

Komputasi Geofisika 1:

Pemodelan dan Prosesing Geofisika dengan Octave/Matlab

Editor:
Agus Abdullah
Mohammad Heriyanto
Hardianto Rizky Prabusetyo

Judul Artikel:

**Konversi Data Biner 32 Bit Menjadi Desimal dengan Menggunakan Format
IBM dan IEEE**

Muhammad Rizaldi, Mochamad Rheza, Frans Riantoni, Virgiawan Listanto dan Abi Mawalid

GitHub: <https://github.com/Metkom/OSGPUP>

Homepage: <https://sites.google.com/site/metkomup>

University: <https://universitaspertamina.ac.id>

Contact Us: metkom.up@gmail.com



Universitas
Pertamina

Program Studi Teknik Geofisika
Fakultas Teknologi Eksplorasi dan Produksi
Universitas Pertamina
Jakarta | Maret 2018

Konversi Data Biner 32 Bit Menjadi Desimal dengan Menggunakan Format IBM dan IEEE

Muhammad Rizaldi^{*)}, Mochamad Rheza, Frans Riantoni, Virgiawan Listanto dan Abi Mawalid

Proyek Pemograman - GP2103 Metode Komputasi
Program Studi Teknik Geofisika
Fakultas Teknologi Eksplorasi dan Produksi, Universitas Pertamina
Jl. Teuku Nyak Arif, Simprug, Jakarta Selatan, Indonesia, 12220

^{*)} muhammadrizaldifarsah@gmail.com (corresponding author)

Abstrak

Komputer merupakan alat yang digunakan dalam melakukan sesuatu dalam mempermudah pekerjaan manusia. Komputer menggunakan bahasa pemograman dengan biner sedangkan manusia umumnya menggunakan sistem bahasa yang menggunakan desimal. Sehingga dibutuhkan konversi data biner ke desimal supaya pengguna dan komputer mengerti maksud dan tujuan dari perintah yang diberikan oleh pengguna atau komputer. Seperti percobaan ini, konversi data biner 32-bit ke desimal dengan menggunakan metode pemisahan digit data biner yakni sign, exp, dan fraction. Dan hasil yang ditemukan merupakan bilangan desimal yang berbeda tergantung dengan format yang digunakan. Format yang ditemukan dalam program ini ialah format IBM dan IEEE.

Kata-kata kunci: Data biner, desimal, IBM, IEEE

Tersedia online 5 Maret 2018 • Lisensi kode: MIT License • Bahasa: Indonesia

Kutip artikel ini sebagai berikut: Muhammad Rizaldi, Mochamad Rheza, Frans Riantoni, Virgiawan Listanto & Abi Mawalid. (2018). <https://doi.org/10.6084/m9.figshare.5947708>

1. PENDAHULUAN

Berdasarkan banyak ilmu yang berkembang atas dasar penerapan konsep dari matematika. Salah satunya perkembangan ilmu komputer yang sedang berkembang pesat dalam era informasi sekarang ini. Jaringan komputer, komputer grafis, aplikasi dari berbagai software diambil dari penerapan konsep dan pemikiran dari para ahli yang telah dirangkum dalam ilmu matematika. Banyak ilmu yang berkembang atas dasar penerapan konsep dari matematika. Ada beberapa sistem bilangan yang digunakan dalam sistem digital yang paling umum adalah sistem bilangan desimal dan biner. Sistem bilangan desimal merupakan sistem bilangan yang paling familiar.

Sistem biner merupakan salah satu konsep dasar sistem komputer. Biner merupakan bahasa yang digunakan oleh komputer dalam segala programnya sehingga sangat penting buat kita untuk memahami sistem kerja dari bilangan biner. Dalam sistem biner adalah sistem yang mengenal 2 buah angka yang disebut dengan istilah bit yakni 1 dan 0. Pada bilangan desimal juga akan ada sistem yang berbeda, terdapat 10 macam simbol yaitu: 0,1,2,3,4,5,6,7,8, dan 9. Sehingga untuk mempermudah pemakai komputer maka komputer harus bisa mengerti setiap perintah dari pengguna dan pengguna juga bisa menyampaikan maksud dan tujuannya dengan benar. Ekstensi setiap angka ini memiliki sistem tersendiri, sehingga untuk setiap bilangan akan berbeda nilainya apabila berbeda ekstensinya. Selain itu dalam proses data juga sangat mempengaruhi nilai/hasil dari proses data, contohnya data seismik yang ditemukan adalah dalam format biner agar lebih mempermudah pemahaman kita maka cukup dengan menggunakan data dengan format desimal.

2. METODOLOGI

Salah satu konsep dasar dalam sistem komputer yaitu adanya sistem biner dan desimal. Sistem bilangan biner hanya mempunyai dua macam simbol angka, yaitu 0 dan 1.

Jika kita bekerja dengan lebih dari satu bilangan, maka kita akan mengalami kebingungan apabila kita tidak memakai suatu tanda yang menyatakan dasar setiap bilangan. Secara umum, semua sistem digital bekerja dengan sistem bilangan biner sehingga sistem binerlah yang paling penting dalam sistem digital. Harga dalam desimal (puluhan) yang dinyatakan oleh suatu bilangan biner dapat dihitung dengan memakai rumus:

$$a_{n-1}a_{n-2}\dots a_1a_0 \text{ a}_{-1}a_{-2}\dots a_{-m} = a_{n-1} R^{n-1} + a_{n-2} R^{n-2} + \dots + a_1 R^1 + a_0 R^0 + a_{-1} R^{-1} + \dots + a_{-m} R^{-m} \tag{1}$$

Keterangan:

R = angka dasar dari pada sistem bilangan

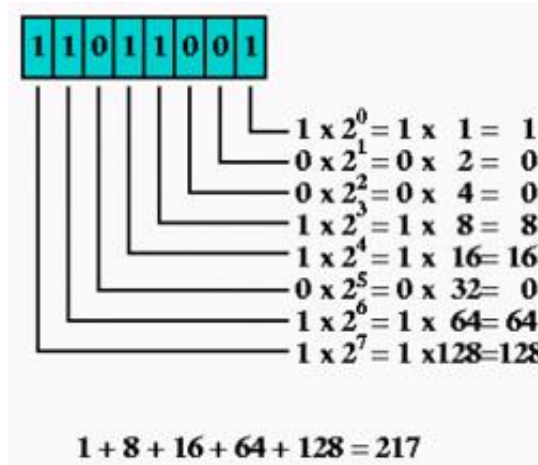
n = cacah angka yang menunjukkan bilangan bulat

m = cacah angka yang menunjukkan bilangan pecahan

Konversi dari biner ke desimal diperlukan untuk menerjemahkan kode hasil pengolahan sistem digital ke informasi yang dikenal oleh manusia. Konversi ini banyak dilakukan karena disamping cacah, angka biner jauh lebih besar dibandingkan dengan angka-angka pada sistem oktal dan heksadesimal. Selain itu, konversi ini sangat mudah. Konversi dari biner ke sistem bilangan desimal, seperti yang telah dijelaskan diatas, dapat dilakukan dengan memakai persamaan (1).

Jika kita perhatikan konversi dari biner ke desimal dengan memakai persamaan (1), maka dapat dilihat bahwa untuk bagian bulat (di kiri tanda koma) kita peroleh dengan melakukan perkalian dengan 2 setiap kita bergerak ke kiri. Untuk bagian pecahan, kita melakukan pembagian dengan 2 setiap kita bergerak ke kanan. Untuk melakukan konversi dari desimal ke biner kita melakukan sebaliknya, yaitu untuk bagian bulat bilangan desimal kita bagi dengan 2 secara berturut-turut dan sisa pembagian pertama sampai yang terakhir merupakan angka-angka biner paling kanan ke paling kiri. Untuk bagian pecahan, bilangan desimal dikalikan 2 secara berturut-turut dan angka di kiri koma desimal hasil setiap perkalian merupakan angka biner yang dicari, berturut-turut dari kiri ke kanan. Contoh berikut ini memperjelas proses itu.

Untuk melakukan konversi bilangan desimal ke biner ini, dilakukan pembagian untuk bagian bulatnya dan pengalihan untuk bagian pecahannya seperti yang dilakukan pada kedua contoh sebelumnya, dengan hasil sebagai berikut ini:



Gambar 1. Konversi data biner 8-bit ke desimal

Dari contoh terakhir ini dapat dilihat bahwa untuk bagian pecahan, pengalihan dengan 2 akan berulang-ulang menghasilkan deretan 1,6; 1,2; 0,4; 0,8 yang berarti bahwa deretan angka biner 11001100 akan berulang terus. Ini berarti bahwa ada bilangan pecahan puluhan yang tidak dapat disajikan dalam biner dengan ketelitian 100 %. Kesalahan atau ralat konversi itu semakin kecil apabila jumlah angka biner (bit) yang dipergunakan lebih besar. Bagaimanapun juga, jumlah bit dalam setiap sistem digital sudah ditentukan sehingga ketelitian pengkodean untuk setiap sistem digital sudah tertentu juga.

2.1. Konversi Bilangan Biner 32 Bit Ke Desimal Dalam Program Octave.

Konversi Binari ke IBM

Proses konversi ini membutuhkan sistem pemisahan biner, dimana bagian pertama disebutkan sebagai *sign* yang terdiri dari satu digit, yang kedua disebut eksponen yang terdiri dari tujuh digit, dan bagian yang ke tiga disebut *fraction* bilangan yang terdiri dari 24 digit bilangan biner. Setelah pemisahan kelompok bilangan biner tersebut, dilakukan proses penentuan tanda. Tanda dengan angka satu digunakan untuk menunjukkan tanda negatif sedangkan angka 0 digunakan untuk menentukan tanda positif. Kemudian pada kelompok eksponen kita melakukan proses konversi seperti biasa untuk bilangan biner delapan digit namun yang dikonversikan hanyalah tujuh digit dikarenakan pada digit pertama telah digunakan untuk penentuan tanda.

Bilangan *fraction* adalah bilangan pelengkap desimal sehingga pada prosesnya harus diselesaikan

$$fraction = \sum(X(i) + 2^m) \tag{2}$$

dimana, $m = p - 1$

Keterangan: X = bilangan biner

m = variabel pengkali

Rumus konversi binari ke IBM adalah:

$$-1^{sign} * fraction * (16^{exp-64}) \tag{3}$$

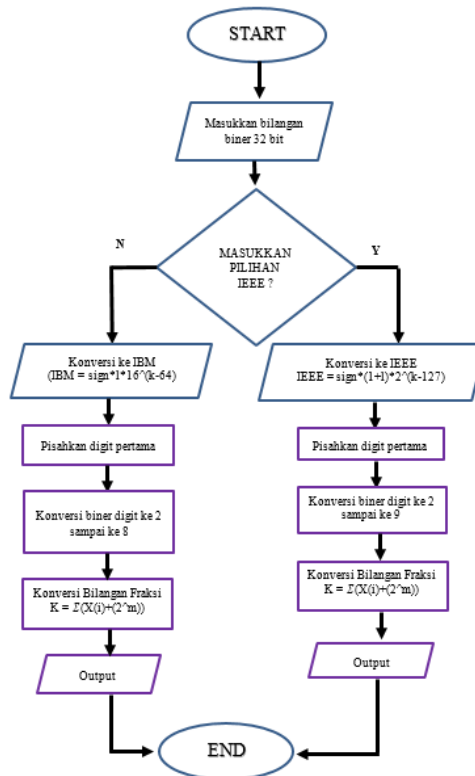
Konversi Binary ke IEEE

Pada konversi data biner ke desimal dengan format IEEE sedikit berbeda dengan IBM. Pada IEEE dilakukan pemisahan untuk ketiga kelompok biner seperti IBM. Pada digit pertama berfungsi sebagai petunjuk tanda negtif atau positif. Kemudian dilakukan proses konversi bilangan Eksponen dari biner ke desimal tetapi untuk proses IEEE jumlah digit yang digunakan dalam konversi ialah delapan digit setelah digit pertama. Sedangkan proses konversi *fraction* menjadi bilangan dilakukan sama seperti IBM.

Rumus konversi binari ke IEEE adalah

$$-1^{sign} * fraction * (2^{exp-127}) \tag{4}$$

2.2. Flowchart Pemograman



Gambar 2. Flowchart pemograman

3. HASIL PEMOGRAMAN DENGAN OCTAVE

Program untuk mengonversi bilangan biner ke desimal berdasarkan IEEE.

```
clear; clc;
x = input('Masukkan nilai biner (32) = ');
% example
% x = [0 1 0 0 0 0 1 0 0 1 1 0 1 1 0 0 1 0 1 0 1 1 0 1 0 0 0 1 0 1 0 1];
d1 = x(1);
sign = (-1)^(d1);
m = 0; k = 0;
a = -1; l = 0;
for i = 9:-1:2;
    k = k+(x(i)*(2^(m)));
    m = m+1;
end
for j = 10:32;
    l = l+(x(j)*(2^(a)));
    a = a-1;
end
IEEE = -1^sign*(1+l)*2^(k-127)
```

Outputnya berupa:

```
Command Window
Masukkan nilai biner (32) = [0 1 0 0 0 0 1 0 0 1 1 0 1 1 0 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1]
IEEE = -59.169
```

Program untuk mengonversi bilangan biner ke desimal berdasarkan IBM.

```
clear; clc;
x = input('masukkan nilai biner (32) = ');
% x = [0 1 0 0 0 0 1 0 0 1 1 0 1 1 0 0 1 0 1 0 1 1 0 1 0 0 0 1 0 1 0 1];
% IBM
d1 = x(1);
sign = (-1)^(d1);
m = 0; k = 0;
a = -1; l = 0;
for i = 8:-1:2;
    k = k+(x(i)*(2^(m)));
    m = m+1;
end
for j = 9:32;
    l = l+(x(j)*(2^(a)));
    a = a-1;
end
k
l
IBM = sign*l*16^(k-64)
```

Outputnya berupa:

```
k = 66
l = 0.42452
IBM = 108.68
```

Program untuk mengonversi bilangan biner ke desimal berdasarkan IEEE dan IBM bisa dilihat di lampiran dan menghasilkan output sebagai berikut:

```
Command Window
< [0 1 0 0 0 0 1 0 0 1 1 0 1 1 0 0 1 0 1 0 1 1 0 1 0 0 0 1 0 1 0 1];
1. IBM
2. IEEE
Masukkan pilihan konversi: 2
n = 2
IEEE = -59.169
```

4. DISKUSI

Bilangan desimal 32-bit dapat diperoleh dengan menggunakan konversi dengan metode IBM ataupun IEEE. Pada dunia industri khususnya minyak dan gas, biasanya disimpan dalam format IBM. Namun beberapa pihak menyimpannya dalam bentuk IEEE. Perbedaan konversi dan format inilah yang dapat membedakan hasil yang didapat.

Pada percobaan ini kami telah menyusun berkali-kali program dan kami menemukan bahwa pada tahap pertama program kami mengalami *error system*. Kesalahan tersebut terjadi karena proses input data biner yang kami masukkan tidak disertai dengan adanya tutup kurung siku. Tutup kurung siku tersebut kami maksudkan untuk menunjukkan kumpulan komponen setiap data berupa 0 dan 1 atau komponen dalam bentuk matriks.

Pada percobaan berikutnya kami juga menemukan masalah dalam deklarasi setiap variabel. Masalah utama terjadi ketika kami akan menentukan rumus/formula dalam konversi ke desimalnya. Karena penentuan variabel sangat menentukan bentuk formulanya.

Sehingga secara bertahap proses pemograman telah terstruktur dan mendapatkan hasil yang sama dengan teori kami yang bersumber dari Bapak Agus ditambah dengan perhitungan sendiri dengan sistem operasional yang sama. Kesamaan hasil ini yang meyakinkan kami bahwa program kami memang sudah benar.

5. KESIMPULAN

Konversi bilangan data biner ke desimal sangat penting dalam penerjemahan data antara komputer dengan pengguna komputer. Semakin banyak jumlah bit yang kita temukan semakin sulit juga kita menentukan maksud dari komputer, sementara data yang kita butuhkan ialah data yang berupa angka desimal agar mudah kita pahami. Disisi lain, semakin banyaknya data bit yang kita butuhkan memungkinkan keakuratan data tidak begitu tepat.

Kami telah berhasil membuat program yang dapat mengkonversi bilangan biner 32-bit menjadi desimal dengan format IBM dan IEEE. Dari percobaan-percobaan yang telah kami lakukan kami mendapat satu masalah, bagaimana bila jumlah bitnya sampai dengan 256-bit atau lebih. Jadi sangat penting bagi kita untuk memahami konversi biner ke desimal karena data yang disampaikan komputer biasanya masih dalam data yang biner. Keunggulan dari program yang kami buat adalah *scriptnya* yang pendek serta mudah untuk dipahami dan terstruktur apabila akan dikembangkan ke depannya karena sistem konversi ini sangat penting dalam pemograman.

UCAPAN TERIMA KASIH

Kami mengucapkan terima kasih kepada Pak Agus Abdullah, Pak Heri dan Pak Hardi yang telah membantu dalam pembuatan *script* pemograman kami. Apabila tidak ada ikut campur tangan Bapak-Bapak ini, program ini mungkin akan selesai dalam jangka yang cukup panjang karena keterbatasan kemampuan kami.

REFERENSI

1. A. Abdullah. 2013. *IBM versus IEEE*. URL <http://ensiklopediseismik.blogspot.co.id/2013/11/ibm-versus-ieee.html> [2017/11/10]
2. H.R. Prabusetyo dan M. Heriyanto. 2017. *Modul Praktikum GP2013 Metode Komputasi*. Universitas Pertamina: Jakarta. doi:10.17605/OSF.IO/5TFVQ
3. G. Feoh. 2011. Sistem Bilangan dan Konversi Bilangan. URL <https://id.scribd.com/doc/259294974/Sistem-Bilangan-Dan-Konversi-Bilangan> [2017/11/10]
4. A. Pribadi. *Presentasi tentang Sistem Bilangan dan Konversinya*. URL <http://27afri-file.weebly.com/uploads/1/3/0/7/13077226/psd01.pdf> [2017/11/10]
5. R. Sitorus. 2011. *System bilangan biner, octal, decimal dan hexsaa decimal*. URL <https://rusonadisitorus.wordpress.com/2011/04/04/system-bilangan-biner-octal-decimal-dan-hexsaa-decimal/> [2011/04/04]

LAMPIRAN**Program konversi biner ke desimal dalam format IBM atau IEEE:**

```

x = input('masukkan nilai biner (32) = ');

% example
% x = [0 1 0 0 0 0 1 0 0 1 1 0 1 1 0 0 1 0 1 0 1 1 1 0 1 0 0 0 1 0 1 0 1];

d1 = x(1);
sign = (-1)^(d1);
m = 0;
k = 0;
a = -1;
l = 0;
disp('1. IBM')
disp('2. IEEE')
n = input('Masukkan pilihan konversi: ');

if n == 1
    for i = 8:-1:2;
        k = k+(x(i)*(2^(m)));
        m = m+1;
    end
    for j = 9:32
        l = l+(x(j)*(2^(a)));
        a = a-1;
    end
    %% IBM
    IBM = sign*1*16^(k-64)

elseif n == 2
    for i = 9:-1:2;
        k = k+(x(i)*(2^(m)));
        m = m+1;
    end
    for j = 10:32
        l = l+(x(j)*(2^(a)));
        a = a-1;
    end
    %% IEEE
    IEEE = -1^sign*(1+l)*2^(k-127)

else
    disp('Nilai yang anda masukkan salah')
end

```