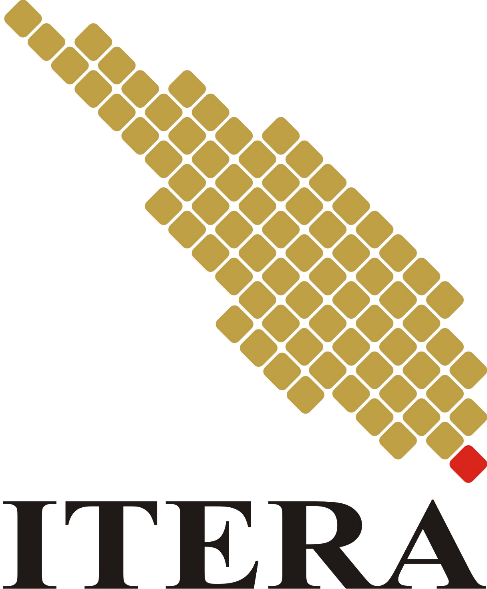
**UJIAN TENGAH SEMESTER**

**PENGOLAHAN SINYAL DIGITAL**

**PROGRAM HAMMING CODE 7,4 SOLVER**

**Mata Kuliah: Pengolahan Sinyal Digital**

**Dosen Pengampu: Martin Clinton Tosima Manullang, S.T.,M.T.**



**Disusun Oleh**

**Reza Octaviany 14117062**

**TEKNIK INFORMATIKA**

**INSTITUT TEKNOLOGI SUMATERA**

**2020**

BAB I   
PENDAHULUAN

* 1. Latar Belakang

Pengolahan sinyal digital adalah cara untuk melakukan berbagai operasi pemrosesan sinyal. Sinyal yang diproses adalah sinyal analog yang diubah terlebih dahulu menjadi sinyal digital dengan tahapan-tahapan seperti sampling, dan kuantisasi. Sampling adalah teknik pengubahan sinyal analog menjadi beberapa bit digital, sementara kuantisasi adalah proses pembandingan level-level sinyal hasil sampling tadi dengan tetapan level tertentu, sinyal diskrit yang ada akan disesuaikan levelnya dengan tetapan yang ada.

Dalam Pengolahan sinyal digital sering kali terjadi Noise, salah satu metode untuk mengolah noise tersebut adalah dengan cara menggunakan metode Hamming Code (7,4). Metode ini adalah metode dengan memecah sinyal yang diterima menjadi 7bit boolean, yang berisi informasi yang terkait satu sama lain. Oleh karna itu program ini dibuat, untuk menyelesaikan permasalahan Hamming code dengan cepat.

* 1. Rumusan Masalah

Program dibuat dalam bahasa pemrograman apa saja dengan ketentuan pada soal sebagai berikut:

1. Masukan dari sebuah program adalah mampu menampung 7bit boolean
2. Keluaran dari program mampu menentukan apakah ada error dalam 7bit boolean yang dimasukkan
3. Keluaran dari program mampu memperbaiki 7bit boolean yang error sehingga mengeluarkan 7bit boolean yang benar
4. Aturan yang digunakan dalam program ini didasarkan pada Hamming Code

BAB II   
LANDASAN TEORI

1. Landasan Teori

Pada Hamming code terdapat pecahan kode per-7bit yang di definisikan secara berurutan dengan abcdxyz, dimana abcd adalah bari kode utama dan xyz adalah pelengkapnya. Metode dilakukan dengan langkah sebagai berikut:

1. Mengidentifikasi kode, misalnya diberikan kode 1010111, maka identitas masing-masing bit kode tersebut adalah

1 0 1 0 1 1 1

a b c d x y z

1. Mencari x, y, z sebenarnya, dengan menggunakan rumus:

x = a ⊕ b ⊕ d

y = a ⊕ c ⊕ d

z = b ⊕ c ⊕ d

jika menggunakan kode yang diberikan maka hasilnya adalah sebagai berikut:

x = 1 ⊕ 0 ⊕ 0 = 1

y = 1 ⊕ 1 ⊕ 0 = 0

z = 0 ⊕ 1 ⊕ 0 = 1

1. Membandingkan dengan x, y, z yang ada pada 7bit kode awal. Maka diperoleh nilai y adalah salah. Dengan menggunakan tabel di bawah ini, kita akan mengganti satu atau dua bit boolean yang ada pada kode.

Tabel 1. Tabel kebenaran hamming code 7.4

|  |  |  |  |
| --- | --- | --- | --- |
| x | y | z | Error |
| 1 | 1 | 1 | - |
| 0 | 1 | 1 | x |
| 1 | 0 | 1 | y |
| 1 | 1 | 0 | z |
| 0 | 0 | 1 | a |
| 0 | 1 | 0 | b |
| 1 | 0 | 0 | c |
| 0 | 0 | 0 | d |

Maka nilai yang harus di ubah adalah nilai Y, sehingga nilai yang awalnya:

1 0 1 0 1 **1** 1

a b c d x **y** z

akan berubah menjadi,

1 0 1 0 1 **0** 1

a b c d x **y** z

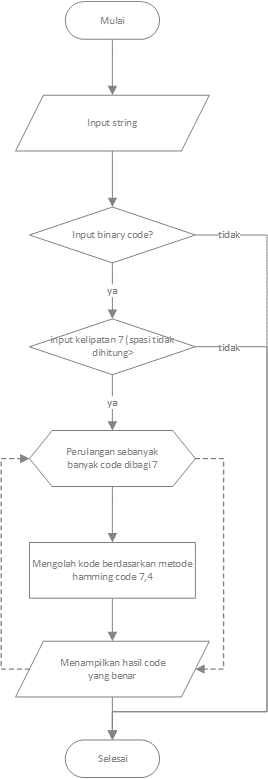
BAB III   
HASIL DAN PEMBAHASAN

1. Penjelasan Program

Program dibuat dengan kriteria sebagai berikut:

1. Bahasa pemrograman yang dipakai adalah C++
2. Program dapat menerima masukan berupa minimal 7bit boolean, Jika kurang atau kelipatan nya bukan 7bit, maka program akan berhenti.
3. Program hanya dapat menerima angka 1 atau 0 dan pemisah berupa spasi ‘ ’. Jika Program menerima inputan lain selain tiga hal tersebut, maka program akan dibatalkan.
4. Keluaran dari program mampu menentukan apakah ada error dalam pecahan 7bit boolean yang dimasukkan.
5. Keluaran dari program mampu memperbaiki 7bit atau lebih boolean yang error sehingga mengeluarkan 7bit boolean yang benar.
6. Aturan yang digunakan dalam program ini didasarkan pada landasan teori tentang Hamming Code.

Berikut adalah flowchart program

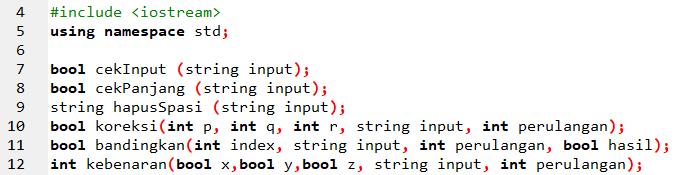


Gambar 1. Flowchart Program

1. Analisis Source Code Program

Gambar 2, menunjukkan library dan sub-program apa saja yang akan dipakai pada program ini. Seperti biasa library yang digunakan adalah iostream untuk input dan output program. Serta digunakan beberapa sub-program yang secara singkat memiliki kegunaan sebagai berikut:

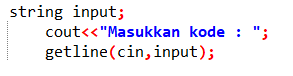
1. bool cekInput(), Mengecek input apakah terdapat karakter lain selain 1, 0, dan spasi.
2. bool cekPanjang(), Mengecek apakah input memiliki panjang 7 karakter atau kelipatannya, dengan tanpa menghitung spasi.
3. string hapusSpasi(), menghapus spasi yang ada pada string agar pemecahan blok string lebih mudah.
4. bool koreksi(), mencari nilai x, y, z yang sebenarnya berdasarkan nilai a, b, c, d.
5. bool bandingkan(), membandingkan nilai x, y, z yang telah dicari dan yang ada pada kode.
6. int kebenaran(), menentukan index error yang ada pada 7bit kode awal berdasarkan hasil dari bandingkan().



Gambar 2. Deklarasi program.

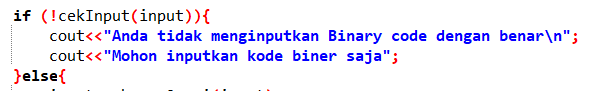
Gambar 14, menunjukkan main program dengan penjelasan masing-masing baris yang ada sebagai berikut:

1. Baris 15 sampai 17, adalah deklarasi variabel string input, menampilkan teks panduan input, dan menerima input dengan getline() agar input dapat menerima input berupa spasi.



Gambar 3. Baris 15 sampai 17

1. Baris 19 sampai 65 secara menyeluruh adalah percabangan yang memeriksa input berdasarkan fungsi cekInput(), jika program menerima string yang mengandung kode lain selain 1, 0, dan spasi ‘ ’ maka program akan berhenti. Jika tidak maka program berlanjut dari poin c hingga poin k.



Gambar 4. Baris 19 sampai 65

1. Baris 28, menentukan panjang blok 7bit yang ada, dengan membagi panjang string keseluruhan dengan 7 maka diperoleh banyak blok yang ada. Hal ini dilakukan agar pengecekan dilakukan per-blok, dan tidak terlalu panjang. Kemudian jumlah tersebut dikurangi dengan 1, karena perulangannya dimulai dari 0.



Gambar 5. Baris 28

1. Baris 30 sampai 55 melakukan perulangan sebanyak int perulangan



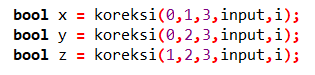
Gambar 6. Baris 30 sampai 55

1. Baris 31, mengoutputkan pemberitahuan blok berapa yang sedang dikoreksi



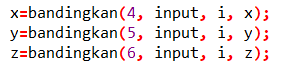
Gambar 7. Baris 31

1. Baris 33 sampai 35, mendeklarasikan boolean x, y, z dengan memanggil sub-program koreksi() . terdapat indeks p, q, r yang merupakan bagian dari rumus yang akan dipakai untuk menghitung masing-masing nilai boolean.



Gambar 8. Baris 33 sampai 35

1. Baris 37 sampai 39, mengubah nilai x, y, z yang sebelumnya telah di rumuskan, dengan memanggil sub-program bandingkan(). Sub program ini mengecek nilai x, y, z, kemudian mengubahnya menjadi nilai benar, jika nilai x, y, z pada kode awal, sama dengan x, y, z pada hasil pencarian dengan sub-program koreksi().



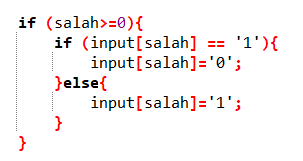
Gambar 9. Baris 37 sampai 39

1. Baris 41 mendeklarasikan integer salah, berdasarkan fungsi kebenaran(). Fungsi kebenaran mengecek bagian yang salah, kemudian menentukan index berapa yang harus diganti, dan menyimpan index tersebut ke integer salah.



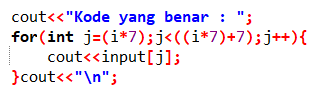
Gambar 10. Baris 41

1. Baris 43 sampai 49, berfungsi mengubah nilai yang salah sesuai dengan index yang telah ditentukan. Jika nilai tersebut bernilai 1 maka akan diubah menjadi 0, dan sebaliknya.



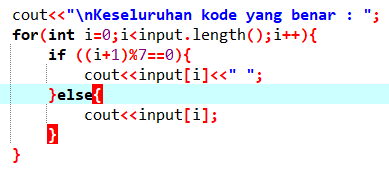
Gambar 11. Baris 43 sampai 49

1. Baris 51 dampai 54, mengoutputkan kode yang benar hanya pada blok yang sedang dikoreksi saja. Sehingga batas awalnya adalah (i\*7) dan batas akhirnya adalah (i\*7)+7. Satu persatu mengoutputkan array of char input[].

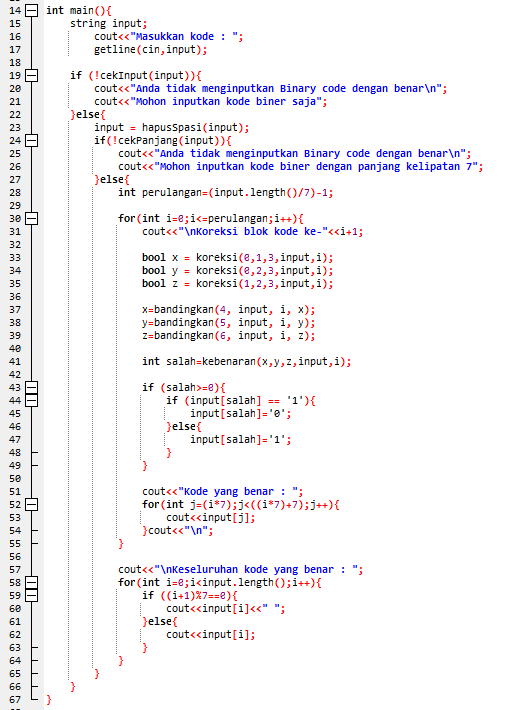


Gambar 12. Baris 51 dampai 54

1. Baris 57 sampai 64, mengoutputkan seluruh kode yang benar setelah dikoreksi, mengoutputkan hanya sekali di akhir program.

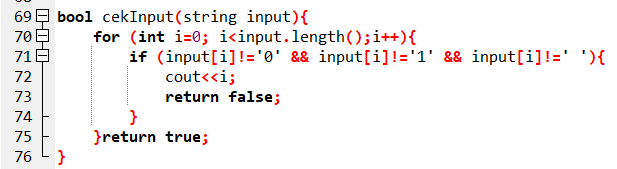


Gambar 13. Baris 57 sampai 64



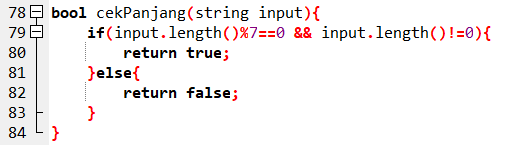
Gambar 14. Main Program

Gambar 15, menunjukkan fungsi boolean cekInput(). Yang berfungsi sebagai pengecek apakah input terdapat char lain selain 0, 1 dan spasi. Cara pengecekan dilakukan dengan melakukan perulangan sebanyak panjang input. Kemudian pada setiap perulangan, akan dilakukan pengecekan, jika input, bukan 0, 1, dan spasi, maka return false. Return false akan menyebabkan program pada main berhenti. Jika perulangan telah habis, dan tidak ada karakter lain, maka nilai fungsi ini akan berupa true, yang mengakibatkan program main tetap berjalan.



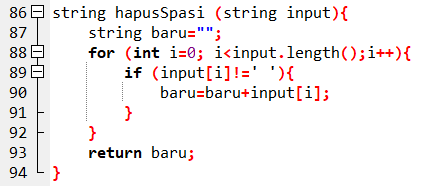
Gambar 15. Fungsi boolean cekInput().

Gambar 16, menunjukkan fungsi boolean cekPanjang(). Yang berfungsi sebagai pengecek apakah panjang input adalah kelipatan 7 atau bukan. Cara pengecekan dilakukan dengan melakukan modulus panjang dari string input, kemudian dicek apakah modulus tersebut bernilai 0 ketika panjang input tidak nol. Jika ya, maka fungsi akan bernilai true, jika tidak fungsi akan bernilai false.



Gambar 16. Fungsi boolean cekPanjang().

Gambar 17, merupakan fungsi string hapusSpasi(). Fungsi ini berguna untuk membuat string baru berdasarkan string inputan awal, hanya saja spasinya di hapus. Metode yang digunakan adalah dengan mendeklarasikan string baru yang kosong, kemudian melakukan perulangan perkarakter, sepanjang string awal, jika karakter bukan karakter spasi ‘ ’ maka string baru akan menambahkan anggotanya dengan string awal ke-i, jika string awal ke-i adalah spasi ‘ ’, maka akan dilewatkan dan tidak dijadikan anggota string baru. Kemudian setelah selesai, fungsi ini akan mengembalikan nilai string baru tersebut.



Gambar 17. Fungsi string hapusSpasi().

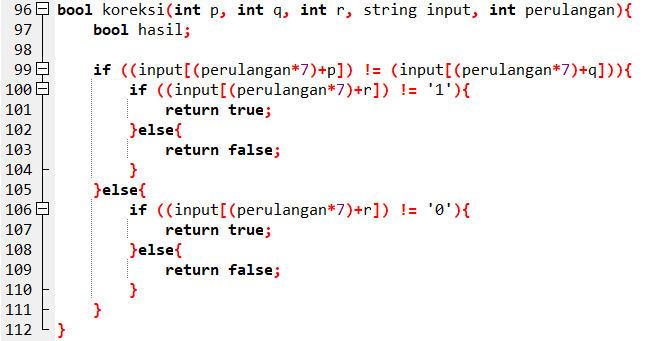
Gambar 18, merupakan fungsi boolean koreksi(). Fungsi ini merupakan fungsi yang berguna untuk mencari x, y, z secara teori. Artinya fungsi ini adalah representasi rumus hamming code 7,4. Hal yang dilakukan pada fungsi ini adalah:

1. Fungsi menerima index p, q, r yang didapat dari main program, yang merupakan index dari karakter yang akan dibandingkan untuk mencari x, y, atau z.
2. Int perulangan berguna untuk menyimpan blok ke berapa saat ini yang sedang dikoreksi

Contoh:

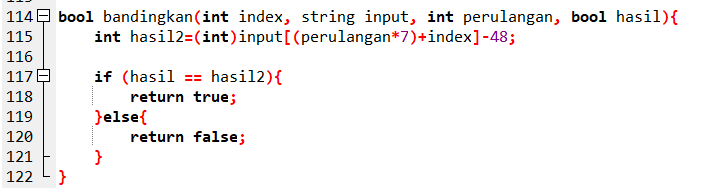
Misalnya kita berada di blok pertama, artinya saat ini nilai perulangan adalah 0. Kemudian kita akan mencari nilai x yang memiliki p=0, q=1, dan r=4. Artinya kita akan menggunakan index pada blok yang memiliki nilai 0, 1, 4 yaitu a, b, dan d.

1. Pada program baris 99, adalah melakukan pengecekan apakah a sama dengan b. input[(perulangan\*7)+p] , jika perulangan bernilai 0 maka hasilnya akan jadi input[0] jika sedang mencari nilai x.
2. Pada pencarian nilai x, Jika nilai a dan b tidak sama, maka akan bernilai 1, kemudian dilakukan pembandingan lagi apakah nilai d tidak sama dengan 1. Jika tidak sama maka fungsi akan mengembalikan nilai true, jika sama maka fungsi akan mengembalikan nilai false.
3. Jika nilai a dan b sama, maka akan bernilai 0. Kemudian dibandingkan dengan nilai d, jika d tidak sama dengan 0 maka program mengembalikan nilai true, sebaliknya jika d adalah 0 maka fungsi akan bernilai false.



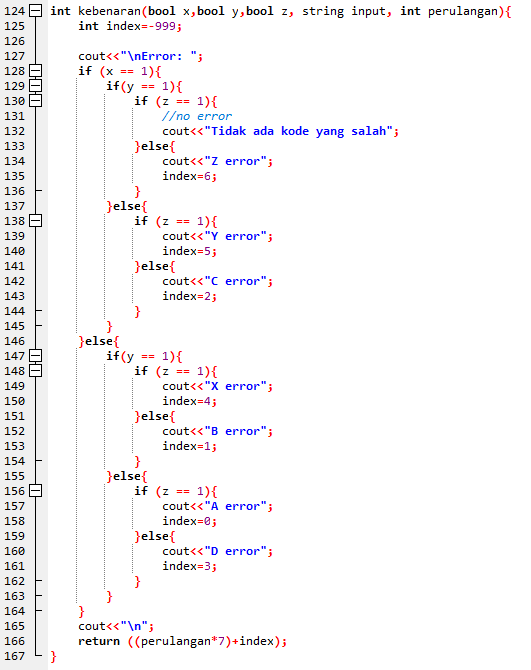
Gambar 18. Fungsi string koreksi().

Gambar 19, merupakan fungsi boolean bandingkan(). Fungsi ini berguna untuk membandingkan nilai x, y, z yang ada pada kode 7bit dengan yang telah dicari. Pada fungsi ini, mendeklarasikan int hasil2 untuk mengubah nilai input[] yang mula nya merupakan char menjadi integer. Input[] dikurangi dengan 48 sesuai dengan ketentuan yang terdapat pada ascii untuk mendapat angka 0 sampai 9. Kemudian hasil2 dibandingkan dengan boolean hasil. Jika sama maka akan bernilai true, jika beda akan mengembalikan nilai false.



Gambar 19. Fungsi boolean bandingkan().

Gambar 20, merupakan fungsi integer kebenaran(). Fungsi ini akan menerima nilai x, y, z yang telah diubah pada main program, dan mengecek eror pada kode, sesuai dengan Tabel 1. Kemudian fungsi ini mengembalikan nilai indeks yang harus diganti, dan mengoutputkan error yang ada pada blok kode 7bit tersebut.

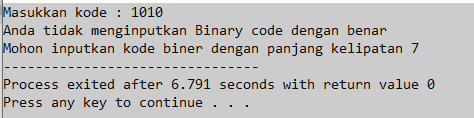


Gambar 20. Fungsi integer kebenaran().

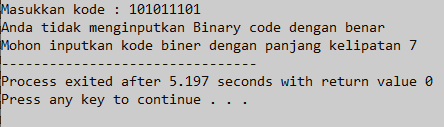
1. Capture Hasil Program

Telah dilakukan beberapa test untuk menguji kebenaran program. Baik dengan menggunakan input yang benar ataupun yang salah. Berikut ini adalah hasil test program yang telah di dokumentasikan:

Jika program diberi input bukan kelipatan 7bit maka program akan mengoutputkan eror dan akan berhenti. Misal kita beri inputan 1010 yang kurang dari 7 bit dan 101011101 yang lebih dari 7bit.

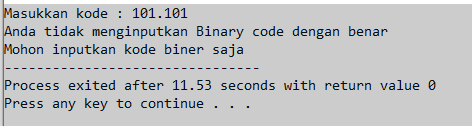


Gambar 21. Error bukan kelipatan 7 bit (kurang)



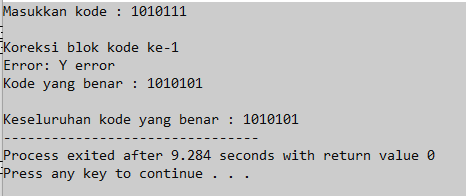
Gambar 22. Error bukan kelipatan 7 bit (lebih)

Jika program diberi input bukan hanya 0, 1 dan spasi ‘ ’ maka program akan mengoutputkan eror dan akan berhenti. Misal kita beri inputan 101.101 yang memiliki inputan titik (.)



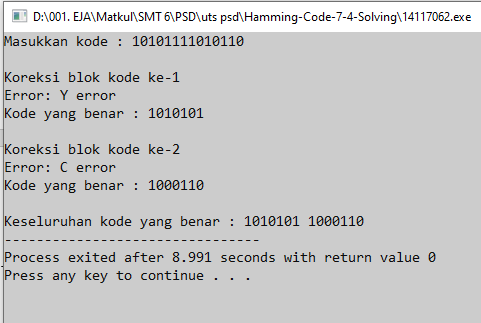
Gambar 23. Error ada karakter lain

Jika program diberi input yang sesuai, maka program akan berjalan mengikuti alur yang benar. Misal diberikan input 1010111.



Gambar 24. Menampilkan hasil koreksi 1 blok saja

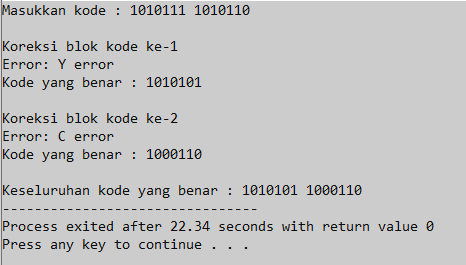
Jika program diberi input 2 blok nilai 7bit, maka akan terjadi perulangan untuk mengoreksi pada blok pertama dan kedua. Misal diberikan input 10101111010110



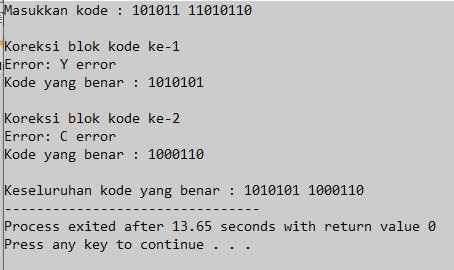
Gambar 25. Menampilkan hasil koreksi 2 blok 7bit

Jika program diberi input 2 blok nilai 7bit namun memiliki spasi, maka akan tetap terjadi perulangan untuk mengoreksi pada blok pertama dan kedua, walaupun diberikan spasi yang berlebih dan spasi yang berada tidak memisahkan blok per-7bit.

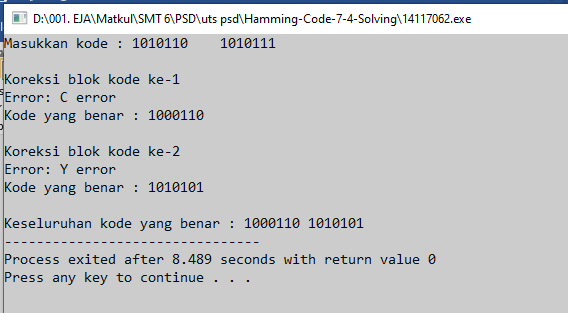
Misal diberikan input: {1010111 1010110},{101011 11010110},{1010111 1010110}



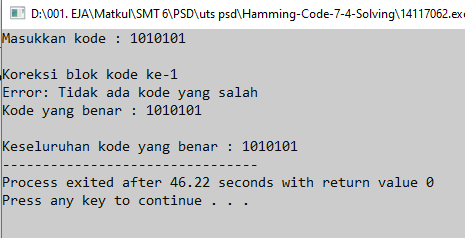
Gambar 26. Menampilkan hasil koreksi 2 blok dengan spasi



Gambar 27. Menampilkan hasil koreksi 2 blok dengan spasi tidak per-7bit

  
Gambar 28. Menampilkan hasil koreksi 2 blok dengan spasi berlebihan

Jika program diberi input yang tidak memiliki kesalahan, maka program akan memberikan notifikasi bahwa tidak ada eror yang terjadi pada program.



Gambar 29. Menampilkan hasil koreksi blok dengan nilai yang tidak error.

Lampiran:

1. Source code program juga telah diupload pada github dengan link: <https://github.com/recchan13/Hamming-Code-7-4-Solving/>