



SATIN – Sains dan Teknologi Informasi

journal homepage : <http://jurnal.stmik-amik-riau.ac.id>



Error Detection System dan Error Correction System pada Citra Digital dengan Hamming Code

Rahmaddeni

Teknik Informatika STMIK Amik Riau
rahmaddeni@stmik-amik-riau.ac.id

Koko Harianto

Teknik Informatika STMIK Amik Riau
kokoharianto@stmik-amik-riau.ac.id

Abstrak

Citra digital atau yang disebut sebagai gambar merupakan media informasi yang terus berkembang penggunaannya hingga saat ini. Setiap pengguna informasi harus memastikan bahwasanya tidak terjadi kerusakan pada gambar agar memperoleh informasi yang benar, sehingga diperlukannya suatu metode untuk mengetahui apakah gambar mengalami kerusakan. Metode Hamming Code merupakan metode yang memiliki sistem pemeriksaan dan perbaikan terhadap error atau kerusakan yang terjadi pada data digital dengan melakukan pemeriksaan terhadap setiap bit data, sehingga metode ini dianggap memiliki kemampuan untuk melakukan pemeriksaan dan perbaikan kesalahan pada citra digital. Penelitian ini menjadi salah satu solusi dalam pengolahan citra digital serta dapat memberikan kontribusi dalam pemanfaatan teori hamming code pada matakuliah *Arsitektur dan Organisasi Komputer serta matakuliah Pengolahan Citra Digital*.

Kata Kunci: Citra Digital, Error Detection, Error Correction, Hamming Code

1. Pendahuluan

Kemampuan gambar sebagai media untuk menyampaikan informasi, mengharuskan para pengguna informasi untuk tetap menjaga keaslian dan sedini mungkin mengetahui perubahan atau kerusakan yang mungkin terjadi pada gambar tersebut. Pendeteksian kerusakan pada gambar membutuhkan ketelitian yang tinggi, dan dikarenakan kemampuan

manusia tidak mampu mendeteksi keseluruhan warna digital, sehingga dibutuhkan kemampuan komputer untuk mendeteksi kerusakan yang mungkin terjadi pada gambar. Salah satu metode untuk mendeteksi kerusakan (error) yang berkembang saat ini adalah Hamming Code.

Hamming code adalah salah satu metode pendeteksi error dan pengkoreksian error (error detection and error correction) yang menggunakan operasi logika XOR dalam proses pendeteksian error maupun pengkoreksian error (Alfi *et al.* 2012). Hamming code bekerja dengan memberikan kode biner tambahan pada data yang berfungsi sebagai bit-bit pendeteksi kesalahan (check bit). Bit-bit ini akan memberikan gambaran mengenai kondisi data yang sesungguhnya sehingga kesalahan (error) yang terjadi dapat dideteksi dengan mudah, karena terdapat suatu keterkaitan antara data dengan bit-bit pendeteksi kesalahan (Irwanto & Widyarningsih 2009).

2. Citra Digital

Penjelasan lebih lanjut oleh Balza Achmad dan Kartika Firdausy dalam bukunya menjelaskan bahwa pada citra warna, setiap titik pada citra mempunyai warna spesifik yang merupakan kombinasi dari tiga warna dasar, yaitu merah (red), hijau(green), dan biru (blue). Setiap warna dasar mempunyai intensitas dengan maksimum 255 (8 bit), sehingga jumlah kombinasi warna dari nilai RGB-nya sebanyak 224 atau lebih dari 16 juta warna. Kombinasi warna RGB terlihat pada gambar 2.1. Pada gambar di bawah terlihat bahwa terdapat 3 warna dasar, yaitu Merah, Hijau, dan Biru. Hasil kombinasi 3 warna tersebut akan

memberikan warna-warna baru seperti kuning, ungu, dan biru muda (Achmad & Firdausy 2013).

Citra warna adalah citra yang lebih umum. Warna yang terlihat pada citra bitmap merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau dan biru. Setiap pixel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB = Red Green Blue). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna. Berarti setiap pixel mempunyai kombinasi warna sebanyak $28.28.28 = 224 = 16$ juta warna lebih. Itulah sebabnya format ini dinamakan true color karena mempunyai jumlah warna yang cukup besar sehingga bisa dikatakan hampir mencakup semua warna di alam (Munir 2004).

3. Hamming Code

Debi Pariaman Deri (Deri 2013) dalam jurnalnya mengatakan bahwa 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 dari algoritma Hamming Code berupa bilangan biner. Untuk data $2n$ bit, jumlah check bit yang disisipkan ada sebanyak $c = (n+1)$ bit, sehingga didapat tabel kenaikan data bit dan check bit seperti yang diperlihatkan pada tabel 1, sedangkan Check bit yang disisipkan ke dalam data ditempatkan pada posisi yang dihitung berdasarkan rumus perhitungan posisi check bit berikut $C_i = 2^{i-2}$

Tabel 1. 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

Check bit ini yang digunakan untuk melakukan proses pendeteksi error (error detection) dan pengoreksi error (error correction). Algoritma Hamming code adalah sebagai berikut:

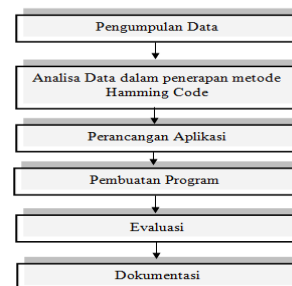
Berdasarkan kerangka kerja diatas, maka masing – masing tahapan tersebut dapat dijelaskan sebagai berikut :

1. Pengumpulan Data

1. Hitung panjang data input dari algoritma Hamming Code yang merupakan hasil penjumlahan dari panjang input data dan panjang check bit. Panjang data output dari algoritma Hamming Code sama dengan panjang data input dari algoritma Hamming Code.
2. Tandai semua posisi bit yang merupakan posisi dari check bit. Posisi selain posisi check bit merupakan posisi dari data bit.
3. Tentukan rumus perhitungan dari masing-masing check bit.
Untuk $n=1$ hingga jumlah dari check bit, lakukan hal berikut:
 - a. Catat semua posisi dimana bit n dari member position bernilai 1, kecuali posisi bit itu sendiri. Member position merupakan bentuk biner dari posisi bit.
 - b. Rumus dari check bit n sama dengan operasi XOR dari posisi-posisi yang dicatat.
4. Hitung nilai dari check bit ntuk data input dan data output.
5. Jika nilai check bit input tidak sama dengan nilai check bit output berarti terdapat kesalahan (error).
6. Lakukan operasi XOR terhadap check bit input dan check bit output.
7. Konversikan hasil operasi XOR ke dalam bentuk bilangan decimal.
8. Jika nilai dari hasil operasi XOR lebih besar daripada panjang data input atau nilai dari hasil operasi XOR sama dengan posisi dari check bit, maka terdapat lebih dari satu kesalahan (error).
9. Jika tidak, maka jumlah kesalahan (error) hanya satu dan hasil operasi merupakan posisi data yang terdapat kesalahan (error).

4. Kerangka Kerja Penelitian

Adapun kerangka kerja dalam penelitian ini sebanyak 6 tahapan yang digambarkan berikut ini:



Gambar 1. Kerangka kerja penelitian

Melakukan pengumpulan data citra digital bitmap yang memiliki warna RGB yang diperoleh dari komputer.

2. Analisa Data

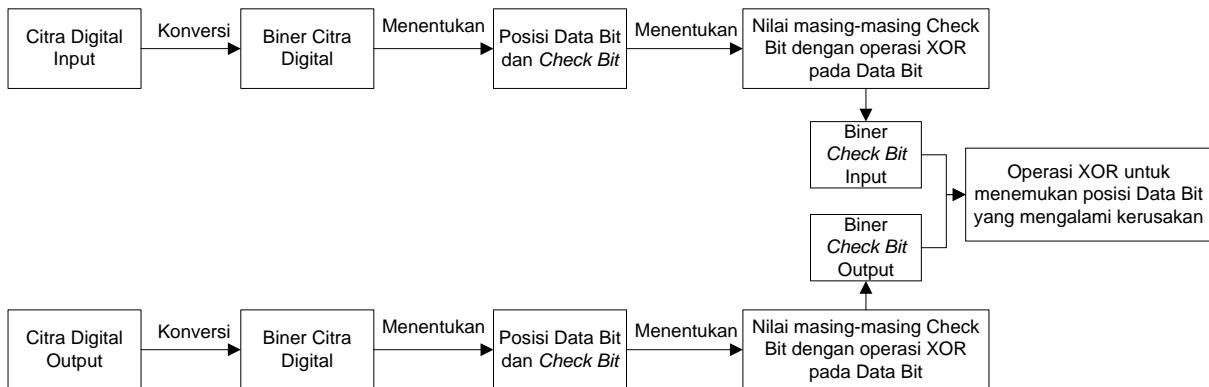
Membuat analisa terhadap data yang diperoleh dari hasil pengumpulan data dengan

- menggabungkan error detection system dan error correction system pada hamming code.
3. Perancangan Aplikasi
Membuat rancangan aplikasi sesuai data yang ada berdasarkan tahapan metode yang ditetapkan pada tahapan analisa data.
 4. Pembuatan Program
Membuat sebuah aplikasi dengan menerapkan metode hamming code untuk mengetahui adanya error dan mengkoreksi error tersebut dari sebuah citra digital berdasarkan sistem yang sudah dirancang.
 5. Evaluasi
Menguji seluruh spesifikasi terstruktur dan aplikasi secara keseluruhan. Pada tahap ini dilakukan uji coba aplikasi yang telah selesai dibuat. Proses uji coba ini diperlukan untuk memastikan bahwa aplikasi yang telah dibuat sudah benar, sesuai dengan karakteristik yang ditetapkan dan tidak ada kesalahan yang terkandung didalamnya.
 6. Dokumentasi
Mendokumentasikan sistem yang dibuat kedalam sebuah jurnal/artikel.

5. Penerapan Hamming Code

Metode hamming code bekerja dengan data bit, maka setiap nilai warna yang terdapat disetiap piksel citra digital terlebih dahulu dikonversi menjadi data biner. Proses konversi dilakukan terhadap kedua citra

digital, yaitu citra digital sebagai input, dan juga citra digital sebagai output. Secara keseluruhan, proses-proses yang digunakan dalam penerapan hamming code pada citra digital diperlihatkan pada gambar 2.



Gambar 2. Proses-proses penggunaan metode hamming code

Setelah diketahui nilai *biner check* pada masing-masing citra input dan citra output, maka selanjutnya adalah melakukan operasi XOR terhadap kedua nilai *check bit* tersebut untuk menemukan posisi *data bit* yang mengalami kerusakan. Dengan diketahui posisi *data bit* yang mengalami kerusakan, maka hamming code secara otomatis akan mengubah nilai *data bit* tersebut. Perubahan *data bit* dilakukan dengan mengganti nilai menjadi 0 atau 1. Apabila *data bit* pada posisi *error* adalah 0, maka diganti menjadi 1, begitu juga sebaliknya.

6. Implementasi

Implementasi merupakan tahapan untuk membuktikan teori-teori yang telah dibahas sebelumnya. Berikut ini implentasi dari tahapan Hamming Code:

1. Tahapan utama adalah menentukan jenis input. Penelitian ini menggunakan citra digital sebagai

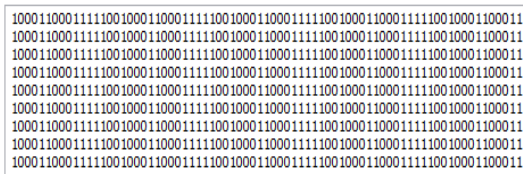
data input, sehingga terlebih dahulu pilih citra digital sebelum melakukan operasi hamming code.



Cek Biner Gambar

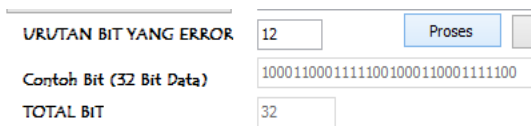
Gambar 3. Citra input

2. Setelah menetapkan citra digital sebagai data input, maka selanjutnya adalah melakukan konversi dari nilai warna pada setiap piksel (dalam implementasi ini, menggunakan contoh warna merah saja). Konvesi nilai warna dapat dilakukan dengan melakukan klik pada tombol “Cek Biner Gambar”.



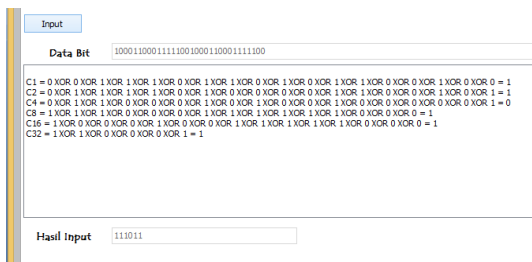
Gambar 4. Contoh data biner citra input

- Setelah seluruh data biner diketahui, maka tahapan selanjutnya adalah mengimplementasikan metode hamming code (contoh 32 bit). Implementasi hamming code dilakukan dengan menentukan bit data yang akan diubah, serta menentukan urutan biner data yang menjadi check bit dan juga data bit.



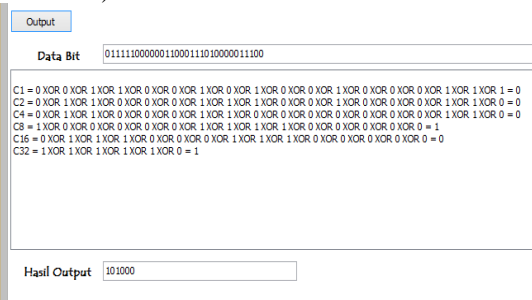
Gambar 5. Bit data yang mengalami perubahan

- Selanjutnya mengoperasikan rumus untuk menentukan semua *check bit* dari data input.



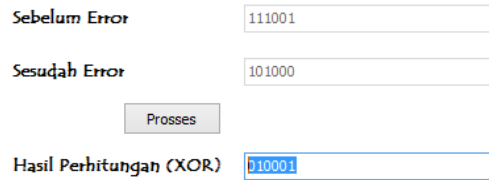
Gambar 6. Perhitungan check bit citra input

- Menentukan operasi hamming code pada data output. Data output adalah data yang sudah mengalami perubahan (dimisalkan mengalami kerusakan).



Gambar 7. Perhitungan check bit citra output

- Tahapan terakhir adalah dengan melakukan operasi XOR terhadap nilai biner hasil input dan nilai biner hasil output.



Gambar 8. Hasil perhitungan operasi xor terhadap data input dan data output

Setelah dilakukan perhitungan XOR terhadap data input dan data output, maka akan menemukan hasil sebagai penanda terhadap posisi *data bit* yang mengalami kerusakan. Untuk mengetahui posisi data yang mengalami kesalahan tersebut, maka lakukan pencocokan nilai hasil perhitungan XOR dengan tabel data input.

Tabel 2. Urutan data bit dan check bit data input

Bit Posisi	Biner	Data Bit	Check Bit
13	0001101	D9	
14	0001110	D10	
15	0001111	D11	
16	0010000		C16
17	0010001	D12	

Berdasarkan hasil perhitungan XOR pada gambar 8 dengan tabel 2, maka diketahui bahwa *data bit* yang mengalami kerusakan (*error*) berada pada posisi data ke 12 (D12). Setelah diketahui bit yang rusak, maka bit tersebut digantikan dengan kebalikan (NOT) dari bit tersebut. Misalnya nilai pada D12 adalah 1 (satu), maka digantikan dengan 0 (nol), begitu juga sebaliknya.

7. Kesimpulan

Setelah menyelesaikan perancangan aplikasi deteksi bit check in error dengan algoritma Hamming Code, maka penulis dapat menarik kesimpulan sebagai berikut :

- Metode Hamming Code mampu mendeteksi kesalahan pada citra digital dengan melakukan pengecekan yang dilakukan dengan cara membandingkan nilai check bit dengan nilai original bit yang ada pada citra digital tersebut.
- Dalam penerapannya Algoritma Hamming Code dalam mendeteksi dan mengkoreksi bit check in error hanya mampu melakukan deteksi kerusakan sebanyak satu bit atau disebut single error detection.
- Dalam pembuatan ataupun perancangan aplikasi deteksi error pada Delphi haruslah sesuai dengan spesifikasi keperluan program baik hardware

maupun software agar sesuai dengan kebutuhan sebuah aplikasi sebagai penunjang sistem atau aplikasi tersebut.

8. Saran

Dalam hal ini penulis ingin memberikan beberapa saran yang mungkin berguna untuk pengembangan lebih lanjut pada perancangan aplikasi deteksi bit error antara lain :

1. Deteksi bit check in error juga dapat dikembangkan untuk melakukan pemeriksaan kesalahan untuk setiap bit data yang mengalami error.
2. Disarankan untuk menggunakan teknik lain dalam melakukan pemeriksaan seluruh error, karena hamming code hanya memeriksa satu bit error.

9. Referensi

- Achmad, B. & Firdausy, K., 2013. *Pengolahan Citra Digital Menggunakan Delphi*, Yogyakarta: Penerbit Andi.
- Alfi, A. et al., 2012. Perancangan Error Detection System And Error Correction System Menggunakan Metode Hamming Code Pada Pengiriman Data Text. *Jurnal Online Program Studi SI Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara*, 1. Available at: <http://jurnal.usu.ac.id/index.php/alkhawarizmi/article/view/503>.
- Deri, D.P., 2013. PERANCANGAN APLIKASI DETEKSI BIT CHECK IN ERROR PADA TRANSMISI DATA TEXT DENGAN SINGLE ERROR CORRECTION MENGGUNAKAN ALGORITMA HAMMING CODE. *Pelita Informatika Budi Darma*, IV, pp.19–24.
- Irwanto, B. & Widyarningsih, S., 2009. Deteksi dan Koreksi Error Pada Pesan Digital Dengan Kode Hamming. *Jurnal Sains & Matematika (JSM)*, 17, pp.127–130. Available at: <http://ejournal.undip.ac.id/index.php/sm/article/view/3276>.
- Munir, R., 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Bandung: Informatika.