering and Technology.
- [4] Hadi,Rahadian. 1997. Pemrograman Windows Api dengan Microsoft Visual Basic.Penerbit:PT alex Media Komputindo ,Jakarta.

- [5] Jogiyanto.2005. Analisis dan Desain system informasi.Penerbit ANDI,Yogyakarta.
- [6] Kurniawan, arief." Pembelajaran perangkat lunak untuk simulasi hamming code".
- [7] Poli Alain and Llorenç Huguet..1989.Error Correcting Codes:Theory and Application.Masson and prentice Hall Internasional,Ltd.UK.
- [8] Yani, Ahmad.2008.Panduan menjadi teknisi jaringan computer,Bandung.Penerbit: kawan pustaka