Pengenalan Representasi Bilangan dan Evaluasi Fungsi Matematika dalam Komputer - Edisi Scilab

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 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 $\pm\ 2.22511\times10^{-308}$ sampai $\pm\ 1.79717\times10^{308}$.

Berikut ini adalah ilustrasi mengenai jangkauan bilangan yang dalam sistem presisi ganda dengan program Scilab.

```
--> 2^1023
 ans  =

   8.99D+307

--> 2^1024
 ans  =

   Inf

--> 1/2^1023
 ans  =

   1.11D-308
```

```
--> 1/2^1024
 ans  =

   0.
```

Pada hasil perhitungan ini, simbol `Inf` merepresentasikan hasil perhitungan yang nilainya di luar jangkauan yang dapat disimpan oleh sistem presisi ganda. Kemudian untuk perhitungan yang terakhir hasilnya lebih kecil daripada nilai yang dapat disimpan dalam sistem presisi ganda sehingga dinyatakan dengan angka nol.

Operasi-operasi aritmatika di atas jika dilakukan dengan spreadsheet hasil perhitungannya adalah sebagai berikut:

```
   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.

Contoh-contoh perhitungan di atas merepresentasikan jangkauan nilai yang dapat dilakukan dengan komputer. Untuk nilai-nilai yang tidak dapat disimpan oleh sistem bilangan yang digunakan dalam komputer akan dinyatakan dalam simbol khusus, yaitu `Inf` dan `0` untuk hasil perhitungan yang *overflow* dan *underflow* pada Scilab dan software lainnya yang sejenis seperti Octave dan Euler Math Toolbox. Selanjutnya #NUM merepresentasikan hasil perhitungan yang *overflow* atau *underflow* pada program-program spreadsheet seperti LibreOffice Calc, Microsoft Excel.

Di dalam Scilab, bilangan terkecil dan terbesar yang dapat direpresentasikan dalam komputer dinyatakan dalam fungsi `number_properties`. Kedua nilai tersebut dapat diperoleh dengan menjalankan perintah `number_properties("huge")` dan `number_properties("tiny")`, seperti di bawah ini.

```
--> number_properties("huge")
 ans  =

   1.80D+308

--> number_properties("tiny")
 ans  =

   2.23D-308
```

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 antara bilangan 1 dengan bilangan berikutnya adalah sekitar `2.2204e-16`, bilangan tersebut dalam Scilab direpresentasikan dalam variabel khusus `%eps`. Berikut ini ilustrasi variabel `%eps` dengan Scilab.

```
    %eps    = 2.22045e-16
    %eps    = 0.00000000000000022045
 x + %eps   = 1.00000000000000022045
 x + %eps/2 = 1
 x + 2*%eps = 1.00000000000000044089
```

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 Scilab dalam format *default* adalah sebagai berikut:

```
--> 34.2 - 33.3
 ans  =

   0.9000000
```

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

```
--> format('v',18)

--> 34.2 - 33.3
 ans  =

   0.900000000000006
```

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 Scilab akan memberikan output sebagai berikut:

```
--> cos(%pi/2)
 ans  =

   6.123D-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.