1 Galat dalam Komputasi Numerik

Di dalam praktek sehari-hari, misalnya dalam bidang teknik dan bisnis, sering terdapat kasus gagalnya pencarian penyelesaian eksak suatu masalah matematika. Hal ini utamanya bukan disebabkan oleh cara mencari penyelesaian yang tidak diketahui, namun karena adanya kenyataan bahwa penyelesaian yang diinginkan tidak dapat dinyatakan secara elementer atau adanya fungsi-fungsi lain yang sudah diketahui. Oleh karena itu komputasi numerik menjadi sangat penting, khususnya dalam kaitannya dengan meningkatnya peranan metode-metode matematika dalam berbagai bidang sains dan teknologi serta hadirnya teknologi pendukung berupa komputer berkemampuan tinggi.

Pengertian komputasi numerik dan metode numerik

Komputasi numerik merupakan suatu pendekatan penyelesaian masalah matematika dengan menggunakan beberapa metode numerik. Metode numerik adalah suatu metode untuk menyelesaikan masalah-masalah matematika dengan menggunakan sekumpulan operasi aritmetika sederhana dan operasi logika pada sekumpulan bilangan atau data numerik yang diberikan. Operasi-operasi tersebut biasanya merupakan operasi-operasi yang dapat dilakukan oleh komputer. Metode komputasi yang digunakan disebut **algoritma**. Tergantung pada kekompleksan masalah yang harus diselesaikan, tingkat keakuratan yang diinginkan, metode yang dipakai, dan seterusnya, proses penyelesaian mungkin memerlukan beberapa puluh sampai jutaan operasi. Apabila banyaknya operasi hitung yang diperlukan hanya beberapa puluh, maka seseorang dapat menyelesaikan masalahnya secara manual atau menggunakan kalkulator. Akan tetapi jika penyelesaian suatu masalah memerlukan jutaan operasi hitung, maka pemakaian komputer berkecepatan tinggi merupakan kebutuhan yang tidak dapat dihindari. Di sinilah kemajuan teknologi komputer memegang peranan penting dalam komputasi numerik. Meskipun demikian, pemilihan metode yang efisien (memerlukan sesedikit mungkin operasi hitung) merupakan aspek lain yang menjadi perhatian dalam komputasi numerik. Hal ini akan semakin terasa di dalam menyelesaikan masalah-masalah berskala besar, yang melibatkan ribuan variabel misalnya.

CONTOH 1.1.
Hitunglah $\sqrt{2}$ sampai empat angka desimal.

Penyelesaian:
Terdapat lebih daripada satu algoritma, yang hanya menggunakan empat operasi aritmetika dasar (penjumlahan/pengurangan dan perkalian/pembagian). Salah satunya, yang cukup populer, adalah

Suatu algoritma untuk menghitung $\sqrt{2}$ dengan hanya menggunakan operasi perkalian, pembagian dan penjumlahan

$$x_1 = 1, \qquad x_n = \frac{1}{2}\left(x_{n-1} + \frac{2}{x_{n-1}}\right), \quad \textit{untuk} \quad n = 2,\ 3,\ 4,\ \ldots.$$

Dengan menggunakan algoritma di atas kita peroleh, untuk $n = 2,\ 3,\ 4,\ 5,$

$$x_2 = \frac{3}{2}, \qquad x_3 = \frac{17}{12}, \qquad x_4 = \frac{1}{2}\left(\frac{17}{12} + \frac{24}{17}\right) = \frac{577}{408},$$

$$x_5 = \frac{1}{2}\left(\frac{577}{408} + \frac{816}{577}\right) = \frac{665857}{470832}$$

atau, dalam bentuk pecahan desimal

$$x_2 = 1,5000000 \qquad x_3 = 1.4166667 \qquad x_4 = 1.4142157 \qquad x_5 = 1.4142136.$$

Jadi, hampiran sampai empat angka desimal untuk $\sqrt{2}$ adalah 1.4142. □

Berikut adalah contoh pemakaian komputer (dalam hal ini dengan menggunakan program MATLAB) untuk menyelesaikan masalah di atas. Anda dapat mengubah batas nilai e untuk mendapatkan tingkat keakuratan yang diinginkan.

Program MATLAB untuk menghitung hampiran $\sqrt{2}$

```
>>x=1;e=1;
>>while e>0.00001,
y=x;x=(y+2/y)/2
e=abs(x-y);
end
 x  =
     1.5
 x  =
     1.4166667
```

rapa parameter masukan. Efek galat data awal dapat diestimasikan dengan menggunakan cara-cara sederhana, misalnya dengan menggunakan variasi data awal dalam batas-batas galatnya dan dengan menetapkan penyelesaiannya. Apabila terdapat beberapa data awal yang memiliki galat yang sifatnya alami, maka pemakaian metode statistika akan bermanfaat. Dalam beberapa kasus, galat bawaan dapat dianggap sebagai galat suatu fungsi yang diakibatkan oleh galat argumen (masukannya).

Dalam kebanyakan kasus, metode-metode numerik merupakan hampiran, sehingga sekalipun data awalnya tidak mengandung galat dan semua operasi aritmetika dilakukan secara ideal, metode-metode tersebut menghasilkan penyelesaian masalah semula yang memuat beberapa galat yang disebut **galat metode** (yang dipakai). Hal ini disebabkan karena suatu metode numerik biasanya digunakan untuk menyelesaikan beberapa masalah lain, yang lebih sederhana, sebagai hampiran masalah asli. Dalam sejumlah kasus, metode numerik yang dipilih disusun berdasarkan pada proses tak berhingga, yang limitnya menuju penyelesaian yang diinginkan.[1] Akan tetapi, dalam kenyataannya tidak mungkin melakukan semua proses tersebut, sehingga prosesnya harus dihentikan pada langkah tertentu dan hasilnya adalah suatu hampiran penyelesaian.

Suatu metode numerik biasanya tergantung pada beberapa parameter yang dapat dikendalikan. Beberapa contoh parameter demikian adalah banyaknya iterasi di dalam menyelesaikan sistem persamaan dan banyaknya suku yang harus dihitung di dalam menjumlahkan suatu deret, dan lebar interval yang digunakan untuk menghitung hampiran suatu integral tentu. Galat suatu metode numerik atau estimasinya biasanya tergantung pada parameter yang sesuai. Estimasi galat yang diperoleh mungkin dinyatakan dalam kuantintas-kuantitas yang diketahui. Dengan estimasi galat ini, nilai-nilai parameter yang menentukan galat metode tersebut berada dalam batas-batas yang diinginkan dapat ditentukan. Akan tetapi, dalam kebanyakan kasus estimasi galat memuat pengali-pengali konstanta yang tidak diketahui nilainya, dan parameter metode berbentuk suatu fungsi pangkat atau fungsi eksponensial. Dengan estimasi galat demikian laju penurunan galat dapat diatur dengan mengubah parameter metode. Laju penurunan galat merupakan karak-

[1]Sebagai contoh adalah perhitungan nilai suatu fungsi dengan menggunakan beberapa suku pertama deret tak berhingga, seperti $\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots$, hanya dihitung n suku pertama, sisanya diabaikan.

teristik penting suatu metode numerik.

Galat yang paling rumit di dalam komputasi numerik adalah galat pembulatan (*round-off errors*), yang diperoleh selama pemakaian operasi-operasi aritmetika. Apabila banyaknya operasi yang dilakukan tidak besar, maka galat-galat pembulatan dalam kalkulasi manual dapat dihitung dengan rumus-rumus perambatan galat yang akan dibahas di belakang. Terdapat dua situasi yang mungkin terjadi dalam penyelesaian suatu masalah numerik dengan komputer. Pertama, jika banyaknya operasi aritmetika yang dilakukan sedikit, maka galat-galat pembulatan mungkin dapat diabaikan, karena komputer menggunakan sepuluh atau lebih angka desimal signifikan, sementara hasil akhir biasanya diambil sampai lima angka signifikan. Kedua, jika masalah yang harus diselesaikan cukup rumit (misalnya masalah persamaan diferensial parsial), dan proses perhitungan hampiran penyelesaian yang diinginkan memerlukan, katakan 10^{10} operasi aritmetika, maka tidaklah realistik dalam hal ini untuk menghitung efek galat pembulatan pada setiap operasi. Dalam kasus seperti ini galat pembulatan dapat dikatakan bersifat acak. Namun, bagaimanapun juga galat pembulatan tidak dapat diabaikan dalam menyelesaikan masalah-masalah numerik yang rumit.

Suatu masalah numerik mungkin dapat diselesaikan dengan beberapa metode hampiran yang berbeda. Sensitivitas galat pembulatan pada dasarnya tergantung pada metode numerik yang dipilih. Suatu metode numerik dianggap berhasil jika galat yang diberikan merupakan pecahan dari galat bawaan, dan galat karena pembulatan, disebut **galat komputasi**, lebih kecil daripada galat metode. Apabila tidak ada galat bawaan, maka galat metode haruslah kurang daripada tingkat keakuratan yang diberikan.

Selain persyaratan tingkat keakuratan, metode numerik yang dipilih juga harus memenuhi sejumlah persyaratan lain. Utamanya, metode tersebut menggunakan seminimum mungkin operasi, memerlukan lebih sedikit unit penyimpanan (memori) pada komputer, dan akhirnya, secara logika lebih sederhana, sehingga lebih cepat dijalankan oleh komputer. Sejumlah syarat yang diberikan mungkin saling menjadi kendala, sehingga pemilihan metode numerik boleh jadi memerlukan suatu kompromi. Suatu algoritma yang menghasilkan galat kumulatif yang terbatas, sehingga hampiran yang diperoleh memenuhi tingkat keakuratan tertentu, disebut algoritma **stabil**. Algoritma yang menghasilkan galat kumulatif yang merusak hampiran penyelesaian yang diperoleh, sehingga

Dalam sistem biner, basis perpangkatan adalah 2, sebagai pengganti basis 10 dalam sistem desimal. Jadi, bilangan 1454 dalam sistem biner dinyatakan sebagai

$$\begin{aligned} 1454 &= 1024 + 256 + 128 + 32 + 8 + 4 + 2 \\ &= 1 \times 2^{10} + 1 \times 2^8 + 1 \times 2^7 + 1 \times 1 \times 2^5 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 \\ &= 10110101110_{dua} \end{aligned}$$

Catatan:

- Indeks "dua" atau "2" digunakan untuk membedakan dengan penulisan sistem desimal, sehingga 110 berarti seratus sepuluh, sedangkan 110_{dua} dan 110_2 berarti enam.

- Fungsi MATLAB `dec2bin` dapat digunakan untuk mendapatkan representasi dalam sistem biner suatu bilangan bulat positif N.

```
>>dec2bin(1454)
ans =
    10110101110
>>dec2bin(1563)
ans =
    11000011011
```

Secara umum, jika N bilangan bulat positif yang dapat dinyatakan dalam ekspansi berbasis pangkat dua sebagai

Nilai tempat dalam sistem biner

$$N = b_n \times 2^n + b_{n-1} \times 2^{n-1} + ... + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0,$$

dengan $b_k \in \{0,\ 1\}$, maka dalam sistem biner N dapat dinyatakan sebagai

$$N = b_n b_{n-1} ... b_2 b_1 b_0{}_{dua}.$$

Jika diketahui bilangan bulat positif N (dalam bentuk desimal), maka bentuk binernya dapat dicari dengan menggunakan algoritma sebagai berikut

$$\begin{aligned} N &= 2q_0 + b_0 \\ q_0 &= 2q_1 + b_1 \\ q_1 &= 2q_2 + b_2 \\ &\vdots \\ q_{n-2} &= 2q_{n-1} + b_{n-1} \\ q_{n-1} &= 2q_n + b_n, \qquad (q_n = 0), \end{aligned} \tag{1.1}$$

Algoritma untuk mengubah bilangan desimal ke bilangan biner

dengan $b_k \in \{0,\ 1\}$. Selanjutnya, dalam sistem biner N dapat dinyatakan sebagai

$$N = b_n b_{n-1} ... b_2 b_1 b_{0\,dua}.$$

CONTOH 1.2.
Dengan menggunakan algoritma 1.1, diterapkan pada $N = 1454$, didapatkan

$$\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl} 1454 &=& 2\times 727+0 & 90 &=& 2\times 45+0 & 5 &=& 2\times 2+1 \\ 727 &=& 2\times 363+1 & 45 &=& 2\times 22+1 & 2 &=& 2\times 1+0 \\ 363 &=& 2\times 181+1 & 22 &=& 2\times 11+0 & 1 &=& 2\times 0+1, \\ 181 &=& 2\times 90+1 & 11 &=& 2\times 5+1 & && \end{array}$$

sehingga $1454 = 10110101110_{dua}$, seperti contoh sebelumnya. □

1.2.2 Sistem Heksadesimal

Dalam sistem heksadesimal digunakan basis perpangkatan 16 dan semua bilangan dinyatakan dengan menggunakan maksimum 16 digit yang berbeda, yang biasanya dinyatakan sebagai

$$0,\ 1,\ 2,\ ...,\ 9,\ A,\ B,\ C,\ D,\ E,\ F,$$

dengan $A_{16} = 10$, $B_{16} = 11$, $C_{16} = 12$, $D_{16} = 13$, $E_{16} = 14$, dan $F_{16} = 15$. Sebagai contoh,

$$14CA_{16} = 1\times 16^3 + 4\times 16^2 + (12)\times 16^1 + (10)\times 16^0 = 5322.$$

Algoritma untuk mengubah bilangan desimal ke bilangan heksadesimal analog dengan algoritma untuk mengubah bilangan desimal ke bilangan biner. Dalam hal ini digunakan pembagi 16.

Sistem heksadesimal memiliki hubungan erat dengan sistem biner. Konversi bilangan biner ke bilangan heksadesimal dan sebaliknya dapat dilakukan secara mudah. Untuk mengkonversi bilangan heksadesimal ke bilangan biner, ganti setiap digit dengan representasi binernya. Sebagai contoh,

$$2AC_{16} = 1010101100_{dua},$$

karena

$$2_{16} = 10_{dua}, \quad A_{16} = 10 = 1010_{dua}, \quad C_{16} = 12 = 1100_{dua}.$$

Sebaliknya, untuk mengubah bilangan biner ke bilangan heksadesimal, kelompokkan setiap empat bit dari kanan ke kiri, dan ganti setiap empat bit tersebut dengan nilai heksadesimalnya. Sebagai contoh,

$$1101001101_{dua} = 34D_{16},$$

karena

$$1101_{dua} = 13 = D_{16}, \quad 0100_{dua} = 4 = 4_{16}, \quad 11_{dua} = 3 = 3_{16}.$$

1.2.3 Bilangan Pecahan dan Deret

Dalam komputasi numerik, nilai-nilai pecahan dinyatakan dalam bentuk desimal. Setiap bilangan pecahan rasional p/q, $q \neq 0$, dinyatakan sebagai pecahan desimal yang terdiri atas berhingga digit atau tak berhingga digit berulang. Berikut adalah beberapa contoh

$$\begin{aligned}
1/2 &= 0.50000000... = 0.5 \\
1/3 &= 0.333333... = 0.33333\overline{3} \\
1/7 &= 0.142857142857... = 0.142857\overline{142857}
\end{aligned}$$

Ekspansi desimal pecahan $1/3$ memuat tak berhingga digit 3 secara berulang, sedangkan ekspansi desimal pecahan $1/7$ memuat beberapa digit yang diulang tak berhingga kali, yakni "142857". Apabila ekspansi desimal ditulis hanya sampai beberapa digit berhingga, maka digit-digit terakhir yang berulang diberi garis atas. Dalam contoh ekspansi $1/3$ di atas $\overline{3}$ artinya digit 3 berulang tak berhingga kali, sedangkan pada $1/7$, $\overline{142857}$ artinya 142857 berulang tak berhingga kali. Notasi tersebut berlaku juga untuk ekspansi pecahan biner (akan dijelaskan nanti).

Dalam praktek kita hanya mengambil beberapa digit untuk meng-

1. *Perhatikan bahwa* $\frac{53}{64} = 1/2 + 1/4 + 1/16 + 1/64$.
   Oleh karena itu, dalam sistem biner $\frac{53}{64} = 0.110101_{dua}$.

2. *Perhatikan deret geometri yang konvergen ke nilai* $1/2$*:*

$$1/4 + 1/8 + 1/16 + 1/32 + 1/64 + ... = 1/2.$$

   Jadi, dalam sistem biner 0.5 *dinyatakan sebagai*

$$0.5 = 0.111111..._{dua} = 0.11111\overline{1}_{dua} = 0.1_{dua}. \quad \square$$

Dari contoh di atas, ternyata pecahan biner juga dapat memuat tak berhingga digit berulang. Cara menyingkat digit-digit yang berulang sama dengan cara penulisan pada sistem desimal, yakni dengan memberi garis di atas digit-digit yang berulang.

Algoritma berikut dapat digunakan untuk mencari penyajian biner suatu pecahan $0 < R < 1$.

Algoritma untuk mengubah pecahan desimal ke pecahan biner

$$\begin{array}{rclll}
2R & = & b_1 + F_1 & b_1 = int(2R) & F_1 = frac(2R) \\
2F_1 & = & b_2 + F_2 & b_2 = int(2F_1) & F_2 = frac(2F_1) \\
2F_2 & = & b_3 + F_3 & b_3 = int(2F_2) & F_3 = frac(2F_2) \\
2F_3 & = & b_4 + F_4 & b_4 = int(2F_3) & F_4 = frac(2F_3) \\
2F_4 & = & b_5 + F_5 & b_5 = int(2F_4) & F_5 = frac(2F_4) \\
 & \vdots & & \vdots & \vdots \\
2F_{n-1} & = & b_n + F_n & b_n = int(2F_{n-1}) & F_n = frac(2F_{n-1}) \\
 & \vdots & & \vdots & \vdots
\end{array} \tag{1.2}$$

dengan $int(x)$ adalah bagian bulat x dan $frac(x)$ adalah bagian pecahan x. Proses tersebut mungkin berhenti setelah langkah ke-n (jika didapatkan $F_n = 0$), mungkin berlanjut terus. Selanjutnya, representasi biner R adalah

$$R = 0.b_1b_2b_3...b_n..._{dua}.$$

CONTOH 1.4.
Nyatakan pecahan $\frac{53}{64}$ *dalam bentuk pecahan biner!*

biner dengan perpangkatan 2 akan menggeser titik (koma) pemisah bagian bulat dan bagian pecahan. Misalnya,

$$\begin{aligned}\frac{53}{64} &= 0.110101_{dua}\\ 2^6 \times \frac{53}{64} &= 110101_{dua}.\end{aligned}$$

Aturan pergeseran titik tersebut benar, karena memang $53 = 110101_{dua}$. (Silakan diperiksa!)

1.2.5 Notasi Ilmiah (*Scientific Notation*)

Cara baku untuk menyajikan bilangan riil, disebut **notasi ilmiah** (*scientific notation*), dapat dinyatakan dalam bentuk

Notasi Ilmiah (Scientific Notation)

$$\pm q \times 10^n,$$

dengan $1 \leq q < 10$. Bilangan q disebut **mantis** dan n disebut **eksponen**. Berikut adalah contoh-contoh penyajian bilangan dengan notasi ilmiah.

1. $0.000342 = 3.42 \times 10^{-4}$
2. $34.4108 = 3.44108 \times 10$
3. $9800000 = 9.8 \times 10^6$
4. $A = 6.02252 \times 10^{23}$ (bilangan Avogadro dalam kimia)
5. $1K = 1.024 \times 10^3$ (pengertian kilo dalam ilmu komputer)
6. $\epsilon_0 = 8.8542 \times 10^{-12}$ (konstanta dielektrik)
7. $\mu_0 = 1.2566 \times 10^{-6}$ (permiabilitas).

1.2.6 Titik-Mengambang Normal (*Normalized Floating-Point*)

Misalkan x adalah suatu bilangan desimal bukan nol. Kita dapat menyatakan x dalam bentuk

$$x = \sigma.m.10^p, \tag{1.3}$$

dengan $\sigma = +1$ atau $\sigma = -1$, dan $0.1 \leq m < 1$. Besaran σ, m, dan p berturut-turut disebut **tanda**, **mantis**, dan **eksponen** atau pangkat. Sebagai contoh,

$$13.642 = 0.13642 \times 10^2.$$

bit pangkat, bernilai $-975 \leq p \leq 1071$, dan sebuah bit tanda. Skema penyajian titik-mengambang ini dijelaskan pada Gambar 1.1.

$$\underbrace{\boxed{\sigma}}_{Bit\ 1}\underbrace{\boxed{\textit{pangkat } (p)}}_{Bit\ 2-12}\underbrace{\boxed{\textit{mantis } (m)}}_{Bit\ 13-60}$$

Gambar 1.1: Alokasi bit tanda (σ), mantis (m) dan pangkat (p) pada komputer CDC dengan 60-bit titik-mengambang

2. *Komputer DEC VAX menggunakan penyajian titik-mengambang 32-bit, seperti skema pada Gambar 1.2. Mantis terdiri atas 24 bit (bernilai $\frac{1}{2} \leq m < 1$), pangkat terdiri atas 8 bit (bernilai $-127 \leq p \leq 127$), dan satu bit tanda. Oleh karena $\frac{1}{2} \leq m < 1$, maka*

$$m = 0.1b_2b_3...b_{24\,dua}.$$

Bit pertama 1 tidak disimpan secara eksplisit di dalam memori, hanya bit b_2, b_3, ..., dan b_{24} yang disimpan di dalam memori. Bit pertama 1 selalu disisipkan ke dalam penyajian tersebut setiap kali dilakukan operasi hitung yang melibatkan m.

$$\underbrace{\boxed{\sigma}}_{Bit\ 1}\underbrace{\boxed{\textit{pangkat } (p)}}_{Bit\ 2-9}\underbrace{\boxed{\textit{mantis } (m)}}_{Bit\ 10-32}$$

Gambar 1.2: Alokasi bit tanda (σ), mantis (m) dan pangkat (p) pada komputer DEC VAX dengan 32-bit titik-mengambang

3. *Pada koprosesor aritmetika yang digunakan pada komputer-komputer mikro, misalnya keluarga Intel 80X87 dan Motorola 6888X, digunakan* **Aritmetika Titik-Mengambang IEEE Baku**. *Dalam standard ini, yang merupakan notasi IEEE baku presisi tunggal, mantis m dinormalkan sehingga memenuhi $1 \leq m < 2$, menggunakan 24 bit, sebuah bit disembunyikan seperti pada DEC VAX. Pangkat p memiliki nilai-nilai $-126 \leq p \leq 127 = 2^8 - 1$. Dalam sistem ini terdapat penyajian untuk ∞, misalnya untuk menyatakan 1/0, dan* **nan**, *yang berarti* **bukan bilangan** *(not a number), misalnya untuk menyatakan 0/0.* □

but adalah 0111111110111_{dua}, yakni bilangan yang memiliki mantis terbesar ($= 0.11111111_{dua}$) dan eksponen terbesar (=+7). Jadi bilangan tersebut adalah 1111111.1_{dua}=$10000000.0_{dua} - 0.1_{dua}$=$2^7 - 0.5 = 126.5$. Bilangan-bilangan yang lebih besar daripada nilai tersebut tidak dapat disajikan dengan notasi di atas, dan disebut nilai *overflow*.

CONTOH 1.8.
Misalkan digunakan mantis m pada (1.4) yang memuat 32 bit. Dalam hal ini, m dapat dituliskan sebagai $m = 0.1b_2b_3b_4...b_{31}b_{32dua}$. Nilai pecahan

$$\frac{1}{10} = 0.000110011001100110011001100110\overline{0011}_{dua},$$

oleh komputer tersebut disimpan sebagai hampiran

$$\frac{1}{10} \approx \underbrace{0.11001100110011001100110011001100}_{memuat\ \ 32\ \ bit\ \ signifikan}{}_{dua} \times 2^{-3}.$$

Galat hampiran tersebut sebesar

$$0.110011\overline{0011}_{dua} \times 2^{-35} \approx 2.328306437 \times 10^{-11}.$$

Jika komputer harus menghitung $\sum_{k=1}^{100000} 0.1$ hasilnya bukan 10000, melainkan hampirannya yang memiliki galat **lebih besar** *daripada $10000 \times 2.328306437 \times 10^{-11} = 2.328306437 \times 10^{-6}$. Silakan Anda coba sendiri dengan kalkulator 10 digit untuk mendapatkan berapa besar galat yang sesungguhnya!* □

CONTOH 1.9.
Perhatikan rumus untuk menghitung nilai z

$$z = \sqrt{x^2 + y^2}.$$

Jika x dan y cukup besar, maka x^2+y^2 mungkin menyebabkan overflow, sekalipun nilai z berada dalam jangkauan titik-mengambang. Untuk menghindari hal ini dapat digunakan rumus alternatif

$$z = \begin{cases} |x|\sqrt{1 + (y/x)^2}, & 0 \le |y| \le |x| \\ |y|\sqrt{1 + (x/y)^2}, & 0 \le |x| \le |y|. \end{cases}$$

Di sini, nilai di dalam tanda akar berbentuk $1 + \omega^2$ dengan $|\omega| \le 1$. Perhitungan bilangan-bilangan biner

$(a)\ 25$ $(b)\ 67$ $(c)\ 435$ $(d)\ 2137$

Gunakan fungsi MATLAB `dec2bin` untuk mengecek hasil perhitungan Anda!

6. Konversikan pecahan-pecahan desimal di bawah ini ke dalam pecahan-pecahan biner berbentuk $0.b_1b_2...b_{n\,dua}$

   $(a)\ \frac{7}{16}$ $(b)\ \frac{13}{16}$ $(c)\ \frac{23}{32}$ $(d)\ \frac{75}{128}$

   Cobalah Anda gunakan fungsi MATLAB `dec2bin` untuk mengubah pecahan-pecahan desimal tersebut ke dalam pecahan-pecahan biner!

7. Gunakan algoritma yang dijelaskan pada persamaan (1.2) untuk menunjukkan bahwa

   $(a)\ \frac{1}{10} = 0.0001100110\overline{0011}_{dua}$ $(b)\ \frac{1}{3} = 0.0101010101\overline{01}_{dua}$
   $(c)\ \frac{1}{7} = 0.001001001\overline{001}_{dua}$

   Tuliskan fungsi MATLAB **dec2binp**, yang mengimplementasikan algoritma yang dijelaskan pada persamaan (1.2), untuk mengubah pecahan desimal ke pecahan biner. Gunakan fungsi tersebut untuk mengerjakan soal-soal di atas!

8. Dengan mengubah pecahan biner ke dalam pecahan desimal, hitunglah galat hampiran-hampiran di bawah ini.

   $(a)\ \frac{1}{10} \approx 0.00011_{dua}$ $(b)\ \frac{1}{3} \approx 0.010101_{dua}$
   $(c)\ \frac{1}{7} \approx 0.001001_{dua}$

9. Misalkan $x, 0 \le x \le 1$, ditulis dalam bentuk biner,

   $$x = 0.b_1b_2b_3..._{dua}.$$

   Jelaskan makna geometris koefisien-koefisien b_1, b_2, b_3, ...!

10. Samakah bilangan-bilangan biner $1.0000..._{dua}$ dan $0.111..._{dua}$? Jelaskan!

$$\pi = 3.1415926536 + galat.$$

Galat relatif pada nilai hampiran 1.414 untuk nilai $\sqrt{2}$ *sekitar*

$$e_{1.414} = \frac{0.0002}{1.414} = 0.00014,$$

sedangkan hampiran yang lebih kasar 1.41 mempunyai galat relatif 0.003.

Hampiran lain yang cukup terkenal adalah $\pi \approx 22/7 = 3.\overline{142857}$. *Nilai* $\pi = 3.1415926535...$, *sehingga*

$$e_{22/7} = \pi - \frac{22}{7} = -0.0012644892, \quad r_{22/7} = \frac{\pi - 22/7}{\pi} = -0.0004024994. \quad \square$$

CONTOH 1.11.
Tentukan galat dan galat relatif pada nilai-nilai hampiran di bawah ini jika nilai eksaknya diketahui:

1. *Hampiran* $\overline{x} = 3.14$ *untuk nilai eksak* $x = 3.141592$.
2. *Hampiran* $\overline{y} = 999,996$ *untuk nilai eksak* $y = 1,000,000$.
3. *Hampiran* $\overline{z} = 0.00009$ *untuk nilai eksak* $z = 0.000012$.

Jawab:

1. $e_{\overline{x}} = 3.141592 - 3.14 = 0.001592$ *dan* $r_{\overline{x}} = \frac{0.001592}{3.141592} \approx 0.000507$.
2. $e_{\overline{y}} = 1,000,000 - 999,996 = 4$ *dan* $r_{\overline{y}} = \frac{4}{1,000,000} = 0.000004$.
3. $e_{\overline{z}} = 0.000012 - 0.00009 = 0.000003$ *dan* $r_{\overline{z}} = \frac{0.000003}{0.000012} = 0.25$.

Pada nomor 1, selisih $e_{\overline{x}}$ *dan* $r_{\overline{x}}$ *tidak terlalu besar, sehingga masing-masing dapat digunakan untuk menentukan tingkat keakuratan* $\overline{x}$. *Pada nomor 2, nilai* y *cukup besar. Sekalipun* $e_{\overline{y}}$ *relatif besar tetapi* $r_{\overline{y}}$ *kecil, sehingga* $\overline{y}$ *dapat dikatakan sebagai hampiran yang cukup baik untuk* y. *Pada nomor 3, nilai* z *terkecil dibanding dengan* x *dan* y, *meskipun galat* $e_{\overline{z}}$ *kecil, tetapi galat relatif* $r_{\overline{z}}$ *cukup besar, yakni 25%. Jadi* $\overline{z}$ *merupakan hampiran yang jelek untuk* z. $\square$

Perhatikan, dari ketiga contoh terakhir, jika $|x|$ nilainya semakin jauh dari 1, baik semakin besar atau semakin kecil, galat relatif $r_{\overline{x}}$ merupakan indikator keakuratan hampiran $\overline{x}$ daripada galat $e_{\overline{x}}$. Galat relatif lebih banyak dipakai pada penyajian bilangan riil dengan **titik-mengambang** (*floating-point representation*), karena galat relatif berkaitan langsung dengan nilai mantis.

Pengertian angka signifikan

DEFINISI 1.2 (ANGKA SIGNIFIKAN).

1. *Misalkan suatu hampiran bilangan x dinyatakan sebagai*

$$\overline{x} = \pm d_n d_{n-1} \ldots d_1 d_0 . d_{-1} d_{-2} \ldots d_{-m} = \pm \sum_{k=-m}^{n} d_k 10^k.$$

Jika $d_k > 0$ dan $d_j = 0$ untuk $j > k$, maka digit-digit d_k, d_{k-1}, ..., d_{-m}, dikatakan angka signifikan.

2. *Suatu digit d_k dikatakan* **benar** *jika $e_{\overline{x}} \leq 10^{-k}$.*

3. *Misalkan x adalah nilai eksak. Hampiran $\overline{x}$ untuk x dikatakan menghampiri x sampai k angka signifikan jika k adalah bilangan bulat positif terbesar yang memenuhi*

$$\frac{|e_{\overline{x}}|}{|x|} = \frac{|x - \overline{x}|}{|x|} < \frac{10^{-k}}{2}.$$

CONTOH 1.12.

1. *Bilangan 25.047 memiliki 5 angka siginitkan.*
2. *Bilangan -0.00250 memiliki 3 angka signifikan, yakni 2, 5, 0.*
3. *Bilangan 0.000068 memiliki 2 angka signifikan, yakni 6 dan 8.*
4. *Bilangan 0.100068 memiliki 6 angka signifikan.*
5. *Jika $x = 3.141592$ dan $\overline{x} = 3.14$, maka $|x - \overline{x}|/|x| = 0.000507 \approx 10^{-3}/2$. Jadi $\overline{x}$ menghampiri x sampai 3 angka signifikan.*
6. *Jika $y = 1000000$ dihampiri oleh $\overline{y} = 999996$, maka $|y - \overline{y}|/|y| = 0.000004 < 10^{-5}/2$. Jadi, $\overline{y}$ menghampiri y sampai lima angka signifikan.*
7. *Jika $z = 0.000012$ dihampiri oleh $\overline{z} = 0.000009$, maka $|z - \overline{z}|/|z| = 0.25 < 10^{-0}/2$. Jadi hampiran $\overline{z}$ tidak memiliki angka signifikan.* □

Jika suatu nilai hampiran ditulis tanpa menyebutkan galat mutlaknya, maka hanya digit-digit yang benar yang ditulis. Dalam hal ini, digit nol di sebelah kanan tidak dihilangkan. Sebagai contoh, bilangan 0.0344 dan 0.034400 adalah dua hampiran yang berbeda. Bilangan 0.0344 memiliki galat mutlak tidak melebihi 0.0001, sedangkan bilangan 0.034400 memiliki galat mutlak tidak lebih daripada 10^{-6}.

Jika bagian bulat suatu bilangan hampiran memiliki lebih banyak angka signifikan daripada cacah digit benar, maka seyogyanya digunakan notasi normal, misalnya $\overline{x} = 0.390 \times 10^5$. Dari notasi ini jelas bahwa $\overline{x}$ memiliki tiga angka signifikan. Dalam hal ini, notasi $\overline{x} = 39000$ tidak disarankan. Bilangan-bilangan hampiran sebelumnya ditulis sebagai 0.344×10^{-1} dan 0.34400×10^{-1}.

Notasi yang sering digunakan untuk menuliskan suatu hampiran adalah

$$x = \overline{x} \pm e_{\overline{x}},$$

yang berarti nilai x memenuhi ketidaksamaan

$$x - e_{\overline{x}} \leq x \leq x + e_{\overline{x}}.$$

Di sini besaran $e_{\overline{x}}$ ditulis dengan cacah digit signifikan yang kurang daripada cacah digit signifikan pada $\overline{x}$. Sebagai contoh,

$$x = 2.730 \pm 0.017.$$

Perlu dibedakan antara cacah digit signifikan benar dan cacah digit benar di sebelah kanan titik pecahan pada suatu nilai hampiran. Misalkan, hampiran $\overline{x} = 25.030$ memiliki lima digit signifikan dan tiga digit benar di sebelah kanan titik pecahan, sedangkan hampiran $\overline{y} = 0.00404$ memiliki tiga digit signifikan benar dan lima digit benar di sebelah kanan titik pecahan.

Jadi, galat mutlak suatu nilai hampiran seutuhnya ditentukan oleh cacah digit benar di sebelah kanan titik pecahan, sedangkan galat relatifnya ditentukan oleh cacah digit signifikan.

1.3.1 Galat Pembulatan (*Rounding Off Error*)

Pembulatan bilangan sering dilakukan di dalam proses komputasi. Pembulatan artinya mengurangi cacah digit pada suatu nilai hampiran dengan cara membuang beberapa digit terakhir. Cara melakukan pembulatan suatu nilai hampiran menggunakan aturan sebagai berikut. *Aturan Pembulatan*

- Jika digit pertama yang dibuang kurang daripada 5, digit di depannya tidak berubah.
- Jika digit pertama yang dibuang lebih atau sama dengan 5, maka digit di depannya ditambah 1 nilainya.

dibulatkan sampai enam angka desimal

$$1 - \frac{(1.5)^2}{2!} + \frac{(1.5)^4}{4!} - \frac{(1.5)^6}{6!} = \frac{50.534375}{720} \approx 0.070187.$$

Galat hampiran tersebut sebesar $0.000550 = 0.550 \times 10^{-3}$ *dan galat relatifnya senilai* $0.007753 < 0.5 \times 10^{-1}$. *Jadi nilai hampiraan tersebut benar sampai 1 angka signifikan.* □

1.3.3 Pemangkasan dan Pembulatan

Perhatikan bahwa setiap bilangan riil x dapat dinyatakan dalam bentuk **desimal normal**:

$$\begin{aligned} x &= \pm 0.d_1 d_2 d_3 ... d_k d_{k+1} ... \times 10^n, \\ \text{dengan} \quad & 1 \le d_1 \le 9, \quad 0 \le d_j \le 9, \quad \text{untuk} \quad j > 1. \end{aligned} \tag{1.9}$$

Misalkan k adalah maksimum banyaknya digit desimal yang dipergunakan oleh komputer untuk melakukan komputasi titik-mengambang. Dalam hal ini, bilangan x disajikan sebagai $fl_{chop}(x)$, yang didefinisikan sebagai

penyajian titik-mengambang terpangkas (chopped floating-point representation)

$$\begin{aligned} fl_{chop}(x) &= \pm 0.d_1 d_2 d_3 ... d_k \times 10^n, \\ \text{dengan} \quad & 1 \le d_1 \le 9, \quad 0 \le d_j \le 9, \quad \text{untuk} \quad 1 < j \le k. \end{aligned} \tag{1.10}$$

Bentuk (1.10) disebut **penyajian titik-mengambang terpangkas** (*chopped floating-point representation*) x. Dalam hal ini, digit ke-k pada $fl_{chop}(x)$ sama dengan digit ke-k pada x. Cara lain penyajian digit ke-k adalah seperti yang sudah dijelaskan sebelumnya, yakni penyajian **titik-mengambang pembulatan** (*rounded floating-point*), ditulis $fl_{round}(x)$, yang didefinisikan sebagai

titik mengambang pembulatan (rounded floating-point)

$$fl_{round}(x) = \pm 0.d_1 d_2 d_3 ... d_{k-1} r_k \times 10^n, \tag{1.11}$$

dengan $1 \le d_1 \le 9$, $0 \le d_j \le 9$, untuk $1 < j < k$ dan digit r_k diperoleh dari pembulatan $d_k.d_{k+1}d_{k+2}...$ ke bilangan bulat terdekat, sebagaimana dijelaskan sebelumnya, pada aturan pembulatan.

Misalnya, bilangan riil $x = 22/7 = 3.142857142857142857...$ memiliki penyajian enam digit dalam bentuk terpangkas

$$fl_{chop}(x) = 0.314285 \times 10^1,$$

sedangkan pembulatannya adalah

$$fl_{round}(x) = 0.314286 \times 10^1.$$

Secara umum kita biasanya menuliskan keduanya sebagai 3.14285 dan 3.14286.

CONTOH 1.15.
Misalkan nilai x disajikan dengan menggunakan notasi **titik-mengambang biner normal** *dengan mantis yang disajikan dalam n bit sebagai $\overline{x}$. Berapakah batas-batas galat mutlak dan galat relatif $\overline{x}$ jika*

1. *digunakan pemangkasan mantis pada bit ke-$(n+1)$, yakni bit ke-$(n+1)$ dan di belakangnya dibuang?*
2. *digunakan pembulatan mantis sampai bit ke-n?*

Jawab:

1. *Pemangkasan mantis pada bit ke-$(n+1)$ menghasilkan galat mutlak kurang atau sama dengan nilai tempat digit ke-n. Jadi,*

$$e_{fl_{chop}(x)} = x - fl_{chop}(x) \leq 2^{-n}.$$

Untuk menghitung galat relatif, kita ingat bahwa normalisasi mantis berarti nilai mantis tidak kurang dari $\frac{1}{2}$. Jadi,

$$|r_{fl_{chop}(x)}| \leq \frac{e_{fl_{chop}(x)}}{1/2} \leq \frac{2^{-n}}{1/2} = 2^{-n+1}.$$

Jika dimisalkan $\frac{e_{fl_{chop}(x)}}{x} = \epsilon$, maka $\frac{x - fl_{chop}(x)}{x} = \epsilon$, sehingga diperoleh hubungan

$$fl_{chop}(x) = x(1-\epsilon) = x - x\epsilon,$$

dengan $\epsilon \leq 2^{-n+1}$. Hubungan di atas dapat dituliskan sebagai

$$fl_{chop}(x) = x(1+\epsilon) = x + x\epsilon,$$

dengan $|\epsilon| \leq 2^{-n+1}$.

2. *Galat pembulatan sampai* n *digit signifikan tidak lebih daripada nilai tempat digit satuan ke-*$(n+1)$, *atau separuh nilai tempat digit satuan ke-*n. *Jadi,*

$$|e_{fl_{round}(x)}| = |x - fl_{round}(x)| \leq 2^{-n-1}.$$

Galat relatifnya adalah

$$|r_{fl_{round}(x)}| \leq \frac{|e_{fl_{round}(x)}|}{1/2} \leq \frac{2^{-n-1}}{1/2} = 2^{-n}.$$

Jika dimisalkan $\frac{e_{fl_{round}(x)}}{x} = \epsilon$, *maka diperoleh hubungan*

$$fl_{round}(x) = x(1+\epsilon) = x + x\epsilon,$$

dengan $|\epsilon| \leq 2^{-n}$. □

Dari contoh di atas dapat disimpulkan bahwa, pada penyajian titik-mengambang biner normal yang menggunakan mantis n-bit untuk bilangan x, ditulis $fl(x)$, memenuhi

Hubungan antara galat relatif dan penyajian titik-mengambang

$$fl(x) = x(1+\epsilon), \quad \text{dengan} \quad \begin{cases} |\epsilon| \leq 2^{-n+1} & \text{pada pemangkasan} \\ |\epsilon| \leq 2^{-n} & \text{pada pembulatan.} \end{cases} \tag{1.12}$$

Oleh karena ϵ kecil, maka hubungan (1.12) menyatakan bahwa $fl(x)$ merupakan **pertubasi** (perubahan) kecil nilai x.

Hubungan (1.12) dapat diperumum untuk sebarang sistem bilangan basis β. Jika $fl(x)$ adalah penyajian bilangan x dalam bentuk titik-mengambang basis β (β adalah suatu bilangan bulat genap) yang menggunakan n digit mantis, maka

$$fl(x) = x(1+\epsilon), \quad \text{dengan} \quad \begin{cases} |\epsilon| \leq \beta^{-n+1} & \text{pada pemangkasan} \\ |\epsilon| \leq \frac{1}{2}\beta^{-n+1} & \text{pada pembulatan.} \end{cases} \tag{1.13}$$

Penyajian bilangan titik-mengambang yang sudah dijelaskan sebelumnya biasanya dikenal sebagai bilangan dengan presisi tunggal. Dalam beberapa bahasa pemrograman, misalnya Pascal dan FORTRAN, bilangan-bilangan tersebut dikenal sebagai jenis REAL. Dalam melakukan banyak perhitungan aritmetika dengan bilangan-bilangan berpresisi tungbuah bilangan yang nilainya hampir sama akan menyebabkan pengurangan angka signifikan. Fenomena ini disebut **kehilangan signifikansi** (*loss of significance*) atau **pembatalan pengurangan** (*substractive cancellation*). Di sinilah perlunya kehati-hatian di dalam komputasi numerik yang melibatkan pengurangan.

Pengurangan dua buah bilangan yang hampir sama nilainya dapat menyebabkan hilangnya beberapa angka signifikan.

CONTOH 1.17.
Fungsi $f(x) = x(\sqrt{x+1} - \sqrt{x})$ dan $g(x) = x/(\sqrt{x+1} + \sqrt{x})$ secara matematis sama. (Buktikan!). Hitunglah $f(500)$ dan $g(500)$ dengan menggunakan enam digit dan pembulatan. Bandingkan hasilnya!

Jawab:
Untuk fungsi f kita hitung

$$f(500) = 500(\sqrt{501} - \sqrt{500}) = 500(22.3830 - 22.3607) = 500(0.0223) = 11.1500.$$

Untuk fungsi g kita hitung

$$\frac{500}{(\sqrt{501} - \sqrt{500})} = \frac{500}{(22.3830 - 22.3607)} = \frac{500}{44.7437} = 11.1748.$$

Perhitungan dengan fungsi g menghasilnya nilai yang lebih mendekati nilai sebenarnya, dan sama dengan hasil pembulatan sampai enam digit nilai yang sebenarnya, yakni 11.174755300747198... . Mengapa terjadi demikian? □

CONTOH 1.18.
Fungsi $f(x) = (e^x - 1 - x)/x^2$, untuk $x \neq 0$, dapat dideretkan ke dalam deret Taylor di sekitar $x = 0$ (Tunjukkan!)

$$\frac{e^x - 1 - x}{x^2} = \frac{1}{2!} + \frac{x}{3!} + \frac{x^2}{4!} + \ldots.$$

Misalkan polinomial $p(x) = \frac{1}{2!} + \frac{x}{3!} + \frac{x^2}{4!}$ digunakan sebagai hampiran fungsi f. Hitunglah $f(0.01)$ dan $p(0.01)$ dengan menggunakan enam digit dan pembulatan. Bandingkan hasilnya!

Jawab:
Dengan menggunakan fungsi f:

$$f(0.01) = \frac{e^{0.01} - 1 - 0.01}{(0.01)^2} \approx \frac{1.010050 - 1 - 0.01}{0.0001} = \frac{0.000050}{0.0001} = 0.50.$$

dengan N_c dan b adalah konstanta-konstanta positif. Jelaskan perbedaan model ini dengan model sebelumnya, dengan melihat perilaku $N(t)$ untuk t cukup besar. Berikan makna fisik konstanta N_c.

3. Hitunglah galat $e_{\overline{x}}$ dan galat relatif $r_{\overline{x}}$ dan banyaknya angka signifikan pada masing-masing nilai hampiran.

   (a) Nilai $x = 2.71828182$ dihampiri dengan $\overline{x} = 2.7182$.

   (b) Nilai $y = 98750$ dihampiri dengan $\overline{y} = 99000$.

   (c) Nilai $z = 0.0000059$ dihampiri dengan $\overline{z} = 0.00006$.

   (d) Nilai $x = e$ dihampiri dengan $\overline{x} = 19/7$.

   (e) Nilai $x = \sqrt{2}$ dihampiri dengan $\overline{x} = 1.414$.

4. Hitunglah hampiran integral $\int_0^{1/4} e^{x^2}\, dx \approx \int_0^{1/4}(1 + x^2 + x^4/2! + x^6/3!)\, dx = \overline{p}$.
   Sebutkan jenis-jenis galat yang ada pada hampiran tersebut! Bandingkan hampiran yang Anda peroleh dengan nilai yang sebenarnya $p = 0.2553074607$.

5. (a) Diberikan data $x = 1.414$ dan $y = 0.06781$, yang masing-masing mempunyai empat angka signifikan. Dengan menggunakan empat angka signifikan, hitunglah $x + y$ dan $x \times y$.

   (b) Diberikan data $x = 31.415$ dan $y = 0.021373$, yang masing-masing memiliki lima angka signifikan. Hitunglah jumlah $x + y$ dan hasil kali $x \times y$ dengan menggunakan lima angka siginifikan.

6. Lengkapilah perhitungan-perhitungan di bawah ini dan sebutkan jenis-jenis galat yang ada.

   (a) $\frac{\sin(\pi/4+0.00001)-\sin(\pi/4)}{0.00001} = \frac{0.7071138523-0.7071067810}{0.00001} = \cdots$.

dan y adalah nilai-nilai eksak (yang tidak diketahui) dan $\overline{x}$ dan $\overline{y}$ berturut-turut adalah hampiran untuk x dan y sedemikian hingga $x = \overline{x} + e_{\overline{x}}$ dan $y = \overline{y} + e_{\overline{y}}$. Galat

$$e_{\overline{x} \circledast \overline{y}} = (x \circledast y) - (\overline{y} \circledast \overline{x}), \tag{1.14}$$

dengan $\circledast$ menyatakan salah satu operasi aritmetika "+", "-", "$\times$", atau "$\div$", disebut **galat rambatan**.

Galat rambatan suatu operasi aritmetika

Dalam praktek komputasi, nilai-nilai x dan y mungkin tidak diketahui. Hanya hampiran-hampirannya, yakni $\overline{x}$ dan $\overline{y}$ yang diketahui, sehingga besar galat (1.14) tidak dapat diketahui secara eksak. Dalam hal ini perlu diketahui batas-batas galat $e_{\overline{x} \circledast \overline{y}}$. Teknik untuk menentukan batas-batas $e_{\overline{x} \circledast \overline{y}}$ dikenal sebagai **aritmetika interval**. Jika batas-batas $x - \overline{x}$ dan $y - \overline{y}$ diketahui, maka dapat ditentukan sebuah interval yang memuat $x \circledast y$.

CONTOH 1.19.
Misalkan diberikan hampiran-hampiran $\overline{x} = 3.14$ *dan* $\overline{y} = 1.412$ *yang memiliki angka signifikan sebagaimana ditunjukkan. Maka* $|x - \overline{x}| \leq 0.005$, $|y - \overline{y}| \leq 0.0005$, *atau* $3.135 \leq x \leq 3.145$, $1.4115 \leq y \leq 1.4125$.
Untuk penjumlahan diperoleh $\overline{x} + \overline{y} = 4.552$, $4.5465 \leq x + y \leq 4.5575$, *sehingga* $4.5465 - 4.552 \leq (x+y) - (\overline{x}+\overline{y}) \leq 4.5575 - 4.552$, *atau* $-0.0055 \leq e_{\overline{x}+\overline{y}} \leq 0.0055$.
Untuk pembagian diperoleh

$$\frac{\overline{x}}{\overline{y}} = \frac{3.14}{1.412} = 2.223796,$$

$$\frac{3.135}{1.4125} = 2.219469 \leq \frac{x}{y} \leq \frac{3.145}{1.4115} = 2.228126,$$

sehingga $2.219469 - 2.223796 \leq (\frac{x}{y}) - (\frac{\overline{x}}{\overline{y}}) \leq 2.228126 - 2.223796$,
atau $-0.004327 \leq e_{\frac{\overline{x}}{\overline{y}}} \leq 0.004330$. □

Cobalah Anda lanjutkan contoh di atas dengan menghitung interval-interval yang memuat $e_{\overline{x}-\overline{y}}$ dan $e_{\overline{x}.\overline{y}}$.

Teknik **aritmetika interval** tersebut sangat berguna dan telah diimplementasikan pada beberapa komputer, baik secara hardware maupun software. Akan tetapi, untuk perhitungan yang lebih luas, aritmetika interval harus diterapkan secara hati-hati karena jika tidak, dapat menghasilkan perkiraan galat yang jauh di luar galat sesungguhnya. Karena alasan inilah, pada saat ini teknik tersebut belum digunakan secara luas dalam komputasi-komputasi praktis (Atkinson, 1993:49).

1.4.1 Galat Penjumlahan dan Pengurangan

Dari hubungan antara nilai eksak, hampiran dan galat di atas,

$$x + y = \overline{x} + e_{\overline{x}} + \overline{y} + e_{\overline{y}} = (x + y) + (e_{\overline{x}} + e_{\overline{y}}).$$

Jadi, galat penjumlahan sama dengan jumlah galat suku-suku yang dijumlahkan, atau dapat dituliskan

$$e_{\overline{x}+\overline{y}} = e_{\overline{x}} + e_{\overline{y}}. \tag{1.17}$$

Galat relatif penjumlahan adalah

$$r_{\overline{x}+\overline{y}} = \frac{e_{\overline{x}+\overline{y}}}{\overline{x}+\overline{y}} = \frac{e_{\overline{x}} + e_{\overline{y}}}{\overline{x}+\overline{y}}. \tag{1.18}$$

Untuk pengurangan,

$$x - y = \overline{x} + e_{\overline{x}} - \overline{y} + e_{\overline{y}} = (x - y) + (e_{\overline{x}} - e_{\overline{y}}).$$

Jadi, analog dengan penjumlahan, galat pengurangan sama dengan selisih galat, atau dapat dituliskan

$$e_{\overline{x}-\overline{y}} = e_{\overline{x}} - e_{\overline{y}}. \tag{1.19}$$

Galat relatif pengurangan adalah

$$r_{\overline{x}-\overline{y}} = \frac{e_{\overline{x}-\overline{y}}}{\overline{x}-\overline{y}} = \frac{e_{\overline{x}} - e_{\overline{y}}}{\overline{x}-\overline{y}}. \tag{1.20}$$

Dari (1.20) dapat dipahami bahwa, apabila $\overline{x} \approx \overline{y}$ maka galat relatif pengurangan kedua hampiran akan semakin besar. (Mengapa?) Akibatnya adalah hilangnya beberapa angka signifikan pada hasil pengurangan. Hal ini persis seperti yang sudah dibahas sebelumnya.

CONTOH 1.20.
Misalkan nilai-nilai $x_1,\ x_2,\ ...,\ x_n$ digunakan sebagai hampiran untuk $X_1,\ X_2,\ ...,\ X_n$ dengan maksimum galat yang mungkin untuk masing-masing hampiran adalah E. Dengan kata lain,

$$x_i - E \leq X_i \leq x_i + E, \quad \textit{untuk} \quad i = 1,\ 2,\ ...,\ n.$$

Jadi galat yang sesungguhnya adalah sekitar 0.017, lebih kecil daripada maksimum galatnya, 0.5. □

Catatan:
Pemakaian MATLAB yang lebih cerdas/baik adalah tanpa *loop*, yakni

```
>> format long g
>> barisan=1:100;s1=sum(round(100*sqrt(barisan)))/100,
   s2=sum(sqrt(barisan))
s1 =
    671.48
s2 =
    671.462947103148
```

CONTOH 1.22.
Hitunglah $\sqrt{543} - \sqrt{540}$ *dengan menggunakan tiga angka signifikan pada masing-masing akar. Bandingkan hasilnya dengan perhitungan menggunakan rumus kesamaan*

$$\sqrt{543} - \sqrt{540} = \frac{3}{\sqrt{543} + \sqrt{540}}.$$

Jawab:
Dengan perhitungan langsung,

$$\sqrt{543} - \sqrt{540} = 23.3 - 23.2 = 0.1$$

Dengan menggunakan rumus ekivalennya,

$$\sqrt{543} - \sqrt{540} = \frac{3}{\sqrt{543} + \sqrt{540}} = \frac{3}{46.5} \approx 0.0645.$$

Hasil kedua memberikan galat yang lebih kecil, karena nilai yang sesungguhnya (dengan menggunakan lebih banyak angka signifikan), adalah 0.0644603. □

Bagaimanakah batas galat penjumlahan dua buah bilangan titik-mengambang biner normal? Misalkan $x = m_1 \times 2^p$ dan $y = m_2 \times 2^q$ dengan $y < x$. Untuk menjumlahkan x dan y, bit-bit mantis m_2 harus digeser $(p - q)$ tempat ke kanan (untuk meratakan titik pecahan biner). Kedua mantis kemudian dijumlahkan dan dibulatkan. Terdapat dua kemungkinan hasil penjumlahan tersebut.

latan pengurangan dua buah bilangan titik-mengambang biner normal.

1.4.2 Galat Perkalian dan Pembagian

Perambatan galat pada perkalian dan pembagian lebih rumit daripada yang terjadi pada penjumlahan dan pengurangan. Hasil kali x dan y adalah

$$xy = (\overline{x} + e_{\overline{x}})(\overline{y} + e_{\overline{y}}) = \overline{xy} + \overline{x}e_{\overline{y}} + e_{\overline{x}}\overline{y} + e_{\overline{x}}e_{\overline{y}}.$$

Jadi, galat hasil kali x dan y adalah

$$e_{\overline{xy}} = xy - \overline{xy} = \overline{x}e_{\overline{y}} + e_{\overline{x}}\overline{y} + e_{\overline{x}}e_{\overline{y}}. \tag{1.22}$$

Apabila harga mutlak $\overline{x}$ dan $\overline{y}$ lebih besar daripada satu, maka suku-suku $\overline{x}e_{\overline{y}}$ dan $e_{\overline{x}}\overline{y}$ menunjukkan adanya kemungkinan peningkatan galat aslinya, $e_{\overline{x}}$ dan $e_{\overline{y}}$. Galat relatif hasil kali tersebut dapat dihitung sebagai berikut. Jika $x \neq 0$ dan $y \neq 0$, maka

$$r_{\overline{xy}} = \frac{e_{\overline{xy}}}{xy} = \frac{\overline{x}e_{\overline{y}} + e_{\overline{x}}\overline{y} + e_{\overline{x}}e_{\overline{y}}}{xy}. \tag{1.23}$$

Selanjutnya, jika galat hampiran $\overline{x}$ dan $\overline{y}$ cukup kecil dibandingkan nilai-nilai hampiran tersebut, maka $\overline{x}/x \approx 1$, $\overline{y}/y \approx 1$ dan $(e_{\overline{x}}/x)(e_{\overline{y}}/y) \approx 0$. Akibatnya, galat relatif (1.23) menjadi

$$r_{\overline{xy}} \approx r_{\overline{x}} + r_{\overline{y}}. \tag{1.24}$$

Jadi, galat relatif hasil kali dua buah hampiran mendekati jumlah galat relatif masing-masing hampiran.

Seringkali suatu galat awal akan merambat selama suatu proses yang terdiri atas serangkaian kalkulasi dalam sebuah algoritma. Dalam hal ini diinginkan agar algoritma yang dipakai memberikan hasil akhir dengan galat kumulatif kecil apabila galat awalnya kecil. Algoritma demikian disebut algoritma **stabil** dan algoritma sebaliknya disebut algoritma **tidak stabil**. Apabila mungkin, komputasi numerik dilakukan dengan menggunakan algoritma stabil.

DEFINISI 1.3.

Misalkan ϵ adalah galat awal dan $e(n)$ menyatakan galat yang terjadi setelah

n langkah. Jika $|e(n)| \approx n\epsilon$, maka algoritma tersebut dikatakan memiliki perambatan galat **linier**. *Jika $|e(n)| \approx K^n\epsilon$, maka algoritma tersebut dikatakan memiliki perambatan galat* **eksponensial**. *Jika $K > 1$, galat eksponensial tumbuh semakin besar tak terbatas jika n semakin besar, dan jika $0 < K < 1$, galat eksponensial akan semakin mengecil jika n semakin besar.*

Untuk menghitung batas galat perkalian dua buah bilangan titik-mengambang biner normal, misalkan $x = m_1 \times 2^p$ dan $y = m_2 \times 2^q$. Selanjutnya, $x \times y = m_1 m_2 \times 2^{p+q}$ dengan $\frac{1}{4} \leq |m_1 m_2| \leq 1$, karena $\frac{1}{2} \leq m_1,\ m_2 \leq 1$. Dengan tetap memperhatikan normalisasi, maka

$$fl_{round}(x \times y) = \begin{cases} (m_1 m_2 + \epsilon) \times 2^{p+q} & \text{jika} \quad |m_1 m_2| \geq \frac{1}{2} \\ (2m_1 m_2 + \epsilon) \times 2^{p+q-1} & \text{jika} \quad \frac{1}{4} \leq |m_1 m_2| < \frac{1}{2} \end{cases}$$

$$\begin{aligned} fl_{round}(x \times y &= m_1 m_2 \times 2^{p+q} \times \begin{cases} 1 + \frac{\epsilon}{m_1 m_2} & \text{jika} \quad |m_1 m_2| \geq \frac{1}{2} \\ 1 + \frac{\epsilon}{2m_1 m_2} & \text{jika} \quad \frac{1}{4} \leq |m_1 m_2| < \frac{1}{2} \end{cases} \\ &= xy(1+E), \end{aligned}$$

dengan $|E| \leq 2|\epsilon| \leq 2^{-n}$.

CONTOH 1.23.

Tunjukkan bahwa ketiga algoritma di bawah ini dapat digunakan untuk menghasilkan barisan $\{1,\ \frac{1}{3},\ \frac{1}{9},\ \frac{1}{27},\ \frac{1}{81},\ \ldots\}$ secara eksak.

$$a_0 = 1, \qquad a_n = \frac{1}{3} a_{n-1} \qquad \textit{untuk } n = 1,\ 2, \ldots \tag{1.25a}$$

$$b_0 = 1, \quad b_1 = \frac{1}{3}, \quad b_n = \frac{4}{3} b_{n-1} - \frac{1}{3} b_{n-2} \qquad \textit{untuk } n = 2,\ 3, \ldots \tag{1.25b}$$

$$c_0 = 1, \quad c_1 = \frac{1}{3}, \quad c_n = \frac{10}{3} c_{n-1} - c_{n-2} \qquad \textit{untuk } n = 2,\ 3, \ldots \tag{1.25c}$$

Penyelesaian:

Barisan yang dimaksud adalah $\{\frac{1}{3^n} : n = 0,\ 1,\ 2, \ldots\}$. Rumus (1.25a) *jelas menghasilkan barisan tersebut. Pada rumus* (1.25b) *persamaan selisih $b_n = \frac{4}{3} b_{n-1} - \frac{1}{3} b_{n-2}$ memiliki penyelesaian umum $b_n = p(1/3^n) + q$. Hal*

Galat awal a_0 pada (1.26a) *adalah 0.00004, dan galat awal b_1 dan c_1 pada* (1.26b) *dan* (1.26c) *adalah 0.000013333. Kode MATLAB berikut menghasilkan ketiga hampiran barisan tersebut sampai suku ke-26 ($n = 25$).*

```
>>a=[];
>>a0=0.99996;a1=a0/3;b0=1;b1=0.33332;c0=1;c1=0.33332;
>>a=[a0 b0 c0; a1 b1 c1]; idx=[0;1];
>>for n=2:25,
an=a1/3;a1=an;bn=4/3*b1-b0/3;b0=b1;b1=bn;
cn=10/3*c1-c0;c0=c1;c1=cn;
a=[a;an bn cn]; idx=[idx;n];
end
>>[idx a]
 ans  =
    0.     0.99996      1.            1.
    1.     0.33332      0.33332       0.33332
    2.     0.1111067    0.1110933     0.1110667
    3.     0.0370356    0.0370178     0.0369022
    4.     0.0123452    0.0123259     0.0119407
    5.     0.0041151    0.0040953     0.0029002
    6.     0.0013717    0.0013518   - 0.0022733
    7.     0.0004572    0.0004373   - 0.0104778
    8.     0.0001524    0.0001324   - 0.0326526
    9.     0.0000508    0.0000308   - 0.0983642
    10.    0.0000169  - 0.0000031   - 0.2952281
    11.    0.0000056  - 0.0000144   - 0.8857294
    12.    0.0000019  - 0.0000181   - 2.6572031
    13.    6.272E-07  - 0.0000194   - 7.9716144
    14.    2.091E-07  - 0.0000198   - 23.914845
    15.    6.969E-08  - 0.0000199   - 71.744535
    16.    2.323E-08  - 0.0000200   - 215.2336
    17.    7.743E-09  - 0.0000200   - 645.70081
    18.    2.581E-09  - 0.0000200   - 1937.1024
    19.    0.         - 0.0000200   - 5811.3073
    20.    0.         - 0.0000200   - 17433.922
    21.    0.         - 0.0000200   - 52301.766
    22.    0.         - 0.0000200   - 156905.3
    23.    0.         - 0.0000200   - 470715.89
    24.    0.         - 0.0000200   - 1412147.7
```

```
   25.    0.         - 0.0000200  - 4236443.
```

Seperti terlihat pada hasil keluaran MATLAB di atas, matriks a *merupakan ketiga hampiran 26 suku pertama barisan* $\{1/3^n\}$*. Kolom pertama matriks di atas adalah indeks (nilai-nilai* n*). Kolom kedua, ketiga, dan keempat berturut-turut adalah* a_n*,* b_n*, dan* c_n*. Terlihat bahwa barisan* $\{a_n\}$ *bersifat menurun secara eksponensial. Barisan* $\{b_n\}$ *bersifat menurun secara eksponensial sampai suku ke-10 dan setelah suku ke-16 suku-sukunya hampir konstan dan kehilangan angka signifikan. Barisan* $\{c_n\}$ *tidak stabil dan mulai suku ke-15 naik secara eksponensial, sehingga jauh dari barisan yang hendak dihampiri.*

Galat ketiga hampiran tersebut dapat dihasilkan dengan kode MATLAB sebagai berikut. Tampak bahwa galat hampiran $\{a_n\}$ *semakin mengecil secara eksponen, galat hampiran* $\{b_n\}$ *semakin konstan, sedangkan galat hampiran* $\{c_n\}$ *cukup besar dibandingkan nilai-nilai yang dihampiri. Jadi dapat disimpulkan bahwa barisan* $\{a_n\}$ *memberikan hampiran terbaik.* □

```
>>x=[1];
>>for k=1:25, x=[x;1/3^k];end
>>e_a=[x x x]-a;
>>[idx e_a]
 ans =
   0.     0.00004      0.           0.
   1.     0.0000133    0.0000133    0.0000133
   2.     0.0000044    0.0000178    0.0000444
   3.     0.0000015    0.0000193    0.0001348
   4.     4.938E-07    0.0000198    0.0004049
   5.     1.646E-07    0.0000199    0.0012150
   6.     5.487E-08    0.0000200    0.0036450
   7.     1.829E-08    0.0000200    0.0109350
   8.     6.097E-09    0.0000200    0.0328050
   9.     2.032E-09    0.0000200    0.098415
   10.    0.           0.0000200    0.295245
   11.    0.           0.0000200    0.885735
   12.    0.           0.0000200    2.657205
   13.    0.           0.0000200    7.971615
   14.    0.           0.0000200    23.914845
   15.    0.           0.0000200    71.744535
   16.    0.           0.00002      215.23361
   17.    0.           0.00002      645.70082
   18.    0.           0.00002      1937.1024
```

Nilai tempat dalam sistem biner

Sistem Biner Jika N bilangan bulat positif yang dapat dinyatakan dalam ekspansi berbasis pangkat dua sebagai

$$N = b_n \times 2^n + b_{n-1} \times 2^{n-1} + ... + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0,$$

dengan $b_k \in \{0,\ 1\}$, maka dalam sistem biner N dapat dinyatakan sebagai

$$N = b_n b_{n-1} ... b_2 b_1 b_0\,_{dua}.$$

Algoritma untuk mengubah bilangan desimal ke bilangan biner

Konversi Desimal ke Biner Jika diketahui bilangan bulat positif N (dalam bentuk desimal), maka bentuk binernya dapat dicari dengan menggunakan algoritma sebagai berikut.

$$\begin{array}{rcl} N & = & 2q_0 + b_0 \\ q_0 & = & 2q_1 + b_1 \\ q_1 & = & 2q_2 + b_2 \\ & \vdots & \\ q_{n-2} & = & 2q_{n-1} + b_{n-1} \\ q_{n-1} & = & 2q_n + b_n, \qquad (q_n = 0), \end{array}$$

dengan $b_k \in \{0,\ 1\}$. Selanjutnya, dalam sistem biner N dapat dinyatakan sebagai

$$N = b_n b_{n-1} ... b_2 b_1 b_0\,_{dua}.$$

Sistem Heksadesimal Dalam sistem heksadesimal digunakan basis perpangkatan 16 dan semua bilangan dinyatakan dengan menggunakan maksimum 16 digit, yang biasanya dinyatakan sebagai

$$0,\ 1,\ 2,\ ...,\ 9,\ A,\ B,\ C,\ D,\ E,\ F,$$

dengan $A_{16} = 10$, $B_{16} = 11$, $C_{16} = 12$, $D_{16} = 13$, $E_{16} = 14$, dan $F_{16} = 15$.

Pecahan Biner Jika $0 < R < 1$ adalah bilangan riil dan $b_1,\ b_2,\ b_3,\ ...,\ b_n \in \{0,\ 1\}$ sedemikian hingga

$$R = b_1 \times 2^{-1} + b_2 \times 2^{-2} + b_3 \times 2^{-3} + ... + b_n \times 2^{-n},$$

maka dalam sistem biner pecahan tersebut dinyatakan sebagai

$$R = 0.b_1 b_2 b_3 ... b_n\ _{dua}$$

Titik-Mengambang Normal (Normalized Floating-Point)

komputer akan disimpan sebagai hampirannya, yakni

$$x \approx \sigma\, m \times 2^p$$

dengan $\sigma = \pm$ disebut **tanda**, m disebut **mantis** bernilai $\frac{1}{2} \leq m < 1$, yang dinyatakan sebagai pecahan biner, dan p disebut **eksponen** atau pangkat.

Galat Hampiran Misalkan $\overline{x}$ adalah suatu nilai hampiran numerik untuk nilai numerik eksak x, yang tidak diketahui. Nilai

$$e_{\overline{x}} = x - \overline{x}$$

disebut **galat**, $|e_{\overline{x}}|$ disebut **galat mutlak**, dan nilai

$$r_{\overline{x}} = \frac{|x - \overline{x}|}{x}$$

asalkan $x \neq 0$, disebut **galat relatif**.
Nilai-nilai $e^*_{\overline{x}}$ dan $r^*_{\overline{x}}$, yang sudah diketahui, dan memenuhi

$$|e_{\overline{x}}| \leq e^*_{\overline{x}} \qquad \text{dan} \qquad |r_{\overline{x}}| \leq r^*_{\overline{x}},$$

disebut berturut-turut **batas galat mutlak** dan **batas galat relatif**, dan jika $x \neq 0$, hubungan keduanya didefinisikan sebagai

$$r^*_{\overline{x}} = \frac{e_{\overline{x}}}{|x|}.$$

Angka Signifikan Pengertian angka signifikan dapat dijelaskan sebagai berikut.

1. Misalkan suatu hampiran bilangan x dinyatakan sebagai

$$\overline{x} = \pm d_n d_{n-1} ... d_1 d_0 . d_{-1} d_{-2} ... d_{-m} = \pm \sum_{k=-m}^{n} d_k 10^k.$$

   Jika $d_k > 0$ dan $d_j = 0$ untuk $j > k$, maka digit-digit d_k, d_{k-1}, ..., d_{-m}, dikatakan angka signifikan.

2. Suatu digit d_k dikatakan **benar** jika $e_{\overline{x}} \leq 10^{-k}$.
3. Misalkan x adalah nilai eksak. Hampiran $\overline{x}$ untuk x dikatakan

menghampiri x sampai k angka signifikan jika k adalah bilangan bulat positif terbesar yang memenuhi

$$\frac{|e_{\overline{x}}|}{|x|} = \frac{|x - \overline{x}|}{|x|} < \frac{10^{-k}}{2}.$$

Aturan Pembulatan Cara melakukan pembulatan suatu nilai hampiran menggunakan aturan sebagai berikut.

- Jika digit pertama yang dibuang kurang daripada 5, digit di depannya tidak berubah.
- Jika digit pertama yang dibuang lebih atau sama dengan 5, maka digit di depannya ditambah 1 nilainya.

Pembulatan dan Pemotongan Mantis Misalkan bilangan riil x dapat dinyatakan dalam bentuk **desimal normal**:

$$x = \pm 0.d_1 d_2 d_3 ... d_k d_{k+1} ... \times 10^n,$$
$$\text{dengan} \quad 1 \leq d_1 \leq 9, \quad 0 \leq d_j \leq 9, \quad \text{untuk} \quad j > 1.$$

Misalkan k adalah maksimum banyaknya digit desimal yang dipergunakan oleh komputer untuk melakukan komputasi titik-mengambang. Penyajian x dalam bentuk **titik-mengambang terpangkas** (*chopped floating-point representation*), ditulis $fl_{chop}(x)$, didefinisikan sebagai

$$fl_{chop}(x) = \pm 0.d_1 d_2 d_3 ... d_k \times 10^n,$$
$$\text{dengan} \quad 1 \leq d_1 \leq 9, \quad 0 \leq d_j \leq 9, \quad \text{untuk} \quad 1 < j \leq k.$$

Dalam hal ini, digit ke-k pada $fl_{chop}(x)$ sama dengan digit ke-k pada x. Cara lain penyajian digit ke-k adalah seperti yang sudah dijelaskan sebelumnya, yakni penyajian **titik-mengambang pembulatan** (*rounded floating-point*), ditulis $fl_{round}(x)$, yang didefinisikan sebagai

$$fl_{round}(x) = \pm 0.d_1 d_2 d_3 ... d_{k-1} r_k \times 10^n,$$

dengan $1 \leq d_1 \leq 9$, $0 \leq d_j \leq 9$, untuk $1 < j < k$ dan digit r_k diperoleh dari pembulatan $d_k.d_{k+1}d_{k+2}...$ ke bilangan bulat terdekat.

Titik-Mengambang Sebarang Basis Jika $fl(x)$ adalah penyajian bilangan x dalam bentuk titik-mengambang basis β (β adalah suatu bilangan bulat genap) yang menggunakan n digit mantis, maka

$$fl(x) = x(1+\epsilon), \quad \text{dgn} \quad \begin{cases} |\epsilon| \leq \beta^{-n+1} & \text{pada pemangkasan} \\ |\epsilon| \leq \frac{1}{2}\beta^{-n+1} & \text{pada pembulatan.} \end{cases}$$

Perambatan Galat Misalkan x dan y adalah nilai-nilai eksak (yang tidak diketahui) dan $\overline{x}$ dan $\overline{y}$ berturut-turut adalah hampiran untuk x dan y sedemikian hingga $x = \overline{x} + e_{\overline{x}}$ dan $y = \overline{y} + e_{\overline{y}}$. Jika $\circledast$ menyatakan salah satu operasi aritmetika "+", "-", "$\times$", atau "$\div$", maka **galat rambatan** didefinisikan sebagai

$$e_{\overline{x}\circledast\overline{y}} = (x \circledast y) - (\overline{y} \circledast \overline{x}).$$

1. Galat Penjumlahan dan Pengurangan
   (a) $e_{\overline{x}+\overline{y}} = e_{\overline{x}} + e_{\overline{y}} \quad r_{\overline{x}+\overline{y}} = \frac{e_{\overline{x}+\overline{y}}}{\overline{x}+\overline{y}} = \frac{e_{\overline{x}}+e_{\overline{y}}}{\overline{x}+\overline{y}}$
   (b) $e_{\overline{x}-\overline{y}} = e_{\overline{x}} - e_{\overline{y}} \quad r_{\overline{x}-\overline{y}} = \frac{e_{\overline{x}-\overline{y}}}{\overline{x}-\overline{y}} = \frac{e_{\overline{x}}-e_{\overline{y}}}{\overline{x}-\overline{y}}$
   (c) Jika x dan y adalah bilangan-bilangan dalam bentuk titik-mengambang biner normal dengan mantis yang terdiri atas n bit, maka $fl_{round}(x+y) = (x+y)(1+\epsilon)$ dengan $|\epsilon| \leq 2^{-n}$.
2. Galat Perkalian
   (a) $e_{\overline{xy}} = xy - \overline{xy} = \overline{x}e_{\overline{y}} + e_{\overline{x}}\overline{y} + e_{\overline{x}}e_{\overline{y}}$
   (b) Jika $x \neq 0$ dan $y \neq 0$, maka $r_{\overline{xy}} = \frac{e_{\overline{xy}}}{xy} = \frac{\overline{x}e_{\overline{y}}+e_{\overline{x}}\overline{y}+e_{\overline{x}}e_{\overline{y}}}{xy}$.
   (c) Jika galat hampiran $\overline{x}$ dan $\overline{y}$ cukup kecil dibandingkan nilai-nilai hampiran tersebut, maka $\overline{x}/x \approx 1$, $\overline{y}/y \approx 1$ dan $(e_{\overline{x}}/x)(e_{\overline{y}}/y) \approx 0$, sehingga $r_{\overline{xy}} \approx r_{\overline{x}} + r_{\overline{y}}$.
   (d) Jika x dan y adalah bilangan-bilangan dalam bentuk titik-mengambang biner normal dengan mantis yang terdiri atas n bit, maka $fl_{round}(x \times y) = xy(1+\epsilon)$, dengan $|\epsilon| \leq 2^{-n}$.