# Pengenalan Representasi Bilangan dan Evaluasi Fungsi Matematika dalam Komputer - Edisi Spreadsheet

Saifuddin Arief
Saifuddin.Arief@rocketmail.com



Komputer merupakan alat hitung yang sangat canggih, dapat diandalkan serta dapat memberikan hasil perhitungan yang akurat. Pada sebagian besar kasus, perhitungan dengan program komputer akan menghasilkan hasil perhitungan yang mendekati akurat dengan kesalahan yang sangat kecil dan hampir dapat diabaikan atau tidak terlihat.

Saat ini kebanyakan software matematika dan spreadsheet menggunakan sistem presisi ganda (*double precision*) untuk merepresentasikan bilangan dalam komputer serta hasil operasi arimatika terhadap bilangan tersebut. Sistem bilangan presisi ganda mempunyai akurasi sekitar 16 digit desimal signifikan. Selain bilangan nol, bilangan-bilangan yang dapat disimpan dalam sistem presisi ganda nilainya kira-kira berada dalam jangkauan dari $\pm\ 2.225\times10^{-308}$ sampai $\pm\ 1.797\times10^{308}$.

Berikut ini adalah ilustrasi mengenai jangkauan bilangan yang dalam sistem presisi ganda dengan program spreadsheet:

```
   2^1023 =  8.988466E+307
   2^1024 =  #NUM!
 1/2^1023 =  0.111254E-307
 1/2^1024 =  #NUM!
```

Simbol `#NUM!` pada hasil perhitungan ini merepresentasikan hasil perhitungan yang nilainya di luar jangkuan nilai yang dapat disimpan oleh program komputer. Apabila suatu bilangan atau hasil perhitungan mempunyai nilai yang lebih besar daripada yang dapat disimpan oleh komputer maka disebut *overflow* dan jika lebih kecil maka disebut *underflow*.

Di dalam spreadsheet tidak variabel khusus yang menyatakan bilangan terkecil dan terbesar yang dapat direpresentasikan dalam komputer bilangan. Kedua bilangan tersebut dapat dihitung sebagai berikut:

```
realmin = 1/2^1023*(2-eps) = 2.225E-308
realmax =   2^1023*(2-eps) = 1.79887E+308
```

di mana `realmin` dan `realmax` masing-masing merepresentasikan bilangan terkecil dan terbesar yang dapat disimpan oleh spreadsheet kemudian `eps` adalah variabel yang merepresentasikan jarak antara angka 1 dengan angka berikutnya. Variabel ini dijelaskan pada paragraf selanjutnya.

Selain hanya dapat menyimpan dalam range tertentu, representasi bilangan komputer juga tidak kontinu melainkan bersifat diskrit di mana suatu terdapat jarak antara bilangan-bilangan yang dapat disimpan dalam komputer. Jarak tersebut tergantung pada nilai bilangan tersebut semakin besar bilangannya maka semakin jauh jarak antara bilangan berikutnya. Dalam sistem bilangan presisi ganda, jarak standar antara bilangan 1 dengan bilangan berikutnya adalah sekitar `2.2204e-16`. Seperti halnya `realmin` dan realmax di dalam spreadsheet juga tidak ada informasi khusus mengenai jarak tersebut. Anggap `eps` adalah variabel yang maka menggambarkan jarak antara angka 1 dengan angka berikutnya maka nilai dari variabel tersebut dapat dihitung sebagai berikut:

| n | x | 1 + x | check |
|---|---|---|---|
| 45 | 2.842170943040400E-14 | 1.00000000000003000E+00 | - |
| 46 | 1.421085471520200E-14 | 1.00000000000001000E+00 | - |
| 47 | 7.105427357601000E-15 | 1.00000000000001000E+00 | - |
| 48 | 3.552713678800500E-15 | 1.00000000000000000E+00 | - |
| 49 | 1.776356839400250E-15 | 1.00000000000000000E+00 | eps/2 |
| eps = | 3.552713678800500E-15 | | |

Terlihat bahwa nilai variabel `eps` pada spreadsheet adalah sedikit lebih besar dari pada standar nilai `eps` pada sistem presisi ganda.

Hanya sebagian kecil bilangan dapat direpresentasikan secara eksak dalam komputer dan sebagian besar bilangan hanya direpresentasikan secara aproksimasi sehingga dapat menyebabkan terjadi suatu kesalahan perhitungan. Pada sebagian besar kasus, kesalahan tersebut sangat kecil dan dapat diabaikan. Sebagai ilustrasi, berikut ini adalah sebuah perhitungan yang terlihat eksak namun sebenarnya mempunyai kesalahan yang sangat kecil sekali. Hasil perhitungan dari operasi 34.2 – 33.3 dengan menggunakan spreadsheet dalam format *default* adalah sebagai berikut:

```
    a =     34.2
    b =     33.3
a - b =      0.9
```

Namun jika digunakan format output dengan jumlah desimal 15 digit maka outputnya adalah:

```
    a =   34.2000000000000000
    b =   33.3000000000000000
a - b =    0.9000000000000060
```

Kesalahan yang terjadi pada contoh ini nilainya sangat kecil sekali dan dapat diabaikan namun nilai tersebut tetap berbeda dengan jawabannya eksaknya yaitu 0.9. Kesalahan ini terjadi karena bilangan 34.2 dan 33.3 jika dinyatakan dalam sistem bilangan biner akan mempunyai representasi bilangan dalam jumlah tak terbatas, seperti di bawah ini

```
34.2 = 100010.00110011001100110011001100110011001100 ...
33.3 = 100001.01001100110011001100110011001100110011 ...
```

Oleh karena itu dilakukan pembulatan pada representasi kedua bilangan tersebut dalam sistem biner bilangan komputer sehingga terjadi error pembulatan pada penyimpanan kedua bilangan tersebut. Hal ini secara otomatis mengakibatkan hasil operasi terhadap kedua tersebut menjadi tidak eksak.

Aritmatika pada komputer pada dasarnya hanya dilakukan dengan operasi aritmatika dasar penjumlahan, pengurangan, perkalian dan pembagian. Operasi-operasi aritmatika lainnya seperti pemangkatan, akar serta fungsi-fungsi matematika pada umumnya akan dilakukan dengan suatu algoritma tertentu di mana operasi-operasi di dalamnya merupakan kombinasi dari hanya operasi-operasi aritmatika dasar sehingga hasilnya tidak eksak. Sebagai ilustrasi, perhitungan `cos(pi/2)` dengan spreadsheet dalam output dengan format scientific adalah sebagai berikut:

```
cos(pi()/2) =    6.12323E-17
```

Terlihat bahwa hasilnya berbeda dengan nilai eksak dari $\cos(\pi/2) = 0$. Hal ini karena fungsi cosinus dalam komputer diaproksimasi dengan menggunakan deret Taylor.

Akurasi dan rentang nilai yang disediakan oleh sistem presisi ganda sudah mencukupi untuk hampir semua kasus perhitungan praktis dalam muncul komputasi matematika, sains dan rekayasa. Perhatian khusus biasanya hanya perlu diberikan untuk perhitungan-perhitungan tingkat lanjut, seperti penyelesaian persamaan diferensial. Penjelesan lebih lanjut dan lebih komprehensif dapat dilihat pada beberapa buku analisis numerik.