MODUL 1

Konsep Dasar Perhitungan Numerik

Drs. Mulyatno, M.Si.

PENDAHULUAN

Dalam mata kuliah Kalkulus, Aljabar Linear, Persamaan Diferensial Biasa, dan mata kuliah lainnya, dapat Anda pelajari berbagai metode penyelesaian eksak (analitik) dari bermacam-macam persoalan (problema) matematika. Namun demikian sebenarnya masih banyak persoalan matematika yang sulit, atau mungkin saat ini tidak dapat diselesaikan secara eksak. Persoalan matematika seperti ini bahkan sering dijumpai dalam kehidupan praktis, dalam pekerjaan riset, atau dalam kehidupan nyata lainnya, dimana persoalan matematika tersebut berasal dari suatu pemodelan matematis. Persoalan seperti ini mendorong para matematikawan, atau para ilmuan lainnya, mencari dan mengembangkan metode penyelesaian yang bersifat pendekatan yang dikenal sebagai *metode numerik*.

Dalam pengembangannya, metode numerik ternyata juga sangat sederhana, karena hanya menggunakan operasi-operasi aritmetika, dan juga sangat efisien, karena dapat digunakan dengan bantuan komputer. Dengan bantuan alat ini, perhitungan-perhitungan (*calculation*) dapat dilakukan dengan cepat dan dengan kesalahan yang relatif kecil, apalagi dengan adanya kemajuan teknologi komputer yang sangat pesat dewasa ini, baik dalam bidang perangkat lunak (*software*) maupun perangkat keras (*hardware*).

Definisi 1.1 :

Metode Numerik adalah metode penyelesaian persoalan matematika secara pendekatan yang di dalamnya hanya menggunakan operasi aritmatika seperti $+$, $-$, $\times$ dan $\div$. Atau dapat dikatakan bahwa Metode Numerik adalah sekumpulan aturan untuk menyelesaikan persoalan matematika secara pendekatan dengan hanya menggunakan operasi aritmatika.

Dalam kegunaannya sebagai metode alternatif, metode numerik banyak dipergunakan untuk menyelesaikan persoalan matematika yang tidak mempunyai pola umum penyelesaian eksak. Sebagai contoh, marilah kita perhatikan suatu bentuk persoalan matematika dalam bentuk persamaan polinomial berikut ini,

$$a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n = 0$$

dengan $a_0, a_1, \ldots, a_n$ adalah koefisien-koefisien polinom. Untuk n = 1, 2, 3 dan 4, bentuk persamaan polinomial diatas masih mempunyai pola umum penyelesaian, sehingga relatif mudah menyelesaikannya, tetapi untuk $n \geq 5$ telah dibuktikan oleh para matematikawan bahwa persamaan polinomial di atas tidak mempunyai pola penyelesaian umum, sehingga banyak orang lebih suka menyelesaikannya secara pendekatan dengan menggunakan metode numerik. Metode yang dikembangkan untuk menyelesaikan persamaan polinomial di atas, untuk $n \geq 5$, secara umum dapat Anda pelajari dalam Modul 2 BMP ini.

Pada perkembangan selanjutnya, dengan mempertimbangkan segi efisiensinya, metode numerik banyak dipergunakan juga untuk menyelesaikan persoalan matematika yang banyak memerlukan perhitungan numerik, meskipun persoalan matematika tersebut mempunyai penyelesaian eksak. Dalam hal ini metode numerik lebih bersifat sebagai metode menghitung. Sebagai contoh marilah kita perhatikan persoalan matriks sistem persamaan linear berikut ini,

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_n \end{pmatrix}$$

Untuk n kecil, misalkan $n \leq 3$, sistem persamaan linear tersebut di atas dapat diselesaikan dengan cepat tanpa bantuan komputer dengan cara-cara yang dapat Anda pelajari pada mata kuliah Aljabar Linear Elementer. Untuk n besar, misalkan n = 25, penyelesaian tanpa bantuan komputer membutuhkan banyak waktu, sehingga tidak efisien. Karena itu orang lebih suka menggunakan metode numerik, dan metodenya akan dapat Anda pelajari pada Modul 3, 4 dan 5 dalam BMP ini.

Selain masalah-masalah yang telah dicontohkan di atas, dalam BMP ini Anda juga akan mempelajari metode numerik untuk mencari penyelesaian masalah eigen (modul 6), penghampiran (*approximation*) dan interpolasi polinomial (Modul 7 , 8 dan 9), masalah diferensiasi dan integrasi numerik (Modul 10), serta masalah nilai awal persamaan diferensial biasa (Modul 11 dan 12). Sajian materi ini dimaksudkan untuk memberikan dasar pengetahuan dalam perhitungan numerik, khususnya untuk matematika dasar seperti kalkulus, aljabar linear elementer dan metode statistika.

Materi yang disajikan di dalam BMP ini didasarkan pada konsep-konsep dan teorema-teorema dasar yang telah diberikan pada mata kuliah Kalkulus I dan II, Aljabar Linear Elementer I dan II, serta Metode Statistika I dan II. Karena itu di dalam BMP ini disajikan kembali konsep-konsep dan teorema-teorema dasar tersebut tetapi tanpa pembuktian. Bagi Anda yang ingin mengetahui pembuktiannya dapat melihat kembali, atau mempelajarinya, pada BMP mata kuliah-mata kuliah tersebut di atas.

Dalam BMP ini, selain membahas metode numerik dari materi-materi yang disebutkan di atas, diberikan pula contoh-contoh pemrograman komputernya. Bahasa pemrograman yang dipergunakan adalah bahasa PASCAL. Tentu saja Anda dapat mengubahnya ke dalam bahasa pemrograman yang lain, yang lebih Anda sukai, misalkan ke dalam bahasa FORTRAN, BASIC, Bahasa C, atau bahasa lainnya, dengan tanpa banyak kesulitan karena logika yang dipergunakan sama.

Meskipun penggunaan komputer sangat efisien, dan contoh-contoh pemrograman komputer juga diberikan, kami anjurkan Anda juga melatih menggunakan kalkultor tangan, terutama dalam mengerjakan soal-soal yang terdapat dalam BMP ini. Hal ini perlu karena di dalam ujian nanti kemungkinan hanya kalkulator yang diperbolehkan dipergunakan sebagai alat bantu menghitung, sehingga Anda tidak asing lagi dalam menggunakan alat tersebut. Di samping itu bagi Anda yang belum mempunyai kesempatan menggunakan komputer, kalkulator-kalkulator dapat Anda jadikan alat bantu yang cukup baik dalam menerapkan ilmu yang sudah Anda pelajari.

Seperti telah disebutkan di atas bahwa metode numerik merupakan penyelesaian pendekatan terhadap persoalan matematika yang kita hadapi, maka dalam penggunaannya kita tak dapat lepas dari masalah galat (error). Hal yang sangat penting dalam penggunaan metode numerik adalah bagaimana membuat galat itu menjadi sekecil mungkin, sehingga penyelesaian pendekatan tersebut dapat dianggap bagus. Masalah galat ini juga akan dibahas, tetapi secara sederhana, di dalam Modul 1 BMP ini. Bagi Anda yang berminat mempelajarinya secara mendalam dapat mempelajarinya dari buku-buku teks Analisis Numerik yang bisa Anda dapatkan dari berbagai sumber. Setelah mempelajari modul ini diharapkan Anda dapat :

1. menggunakan sistem bilangan biner dan heksadesimal;
2. menggunakan titik kambang dalam penulisan bilangan pada sistem bilangan biner, desimal, dan heksadesimal;
3. menjelaskan sumber-sumber galat dan perambatan galat;
4. menghitung galat.

KEGIATAN BELAJAR 1

Representasi Bilangan dengan Menggunakan Komputer

Dalam melakukan perhitungan, komputer tidak menggunakan sistem bilangan desimal seperti yang biasa kita pergunakan. Sesuai dengan sifat mesin komputer, sistem bilangan yang dipergunakan adalah sistem bilangan biner atau perluasannya, seperti misalnya sistem heksadesimal. Masing-masing sistem bilangan menggunakan bilangan dasar (basis) yang berbeda. Sistem bilangan desimal, yaitu sistem bilangan yang biasa kita pergunakan, menggunakan bilangan dasar 10, sistem bilangan biner menggunakan bilangan dasar 2, dan sistem bilangan heksadesimal menggunakan bilangan dasar 16. Selain perbedaan pada bilangan dasarnya, penulisan simbol-simbol bilangan pada masing-masing sistem bilangan juga berbeda. Pada sistem bilangan desimal simbol bilangan dinyatakan dengan angka 0, 1, 2, 3, 4, 5, 6, 7, 8 dan 9, pada sistem bilangan biner dipergunakan simbol bilangan angka 0 dan 1, sedangkan pada sistem heksadesimal dipergunakan simbol bilangan angka 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 dan huruf A, B, C, D, E, F. Pada Modul ini kita tidak akan membahas sistem bilangan itu secara mendalam. Pembahasannya lebih ditekankan pada penggunaannya, terutama yang berkaitan dengan penggunaan komputer.

A. SISTEM BILANGAN BINER

Seperti yang sudah Anda ketahui, dalam sistem bilangan desimal setiap bilangan dapat dinyatakan sebagai jumlahan dari kelipatan 10. Sebagai contoh kita ambil bilangan 243,15 yang dapat dituliskan sebagai,

$$243{,}15 = 2 \cdot 10^2 + 4 \cdot 10^1 + 3 \cdot 10^0 + 1 \cdot 10^{-1} + 5 \cdot 10^{-2}$$

Dalam sistem bilangan biner, dengan bilangan dasar 2 dan simbol bilangan menggunakan angka 0 dan 1, suatu bilangan dapat dinyatakan sebagai jumlahan dari kelipatan 2. Sebagai contoh kita ambil bilangan biner 110101,101 yang dapat dituliskan sebagai,

$$\begin{aligned}110101{,}101 &= 1\cdot 2^5 + 1\cdot 2^4 + 0\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 + 1\cdot 2^{-1} \\ &\quad + 0\cdot 2^{-2} + 1\cdot 2^{-3} \\ &= 53{,}625_{10}\end{aligned}$$

dimana indeks 10 menyatakan sistem desimal. Dari contoh di atas didapatkan suatu konversi bilangan biner ke dalam bilangan desimal,

$$110101{,}101_2 = 53{,}625_{10}$$

dimana indeks 2 menyatakan sistem bilangan biner.

B. OPERASI ARITMATIKA

Operasi aritmetika pada sistem bilangan biner pada dasamya sama dengan pada sistem desimal. Aturan penjumlahan untuk bilangan biner satu angka (digit) adalah sebagai berikut,

$$\begin{array}{r} 0 \\ +\ 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ +\ 1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\ 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\ 1 \\ \hline 10 \end{array}$$

sedangkan aturan perkalian dua bilangan biner satu angka (satu digit) adalah sebagai berikut,

$$\begin{array}{r} 0 \\ \times\ 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ \times\ 1 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ \times\ 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ \times\ 1 \\ \hline 1 \end{array}$$

Contoh 1.1 :

Selesaikanlah penjumlahan dan perkalian dua bilangan biner di bawah ini :

1.a) $\begin{array}{r} 10110 \\ +\ \underline{11011} \end{array}$ b) $\begin{array}{r} 110001 \\ +\ \underline{\ 10111} \end{array}$ c) $\begin{array}{r} 10101 \\ +\ \underline{111001} \end{array}$

2.a) $\begin{array}{r} 101 \\ \times\ \underline{110} \end{array}$ b) $\begin{array}{r} 1010 \\ \times\ \underline{\ 101} \end{array}$ c) $\begin{array}{r} 1110 \\ \times\ \underline{\ \ 11} \end{array}$

Penyelesaian :

1.a)
$$\begin{array}{r} {\scriptstyle 1\,1\,1} \\ 10110 \\ +\ \underline{11011} \\ 110001 \end{array}$$

b)
$$\begin{array}{r} {\scriptstyle 1\;\;1\,1\,1} \\ 110001 \\ +\ \underline{10111} \\ 1001000 \end{array}$$

c)
$$\begin{array}{r} {\scriptstyle 1\qquad 1} \\ 10101 \\ +\ \underline{111001} \\ 1001110 \end{array}$$

2.a)
$$\begin{array}{r} 101 \\ \times\underline{110} \\ 000 \\ 101 \\ +\ \underline{101} \\ 11110 \end{array}$$

b)
$$\begin{array}{r} 1010 \\ \underline{101}\ \times \\ 1010 \\ 0000 \\ +\ \underline{1010} \\ 110010 \end{array}$$

c)
$$\begin{array}{r} 1110 \\ \underline{11}\ \times \\ 1110 \\ +\ \underline{1110} \\ 101010 \end{array}$$

Selanjutnya kita dapat membuat tabulasi penjumlahan bilangan biner seperti pada tabel di bawah ini.

+	1	10	11	100	101	⋯
1	10	11	100	101	110	⋯
10	11	100	101	110	111	⋯
11	100	101	110	111	1000	⋯
100	101	110	111	1000	1001	⋯
101	110	111	1000	1001	1010	⋯
…	…	…	…	…	…	…

dan tabulasi perkalian bilangan biner seperti tabel di bawah ini :

×	1	10	11	100	101	⋯
1	1	10	11	100	101	⋯
10	10	100	110	1000	1010	⋯
11	11	110	1001	1100	1111	⋯
100	100	1000	1100	10000	10100	⋯
101	101	1010	1111	10100	11001	⋯
…	…	…	…	…	…	…

1) Mengubah Bilangan Bulat Sistem Desimal Ke Bentuk Biner

Misalkan kita akan mengubah bilangan desimal 25 ke bentuk biner, maka kita dapat menuliskan bilangan 25 dalam bentuk,

$$25 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

dan kita dapatkan kesetaraan bilangan desimal dengan bilangan biner,

$$\left(25\right)_{10} = \left(11001\right)_2$$

Secara umum jika x menyatakan suatu bilangan bulat pada sistem desimal yang akan diubah ke bentuk bilangan biner, maka x dapat dinyatakan dengan persamaan,

$$x = a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \cdots + a_1 \cdot 2^1 + a_0 \cdot 2^0 \qquad (1.1)$$

dimana $a_0, a_1, \ldots, a_n$ adalah koefisien-koefisien yang dapat bernilai 0 atau 1, dan bilangan biner yang setara dengan bilangan desimal x dinyatakan dengan,

$$\left(a_n a_{n-1} \cdots a_1 a_0\right)_2 = \left(x\right)_{10} .$$

Ada cara praktis untuk mencari nilai koefisien $a_0, a_1, \ldots, a_n$ sehingga kita tidak perlu selalu menggunakan bentuk persamaan (1.1) untuk mengubah bilangan bulat desimal ke bentuk biner. Cara mencari koefisien $a_0, a_1, \ldots, a_n$ adalah sebagai berikut :

Misalkan bilangan bulat x akan kita ubah ke bentuk biner. Pertama-tama kita bagi x dengan 2, hasil baginya kita nyatakan dengan x_1 dan sisanya a_0. Selanjutnya bagi lagi x_1 dengan 2 dan hasil baginya kita nyatakan dengan x_2 dan sisanya a_1. Demikian seterusnya sampai kita dapatkan $x_{n+1} = 0$ dan sisanya a_n.

Contoh 1.2 :

Ubahlah bilangan desimal 73 ke bentuk biner.

Penyelesaian:

$$\begin{array}{r|l l}
2 & 73 & \\
2 & 36 & \longrightarrow \quad \text{sisa } a_0 = 1 \\
2 & 18 & \longrightarrow \quad \text{sisa } a_1 = 0 \\
2 & 9 & \longrightarrow \quad \text{sisa } a_2 = 0 \\
2 & 4 & \longrightarrow \quad \text{sisa } a_3 = 1 \\
2 & 2 & \longrightarrow \quad \text{sisa } a_4 = 0 \\
2 & 1 & \longrightarrow \quad \text{sisa } a_5 = 0 \\
 & 0 & \longrightarrow \quad \text{sisa } a_6 = 1
\end{array}$$

Jadi hasil konversinya,

$$(73)_{10} = (1001001)_2$$

2) Mengubah Bilangan Pecahan Sistem Desimal Ke Sistem Biner

Kita sudah mempelajari cara mengubah bilangan bulat sistem desimal ke bentuk biner, lalu bagaimana mengubah bilangan pecahan pada sistem desimal ke bentuk biner?

Untuk bilangan pecahan, misalkan y adalah bilangan pecahan kurang dari 1,0 pada sistem desimal, bilangan y dapat dinyatakan dalam bentuk,

$$y = b_1 \cdot 2^{-1} + b_2 \cdot 2^{-2} + \cdots + b_n \cdot 2^{-n} \qquad (1.2)$$

$b_1, b_2, \ldots, b_n$ adalah koefisien-koefisien yang dapat bernilai 0 atau 1.

Bilangan biner yang setara dengan y adalah,

$$(0, b_1 b_2 \cdots b_n)_2 = (y)_{10}$$

Kita juga dapat menentukan secara langsung koefisien-koefisien $b_1, b_2, \ldots, b_n$ tanpa harus menggunakan persamaan (1.2). Caranya adalah sebagai berikut :

Misalkan y adalah bilangan pecahan pada sistem desimal. Pertama-tama kita misalkan $y_1 = y$ dan kalikan y_1 dengan 2. Bagian pecahan dari $2y_1$, atau $\text{fraksi } 2y_1$, kita nyatakan dengan y_2, sedangkan bagian bulatnya, atau $\text{int } 2y_1$, menyatakan b_1. Kita ulangi cara di atas dengan mengalikan y_2 dengan 2, dan kita nyatakan $y_3 = \text{fraksi } 2y_2$ serta $b_2 = \text{int } 2y_2$. Demikian seterusnya sampai kita dapatkan $2y_n = 1{,}0$ dan $b_n = \text{int } 2y_n = 1$.

Contoh 1.3 :

Ubahlah bilangan desimal 0,828125 ke bentuk biner.

Penyelesaian:

$y_1 = 0{,}828125$

$2\,y_1 = 1{,}65625 \longrightarrow y_2 = 0{,}65625 \quad b_1 = 1$

$2\,y_2 = 1{,}3125 \longrightarrow y_3 = 0{,}3125 \quad b_2 = 1$

$2\,y_3 = 0{,}625 \longrightarrow y_4 = 0{,}625 \quad b_3 = 0$

$2\,y_4 = 1{,}25 \longrightarrow y_5 = 0{,}25 \quad b_4 = 1$

$2\,y_5 = 0{,}5 \longrightarrow y_6 = 0{,}5 \quad b_5 = 0$

$2\,y_6 = 1{,}0 \longrightarrow y_7 = 0{,}0 \quad b_6 = 1$

Jadi hasil konversinya,

$$0{,}828125_{10} = 0{,}110101_{2}.$$

Secara umum suatu bilangan desimal z dapat dinyatakan dalam bentuk,

$$a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \cdots + a_1 \cdot 2^1 + a_0 \cdot 2^0 + b_1 \cdot 2^{-1} + b_2 \cdot 2^{-2} + \cdots + b_m \cdot 2^{-m}$$

dan bentuk konversi bilangan desimal ke bilangan biner adalah,

$$a_n a_{n-1} \cdots a_0 , b_1 b_2 \cdots b_m \;{}_2 = z\;{}_{10}$$

Contoh 1.4 :

Ubahlah bilangan desima1 14,25 ke bentuk biner.

Penyelesaian :

Bagian bulat dari bilangan 14,25 yaitu 14 , kita ubah ke bentuk biner sebagai berikut,

$$
\begin{array}{lll}
2 \,\underline{|\; 14} & & \\
2 \,\underline{|\; 7} & \longrightarrow & \text{sisa } a_0 = 0 \\
2 \,\underline{|\; 3} & \longrightarrow & \text{sisa } a_1 = 1 \\
2 \,\underline{|\; 1} & \longrightarrow & \text{sisa } a_2 = 1 \\
\quad 0 & \longrightarrow & \text{sisa } a_3 = 1
\end{array}
$$

Kita dapatkan bagian bulatnya,

$$14_{10} = 1110_2$$

Bagian pecahan dari bilangan 14,25, yaitu 0,25, kita ubah ke bentuk biner sebagai berikut,

$$
\begin{array}{llll}
y_1 = 0{,}25 & & & \\
2\,y_1 = 0{,}5 & \longrightarrow & y_2 = 0{,}5 & b_1 = 0 \\
2\,y_2 = 1{,}0 & \longrightarrow & y_3 = 0{,}0 & b_2 = 1
\end{array}
$$

Kita dapatkan bagian pecahannya,

$$0{,}25_{10} = 0{,}01_2$$

Jadi kita dapatkan,

$$14{,}25_{10} = 1110{,}01_2$$

Suatu bilangan pecahan desimal dapat menjadi bilangan pecahan berulang jika diubah ke bentuk biner. Sebagai contoh kita ambil bilangan desimal 0,1 jika diubah ke bentuk biner menjadi,

$$0{,}1_{10} = 0{,}000110001100011\cdots_2 \tag{1.3}$$

Bentuk di atas terjadi karena bilangan desimal 0,1 tidak dapat dinyatakan dalam bentuk,

$$\sum_{i=1}^{n} b_i \, 2^{-i}$$

n adalah suatu bilangan bulat berhingga dan b_i dapat bernilai 0 atau 1.

Di dalam komputer terdapat keterbatasan tingkat akurasi dalam merepresentasikan bilangan biner. Tingkat akurasi komputer biasanya dinyatakan dengan titik kambang. Pernyataan bilangan biner seperti pada persamaan (1.3) dalam komputer yang tingkat akurasinya terbatas akan menimbulkan galat pemenggalan. Kedua masalah ini, yaitu masalah titik kambang dan galat pemenggalan, akan Anda pelajari pada pembahasan berikutnya.

Dalam mengkonversi bilangan pecahan biner yang bentuknya berulang tak hingga dapat dipergunakan aturan deret geometri tak hingga. Untuk jelasnya marilah kita perhatikan contoh berikut ini,

Contoh 1.5 :

Ubahlah bilangan biner $0{,}10101\cdots$ ke bentuk desimal.

Penyelesaian:

Misalkan $x = 0{,}010101\cdots_2$ maka dapat kita tuliskan,

$$\begin{aligned} x &= 2^{-1} + 2^{-3} + 2^{-5} + \cdots \\ &= \frac{1}{2} + \frac{1}{8} + \frac{1}{32} + \cdots \end{aligned}$$

Dengan demikian x merupakan jumlah dari deret geometri tak hingga dengan suku awal $a = 1/2$ dan rasio $p = 1/4$. Dengan menggunakan aturan deret geometri tak hingga kita dapatkan,

$$x = \frac{a}{1-p} = \frac{1/2}{1-\,1/4} = \frac{2}{3}$$

atau dalam bentuk pecahan desimal, $x = 0,666\cdots_{10}$

C. SISTEM BILANGAN HEKSADESIMAL

Sistem bilangan heksadesimal mempunyai bilangan dasar 16 dan angka digitnya dinyatakan dengan 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, dan F, dimana huruf-huruf pada sistem tersebut merepresentasikan,

$A = 10_{10}$	$D = 13_{10}$
$B = 11_{10}$	$E = 14_{10}$
$C = 12_{10}$	$F = 15_{10}$

Setiap bilangan heksadesimal x dapat dinyatakan dalam bentuk,

$$x_{16} = a_n 16^n + a_{n-1}16^{n-1} + \cdots + a_1 16^1 + a_0 16^0 + b_1 16^{-1} + b_2 16^{-2} + \cdots + b_m 16^m$$

dimana $a_0, a_1, \ldots, a_n$ dan $b_1, b_2, \ldots, b_m$ dapat bernilai 0 sampai dengan 15.

Contoh 1.6 :

Nyatakan bilangan heksadesimal BA,09_{16} dalam bentuk desimal.

Penyelesaian :

Kita ubah bilangan BA,09_{16} menjadi,

$$\begin{aligned} \text{BA,09}_{16} &= 11\cdot 16^1 + 10\cdot 16^0 + 0\cdot 16^{-1} + 9\cdot 16^{-2} \\ &= 186,3515625_{10} \end{aligned}$$

Untuk mengubah bilangan desimal ke bentuk heksadesimal dapat dipergunakan cara yang sama dengan konversi bilangan desimal ke bentuk biner, hanya di sini faktor pengali/pembaginya adalah 16.

Contoh 1.7 :

a) Konversikan bilangan 161_{10} kebentuk heksadesimal.

b) Konversikan bilangan $0,3359375_{10}$ kebentuk heksadesimal.

c) Konversikan bilangan $1101001,0110101_{2}$ ke bentuk heksadesimal.

Penyelesaian :

a) Misalkan $x = 161$ kita bagi berturut-turut dengan faktor pembagi 16,

$$\begin{array}{rl} & 161 \\ 16 & \overline{} \\ & 10 \quad \longrightarrow \quad \text{sisa } a_0 = 1 \\ 16 & \overline{} \\ & 0 \quad \longrightarrow \quad \text{sisa } a_1 = 10 = \text{A} \end{array}$$

Jadi,

$$161_{10} = \text{A1}_{16}.$$

b) Misalkan $y_1 = 0,3359375$, kemudian kita pergunakan analogi konversi desimal ke biner,

$y_1 = 0,3359375$

$16\, y_1 = 5,375 \longrightarrow y_2 = 0,375 \qquad b_1 = 5$

$16\, y_2 = 6,0 \longrightarrow y_3 = 0,0 \qquad b_2 = 6$

Jadi kita dapatkan,

$$0,3359375_{10} = 0,56_{16}$$

.

Konversi bilangan heksadesimal ke bentuk biner lebih mudah dibandingkan konversi bilangan desimal ke bentuk biner. Untuk mengubah bilangan heksadesimal kebentuk biner cukup kita ubah setiap digit pada bilangan heksadesimal menjadi 4 digit bilangan biner. Keenam belas digit pada sistem heksadesimal jika dikonversikan ke bentuk biner adalah sebagai berikut:

$0_{16} = 0000_2$	$4_{16} = 0100_2$	$8_{16} = 1000_2$	$C_{16} = 1100_2$
$1_{16} = 0001_2$	$5_{16} = 0101_2$	$9_{16} = 1001_2$	$D_{16} = 1101_2$
$2_{16} = 0010_2$	$6_{16} = 0110_2$	$A_{16} = 1010_2$	$E_{16} = 1110_2$

$3_{16} = 0011_2$ $\quad 7_{16} = 0111_2$ $\quad B_{16} = 1011_2$ $\quad F_{16} = 1111_2$

c) Terlebih dahulu kita harus mengubah bagian bulatnya menjadi,

$$1101001 \rightarrow 01101001$$

dan bagian pecahannya menjadi,

$$0110101 \rightarrow 01101010$$

sehingga bilangan $1101001,0110101_2$ dapat dituliskan sebagai,

$$\underbrace{0110}_{6}\underbrace{1001}_{9},\underbrace{0110}_{6}\underbrace{1010}_{A}$$

dan kita dapatkan,

$$1101001,0110101_2 = 69,6A_{16}$$

LATIHAN 1

Untuk memperdalam pemahaman Anda mengenai materi di atas, kerjakanlah latihan berikut!

1) Selesaikan penjumlahan dan perkalian bilangan berikut ini:

a) $\begin{array}{r} 11011 \\ \underline{+\ 1101} \end{array}$ b) $\begin{array}{r} 1010,011 \\ \underline{+\ 11,101} \end{array}$ c) $\begin{array}{r} 11111 \\ \underline{\times\ 101} \end{array}$ d) $\begin{array}{r} 1010,011 \\ \underline{\times\ 11,101} \end{array}$

2) Konversikan bilangan desimal di bawah ini ke bentuk biner
a) 59 b) 5,75 c) $1/3$ d) π

3) Konversikan bilangan biner di bawah ini ke bentuk desimal

a) 110110 b) 1011,01 c) $0,100100100\cdots$

4) Konversikan bilangan desimal di bawah ini ke bentuk heksadesimal

a) 3375 b) 27,25 c) 0,1

5) Konversikan bilangan heksadesimal di bawah ini ke bentuk desimal

a) DF11 b) E9,B8 c) 0,F8

6) Konversikan bilangan heksadesimal di bawah ini ke bentuk biner.

   a) ABAD b) B1,96

7) Konversikan bilangan biner di bawah ini ke bentuk heksadesimal

   a) 101011011 b) 10110,11011

Petunjuk Jawaban Latihan

Untuk Latihan Nomor 1 s.d. 7 kerjakan seperti pada contoh-contoh yang sudah diberikan pada Kegiatan Belajar ini.

RANGKUMAN

Sistem bilangan desimal, yaitu sistem bilangan yang biasa kita pergunakan, menggunakan bilangan dasar 10, sistem bilangan biner menggunakan bilangan dasar 2, dan sistem bilangan heksadesimal menggunakan bilangan dasar 16.

Dalam sistem bilangan biner, dengan bilangan dasar 2 dan simbol bilangan menggunakan angka 0 dan 1, suatu bilangan dapat dinyatakan sebagai jumlahan dari kelipatan 2.

Operasi aritmatik pada sistem bilangan biner pada dasarnya sama dengan pada sistem desimal. Aturan penjumlahan untuk bilangan biner satu angka (digit) adalah sebagai berikut :

$$\begin{array}{r} 0 \\ +\,0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ +\,1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\,0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\,1 \\ \hline 10 \end{array}$$

sedangkan aturan perkalian dua bilangan biner satu angka (digit) adalah sebagai berikut:

$$\begin{array}{r} 0 \\ \times\,0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ \times\,1 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ \times\,0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ \times\,1 \\ \hline 1 \end{array}$$

Jika x menyatakan suatu bilangan bulat pada sistem desimal yang akan diubah ke bentuk bilangan biner, maka x dapat dinyatakan dengan ,

$$x = a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \cdots + a_1 \cdot 2^1 + a_0 \cdot 2^0$$

dimana $a_0, a_1, \ldots, a_n$ adalah koefisien-koefisien yang dapat bernilai 0 atau 1, dan bilangan biner yang setara dengan bilangan desimal x dinyatakan dengan,

$$a_n a_{n-1} \cdots a_1 a_0{}_2 = x{}_{10}$$

Untuk bilangan pecahan, misalkan y adalah bilangan pecahan kurang dari 1,0 pada sistem desimal, bilangan y dapat dinyatakan dalam bentuk,

$$y = b_1 \cdot 2^{-1} + b_2 \cdot 2^{-2} + \cdots + b_n \cdot 2^{-n}$$

dimana $b_1, b_2, \ldots, b_n$ adalah koefisien-koefisien yang dapat bernilai 0 atau 1.

Bilangan biner yang setara dengan y adalah,

$$0, b_1 b_2 \cdots b_n{}_2 = y{}_{10}.$$

Sistem bilangan heksadesimal mempunyai bilangan dasar 16 dan angka digitnya dinyatakan dengan 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E dan F, dimana huruf-huruf pada sistem tersebut merepresentasikan,

$A = 10_{10}$ $\qquad D = 13_{10}$

$B = 11_{10}$ $\qquad E = 14_{10}$

$C = 12_{10}$ $\qquad F = 15_{10}$

Setiap bilangan heksadesimal x dapat dinyatakan dalam bentuk,

$$x{}_{16} = a_n 16^n + a_{n-1} 16^{n-1} + \cdots + a_1 16^1 + a_0 16^0 + b_1 16^{-1} + b_2 16^{-2} + \cdots + b_m 16^m$$

$a_0, a_1, \ldots, a_n$ dan $b_1, b_2, \ldots, b_n$ dapat bernilai 0 sampai dengan 15.

Untuk mengubah bilangan heksadesimal ke bentuk biner cukup kita ubah setiap digit pada bilangan heksadesimal menjadi 4 digit bilangan biner. Keenam belas digit pada sistem heksadesimal jika dikonversikan ke bentuk biner adalah sebagai berikut:

$0_{16} = 0000_2$ $\quad 4_{16} = 0100_2$ $\quad 8_{16} = 1000_2$

$C_{16} = 1100_2$

$1_{16} = 0001_2$ $\quad 5_{16} = 0101_2$ $\quad 9_{16} = 1001_2$

$D_{16} = 1101_2$

$2_{16} = 0010_2$ $6_{16} = 0110_2$ $A_{16} = 1010_2$

$E_{16} = 1110_2$

$3_{16} = 0011_2$ $7_{16} = 0111_2$ $B_{16} = 1011_2$

$F_{16} = 1111_2$

TES FORMATIF 1

Pilihlah satu jawaban yang paling tepat!

1) Selesaikan penjumlahan angka biner, 1110.101 + 10.010 = ...
   A. 10000,001
   B. 10000,111
   C. 10101,010
   D. 11011,101

2) Selesaikan perkalian angka biner, 10101,01 × 10,10 = ...
   A. 101001,110
   B. 101001,011
   C. 110101,001
   D. 110101,101

3) Bentuk biner dari bilangan desimal 125,25 adalah . . . .
   A. 1111101,01
   B. 1111011,10
   C. 1110111,01
   D. 1101111,10

4) Bentuk desimal dari bilangan biner 1101001,101 adalah . . . .
   A. 409,315
   B. 325,245
   C. 231,125
   D. 105,625

5) Bentuk heksadesimal dari bilangan desimal 1234 adalah . . . .
   A. E31
   B. E13
   C. 2D4
   D. 4D2

6) Bentuk desimal dari bilangan heksadesimal 2B9 adalah . . . .
   A. 697
   B. 534
   C. 441
   D. 217

7) Bentuk biner dari bilangan heksadesimal 2AB adalah . . . .
   A. 1110101011
   B. 1011101011
   C. 1100101011
   D. 1010101011

8) Bentuk heksadesimal dari bilangan biner 11010111010011 adalah . . . .
   A. 71A3
   B. A51E
   C. 35D3
   D. 2B18

Cocokkanlah jawaban Anda dengan Kunci Jawaban Tes Formatif 1 yang terdapat di bagian akhir modul ini. Hitunglah jawaban yang benar. Kemudian, gunakan rumus berikut untuk mengetahui tingkat penguasaan Anda terhadap materi Kegiatan Belajar 1.

$$\text{Tingkat penguasaan} = \frac{\textit{Jumlah Jawaban yang Benar}}{\textit{Jumlah Soal}} \times 100\%$$

Arti tingkat penguasaan: 90 - 100% = baik sekali
80 - 89% = baik
70 - 79% = cukup
< 70% = kurang

Apabila mencapai tingkat penguasaan 80% atau lebih, Anda dapat meneruskan dengan Kegiatan Belajar 2. **Bagus!** Jika masih di bawah 80%, Anda harus mengulangi materi Kegiatan Belajar 1, terutama bagian yang belum dikuasai.

KEGIATAN BELAJAR 2

Masalah Galat

Seperti telah disebutkan sebelumnya, penyelesaian persoalan matematika dengan metode numerik adalah penyelesaian pendekatan, sehingga keakuratannya ditentukan oleh seberapa besar galat yang timbul pada penggunaan metode numerik, dan oleh presisi alat (komputer) yang digunakan.

Galat pada suatu perhitungan numerik adalah selisih antara nilai sebenarnya (yang dicari) dengan nilai pendekatan (yang didapat),

$$Galat = harga\ sebenarnya - harga\ pendekatan$$

semakin kecil galat dari hasil perhitungan numerik menunjukkan semakin bagus pendekatannya.

Contoh 1.8 :

Kita ketahui bilangan π adalah 3,14159265... dan nilai pendekatan yang biasa digunakan adalah $22/7$. Jika kita ubah ke bentuk desimal kita dapatkan $22/7 = 3{,}1428571428...$

$$\begin{aligned} \text{Galat} &= 3{,}14159265\ ... - 3{,}14285714... \\ &= -\ 0{,}00126449... \end{aligned}$$

A. SUMBER GALAT

Galat yang dimaksud pada perhitungan numerik adalah galat yang terjadi karena pendekatan matematis. Galat lainnya seperti galat karena kesalahan akurasi pada alat ukur, dan sebagainya, tidak termasuk dalam pembahasan metode numerik. Galat numerik dapat terjadi karena adanya pembulatan atau pemenggalan pada hasil perhitungan numerik. Galat pembulatan adalah galat karena pembatasan banyaknya digit/angka yang harus kita tuliskan. Misalkan untuk nilai π pada contoh sebelumnya, jika harus dituliskan sampai 3 angka di belakang tanda desimal maka kita dapat menuliskan $\pi = 3{,}141$ atau $\pi = 3{,}142$. Cara pembulatan yang pertama, yaitu dengan mengambil nilai $\pi = 3{,}141$, disebut pembulatan dengan cara *chopping*. Dengan cara ini semua angka di belakang angka pembulatan terakhir dihilangkan. Cara pembulatan yang kedua, yaitu dengan mengambil nilai π = 3,142, disebut pembulatan dengan cara *rounding*. Cara pembulatan ini memperhitungkan satu angka berikutnya dari angka pembulatan terakhir. Misalkan nilai π kita tuliskan sebagai berikut:

$$\pi = 3{,}141x\ldots$$

angka pembulatan terakhir

Pada penulisan di atas, x kita sebut satu angka setelah angka pembulatan terakhir. Pembulatan dengan cara *rounding* adalah jika $x \geq 5$ maka dilakukan pembulatan ke atas, yaitu angka pembulatan terakhir ditambah satu menjadi 3,142. Jika $x < 5$ maka dilakukan pembulaan ke bawah, yaitu angka pembulatan terakhir tidak bertambah satu, jadi π = 3,141. Jika kita lihat nilai π pada contoh sebelumnya, yaitu π = 3,14159265..., maka dengan cara *rounding* menghasilkan π = 3,142. Jika kita perhatikan, pembulatan dengan cara *chopping* dan *rounding* menimbulkan galat yang berbeda. Pada contoh di atas, jika nilai π dibulatkan dengan cara *chopping* kita dapatkan galat absolutnya,

$$\varepsilon_1 = |3{,}14159265\ldots - 3{,}141| = 0{,}00059265\ldots$$

dan jika dibulatkan dengan cara *rounding* kita dapatkan galat absolutnya,

$$\varepsilon_2 = |3{,}14159265\ldots - 3{,}142| = 0{,}00040735\ldots$$

dan jika dibulatkan dengan cara *rounding* menghasilkan galat yang lebih kecil. Secara umum, akurasi pembulatan dinyatakan dengan definisi berikut ini.

Definisi 1.2:

Jika x dan y bilangan real dan y mempunyai k digit/angka di belakang tanda desimal, maka kita dapatkan y adalah pembulatan (*rounding*) dari x sampai ke angka desimal jika dan hanya jika

$$|x - y| \leq \frac{1}{2} \cdot 10^{-k}$$

Contoh 1.9 :

Seperti pada pembahasan di atas pembulatan π sampai 3 desimal dengan cara rounding memenuhi $|\pi - 3{,}142| \le \frac{1}{2}\cdot 10^{-3}$.

Sekarang kita perhatikan pendekatan nilai π oleh $22/7$ seperti pada contoh sebelumnya. Untuk mendapatkan pendekatan yang akurat kita harus menentukan banyaknya angka di belakang tanda desimal. Definisi di bawah ini menjadi dasar untuk menentukan akurasi dari suatu pendekatan

Definisi 1.3 :

Jika x bilangan real dan y adalah nilai pendekatan dari x, maka kita katakan y akurat sampai k angka desimal jika,

$$|x - y| \le \frac{1}{2}\cdot 10^{-k}$$

tetapi,

$$|x - y| \ge \frac{1}{2}\cdot 10^{-\ k+1}$$

Contoh 1.10 :

Berapa angka desimalkah $22/7$ akurat sebagai pendekatan nilai π.

Penyelesaian:

Dari contoh sebelumnya dapat kita tentukan,

$$\varepsilon = \left|\pi - \frac{22}{7}\right| = 0{,}00126449...$$

dan kita dapatkan,

$$0{,}0005 < \varepsilon < 0{,}005$$

atau,

$$\frac{1}{2}\cdot 10^{-3} < \varepsilon < \frac{1}{2}\cdot 10^{-2}$$

jadi pendekatan nilai π oleh $22/7$ akurat sampai dua angka desimal.

Seperti telah disebutkan di atas, selain galat pembulatan, pada perhitungan numerik dapat pula terjadi galat pemenggalan. Sebagai contoh, misalkan suatu fungsi $f(x)$ dinyatakan dalam bentuk suatu deret,

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots$$

atau

$$f(x) = \sum_{n=0}^{\infty} a_n x^n$$

Kita dapat mengambil nilai pendekatan $f(x)$ untuk N suku pertama,

$$\tilde{f}(x) = \sum_{n=0}^{N-1} a_n x^n$$

dan mengabaikan suku-suku $\sum_{n=0}^{N-1} a_n x^n$.

Dengan cara pendekatan ini didapatkan galat pemenggalannya:

$$\varepsilon = \sum_{n=0}^{N-1} a_n x^n$$

Dalam metode numerik, kedua jenis galat ini, yaitu galat pembulatan dan galat pemenggalan, harus diperhitungkan. Selain kedua jenis galat ini, di dalam metode numerik juga akan dijumpai adanya galat pendistribusian. Jenis galat ini akan kita bahas pada pembahasan Integral Numerik.

B. GALAT ABSOLUT DAN GALAT RELATIF

Galat dapat dinyatakan sebagai galat absolut dan galat relatif. Jika $\tilde{x}$ menyatakan nilai pendekatan dari x, maka galat absolutnya dinyatakan dengan,

$$\varepsilon_{\text{a}} = |x - \tilde{x}|$$

dan galat relatifnya dinyatakan dengan,

$$\varepsilon_{\mathrm{r}} = \frac{|x - \tilde{x}|}{|x|}$$

Contoh 1.11 :

Misalkan dengan pembulatan sampai 4 angka di belakang tanda desimal kita ambil nilai π = 3,1416 dan pendekatannya $22/7 = 3{,}1429$, maka kita dapatkan galat absolutnya,

$$\varepsilon_{\mathrm{a}} = |3{,}1416 - 3{,}1429| = 0{,}0013$$

dan galat relatifnya,

$$\varepsilon_{\mathrm{r}} = \frac{|3{,}1416 - 3{,}1429|}{3{,}1416} = 0{,}0004138$$

Pada dasarnya akurasi dari suatu pendekatan numerik ditentukan oleh galat relatifnya. Semakin kecil galat relatifnya semakin bagus pendekatannya. Suatu pendekatan numerik yang menghasilkan galat absolut yang sangat kecil belum tentu akurat, karena ada kemungkinan galat relatifnya besar. Sebagai contoh, misalkan $\tilde{x} = 0{,}00015$ adalah pendekatan dari x = 0,00010. Dapat kita tentukan galat absolutnya,

$$\varepsilon_{\mathrm{a}} = |x - \tilde{x}| = 0{,}00005$$

dan galat relatifnya,

$$\varepsilon_{\mathrm{r}} = \frac{|x - \tilde{x}|}{|x|} = 0{,}5$$

dari sini kita dapatkan bahwa ternyata akurasi dari pendekatan x oleh $\tilde{x}$ hanya 50%. Sebaliknya, suatu pendekatan numerik yang menghasilkan galat absolut yang cukup besar kemungkinan merupakan pendekatan yang akurat karena galat relatifnya kecil. Sebagai contoh, misalnya $\tilde{y} = 125{,}35$ merupakan pendekatan dari $y = 124{,}25$. Dapat kita tentukan galat absolutnya,

$$\varepsilon = |y - \tilde{y}| = 1{,}1$$

dan galat relatifnya,

$$\varepsilon_r = \frac{|y - \tilde{y}|}{|y|} = 0,00885...$$

Kita dapatkan bahwa akurasi pendekatan y oleh $\tilde{y}$ ternyata mencapai 99%, artinya pendekatannya sangat baik.

1. Titik Kambang (*Floating Point*)

Untuk mendapatkan gambaran yang lebih jauh mengenai galat pembulatan, perlu kita ketahui dahulu bagaimana suatu bilangan ditampilkan (dinyatakan) di dalam komputer. Pada perangkat lunak (*software*) untuk perhitungan numerik, pada umumnya bilangan-bilangan dinyatakan dalam bentuk titik kambang. Dalam bentuk titik kambang, suatu bilangan *x* dituliskan,

$$x = \sigma \ \ 0,b_1b_2b_3 \cdots b_n \ \ \beta^e$$

dimana $\sigma = +$ atau – adalah tanda bilangan, $b_1, b_2, \cdots, b_n$ adalah digit/angka, β adalah bilangan dasar (basis) dan $\beta \geq 2$, dan e menyatakan pangkat. Pada penulisan di atas, *n* menunjukkan tingkat presisi dan $0,b_1b_2b_3 \cdots b_n$ disebut mantisa.

Contoh 1.12 :

Tuliskan bilangan *x* = 24,625 dalam bentuk titik kambang dengan bilangan dasar 10, 2 dan 16.

Penyelesaian:

Dengan bilangan dasar 10 dengan mudah kita tuliskan, $x = \ 0,24625 \ \cdot 10^2$

Dengan bilangan dasar 2 kita harus mengubah bilangan x ke sistem biner dengan cara yang sudah kita pelajari sebelumnya, kita dapatkan,

$$\begin{aligned} x &= 11000,101 \\ &= \ 0,11000101 \ \cdot 2^5 \end{aligned}$$

Dengan bilangan dasar 16 kita harus mengubah *x* ke sistem heksadesimal. Juga dengan cara yang sudah kita pelajari sebelumnya, kita dapatkan,

$$x = 18, A$$
$$= \ 0{,}18A \ \cdot 16^2$$

Contoh 1.13 :

Ubahlah bilangan $x = \ 0.BC \ 16^2$ ke bentuk titik kambang dengan bilangan dasar 2 dan 10.

Penyelesaian:

Kita konversikan angka heksadesimal B, dan C ke bentuk biner menjadi 1011 dan 1100 dan bilangan dasar 16 menjadi 2^4, sehingga kita dapatkan,

$$x = \ 0{,}10111100 \quad 2^{4}{}^{\ 2}$$
$$= \ 0{,}101111 \ .2^8$$

Untuk mengubah ke bentuk desimal kita tuliskan,

$$x = \ 0, BC \ \cdot 16^2$$
$$= \ B \cdot 16^{-1} + C \cdot 16^{-2} \ \cdot 16^2$$
$$= 11 \cdot 16 + 12$$
$$= 188$$
$$= 0{,}188 \cdot 10^3$$

Setiap komputer mempunyai tingkat akurasi tertentu dalam mempresentasikan bilangan dalam bentuk titik kambang. Untuk mengetahui seberapa akurat suatu bilangan dapat dipresentasikan ke dalam bentuk titik kambang, kita perlu mengetahui bilangan bulat sebesar yang dapat dipresentasikan ke dalam bentuk titik kambang. Misalkan M menyatakan bilangan bulat terbesar yang dapat dipresentasikan ke bentuk titik kambang, maka setiap bilangan bulat *x*, dengan $0 \le x \le M$ tentu dapat dipresentasikan secara akurat ke bentuk titik kambang.

Kita ketahui bahwa representasi bilangan pada komputer pada umumnya menggunakan sistem biner, atau variasinya (misalkan sistem heksadesimal). Mesin-mesin komputer PRIME dan menggunakan sistem biner dan kebanyakan mesin komputer IBM menggunakan sistem heksadesimal. Misalkan suatu komputer menggunakan sistem bilangan biner dan mempunyai kemampuan merepresentasikan titik kambang sampai n angka biner pada mantisanya, maka bilangan bulat terbesar yang dapat direpresentasikan adalah,

$$\underbrace{0{,}11...1}_{\text{n digit}} \cdot 2^n$$

Jika kita konversikan ke bilangan desimal, maka bilangan bulat terbesar itu adalah,

$$2^{n-1} + 2^{n-2} + \cdots + 2^1 + 2^0 = 2^n - 1$$

Jadi bilangan bulat yang dapat direpresentasikan secara akurat adalah dari 0 sampai dengan $2^n - 1$, tetapi pada kenyataannya 2^n juga dapat direpresentasikan secara akurat, sehingga batas akurasinya adalah,

$$M = 2^n$$

Contoh 1.14 :

1) Pada mesin komputer CDC diketahui $M = 2^{48} = 2{,}81 \cdot 10^{14}$, artinya semua bilangan bulat sampai dengan 15 digit, tetapi nilainya tidak lebih dari $2{,}81.10^{14}$, dapat direpresentasikan secara akurat pada mesin tersebut.
2) Pada mesin komputer PRIME diketahui $M = 2^{23} = 8388608$, artinya bilangan bulat yang dapat direpresentasikan secara akurat adalah sampai tujuh digit dengan angka maksimal 8388608.

2. Galat Atas (*Overflows Errors*) dan Galat Bawah (*Underflows Errors)*

Penggunaan titik kambang pada komputer mempunyai batas atas dan batas bawah. Misalkan M menyatakan batas atas dan m menyatakan batas bawah pada suatu komputer, maka bilangan real x yang bisa direpresentasikan dengan titik kambang harus memenuhi $m \le x \le M$. Jika $x > M$ akan terjadi galat atas (*overflows*), dan jika $x < m$ akan terjadi galat bawah (*underflows*), dan mesin komputer akan menyelesaikan pernyataan overflows atau underflows.

Seperti telah disebutkan di atas, dengan menggunakan titik kambang kita akan mencoba untuk lebih memahami adanya galat pembulatan. Jika komputer mempunyai kemampuan merepresentasikan titik kambang sampai *n* digit pada mantisanya, sedangkan bilangan yang akan ditampilkan ternyata lebih dari *n* digit, maka komputer akan memendekkan digit bilangan tersebut menjadi *n* digit dengan cara pembulatan, yang dapat dilakukan dengan cara *chopping* atau *rounding* tergantung pada mesin komputernya.

Suatu komputer yang menggunakan sistem bilangan dengan bilangan dasar β dan mempunyai akurasi sampai *n* digit dalam representasi titik kambangnya, akan mempunyai galat pembulatan ε yang besarnya,

$$-\beta^{-n+1} \le \varepsilon \le 0 \qquad \text{, jika menggunakan } \textit{chopping}$$

dan,

$$-\frac{1}{2}\beta^{-n+1} \le \varepsilon \le \frac{1}{2}\beta^{-n+1} \qquad \text{, jika menggunakan } \textit{rounding}.$$

Dengan adanya galat pembulatan ini maka sebenarnya representasi bilangan *x* pada komputer mempunyai nilai,

$$\text{float}\ x = 1+\varepsilon\ x$$

dengan float x menyatakan representasi dengan titik kambang, dan ε adalah galat pembulatan.

Contoh 1.15:

1) Mesin komputer CDC memakai sistem biner dengan akurasi titik kambang sampai 48 digit. Apabila komputer menggunakan cara *chopping* maka besarnya galat pembulatan,

$$-2^{-47} \le \varepsilon \le 0$$

2) Mesin komputer IBM memakai sistem heksadesimal dengan akurasi titik kambang sampai 6 digit. Apabila komputer menggunakan cara *rounding*, maka besarnya galat pembulatan,

$$-\frac{1}{2}\cdot 16^{-5} \le \varepsilon \le \frac{1}{2}\cdot 16^{-5}$$

3. Titik Kambang dengan Presisi (Ketepatan) Ganda

Sejauh ini akurasi komputer yang kita bicarakan adalah untuk komputer dengan presisi tunggal (*single precission*). Saat ini sebenarnya kebanyakan komputer sudah menggunakan presisi ganda (*double precission*). Di sini kita tidak akan membicarakan lebih jauh mengenai bagaimana kerja mesin komputer dengan presisi ganda, yang perlu kita ketahui adalah bahwa dengan presisi ganda akurasi komputer menjadi dua kali lebih baik. Misalkan dengan presisi tunggal suatu komputer mempunyai akurasi sampai 23 angka biner, maka dengan presisi ganda komputer tersebut mempunyai akurasi sampai kira-kira 46 angka biner.

C. PERAMBATAN GALAT

Kita ketahui bahwa penggunaan titik kambang menimbulkan galat pembulatan, maka operasi aritmatik dari titik kambang tersebut mengakibatkan terjadinya perambatan galat. Untuk jelasnya, marilah kita lihat operasi penjumlahan dari dua bilangan real x dan y yang bertanda (sama-sama positif atau negatif) pada mesin komputer. Pada komputer kita ketahui bahwa x dan y akan direpresentasikan dengan titik kambangnya. Misalkan $\tilde{x} = f(x)$ dan $\tilde{y} = f(y)$ adalah titik kambang dari x dan y dengan galat pembulatan ε_1 dan ε_2 dan operasi aritmatik + pada komputer direpresentasikan dengan + dengan galat relatif operasi + adalah ρ.

Dari pembahasan terdahulu kita ketahui bahwa $\tilde{x}$ dan $\tilde{y}$ dapat dituliskan sebagai:

$$\tilde{x} = x(1+\varepsilon_1)$$
$$\tilde{y} = y(1+\varepsilon_2)$$

dan operasi penjumlahan $\tilde{x}$ dan $\tilde{y}$ pada komputer adalah,

$$\begin{aligned}\tilde{x}+\tilde{y} &= (1-\rho)(\tilde{x}+\tilde{y}) \\ &= (1-\rho)\left(x(1+\varepsilon_1)+y(1+\varepsilon_2)\right) \\ &= (1-\rho)\left((x+y)+\varepsilon_1 x+\varepsilon_2 y\right)\end{aligned}$$

Kita dapat memisalkan suatu bilangan ϕ yang memenuhi $\phi(x+y) = \varepsilon_1 x+\varepsilon_2 y$, dan dapat dibuktikan $|\phi| \le \text{maks}(|\varepsilon_1|,|\varepsilon_2|)$, sehingga kita dapatkan,

$$\tilde{x} \oplus \tilde{y} = \; 1-\rho \quad 1+\phi \quad x+y$$

dan kemudian kita dapatkan perambatan galat relatif dari operasi penjumlahan ini,

$$\left| \frac{x+y \; - \; \tilde{x} \oplus \tilde{y}}{x+y} \right| = |\rho + \phi + \rho\phi| \le |\rho| + |\phi| + |\rho\phi|$$

Karena pada umumnya $|\rho| < 1$ dan $|\phi| < 1$, maka $|\rho\phi|$ menjadi tidak signifikan (sangat kecil), sehingga dapat kita tuliskan,

$$\left| \frac{x+y \; - \; \tilde{x} \oplus \tilde{y}}{x+y} \right| \approx |\rho| + |\phi|$$

dan ternyata besarnya perambatan galat kira-kira sama dengan jumlah galat relatif dari operasi $\oplus$ dan galat terbesar dari bilangan yang dioperasikan.

Pada Modul ini kita tidak akan membahas lebih jauh mengenai perambatan galat untuk operasi-operasi aritmatik yang lain, yaitu –, ÷ dan ×. Di sini hanya dituliskan hasil kesalahan relatif yang dihasilkan karena adanya perambatan galat karena operasi-operasi aritmatik tersebut. Bagi yang menginginkan pembahasan yang lebih mendalam dapat mempelajari sendiri dari buku-buku teks analisis numerik. Perambatan galat yang dihasilkan oleh operasi perkalian dapat ditulis sebagai,

$$\left| \frac{x \cdot y - \; \tilde{x} * \tilde{y}}{x \cdot y} \right| \approx |\sigma| + |\rho| + |\tau|$$

dengan ρ adalah galat relatif dari operasi perkalian (*) pada komputer, sedangkan σ dan τ adalah galat relatif dari titik kambang $\tilde{x}$ dan $\tilde{y}$ terhadap x dan y. Pada operasi pembagian, galat relatifnya dapat dituliskan sebagai,

$$\left| \frac{x : y - \tilde{x} / \tilde{y}}{x : y} \right| \approx |\sigma| + |\tau| + |\gamma|$$

dimana γ adalah galat relatif karena operasi pembagian (/) pada komputer.

Selain karena operasi aritmatik, perambatan galat dapat pula terjadi pada saat mengevaluasi suatu fungsi atau pada saat menghitung suatu sumasi. Untuk kedua hal ini dapat Anda ikuti pembahasannya pada akhir Modul 2 dan 7.

LATIHAN 2

Untuk memperdalam pemahaman Anda mengenai materi di atas, kerjakanlah latihan berikut!

1) Hitunglah $\sqrt{2}$ dan log 2 dengan menggunakan kalkulator dengan ketelitian sampai 6 angka di belakang tanda desimal, kemudian carilah galat relatif dari pendekatan x_t oleh x_a, jika:
   a) $x_t = \sqrt{2}$ dan $x_a = 1,414$
   b) $x_t = \log 2$ dan $x_a = 0,7$
2) Carilah signifikan dari pendekatan nilai e (bilangan natural) oleh $19/7$.
3) Suatu komputer menggunakan bilangan dasar 10 mempunyai presisi 3, $m = 10^{-5}$ dan $M = 10^5$. Komputer tersebut mempresentasi bilangan real x dan y dalam bentuk titik kambang $\tilde{x} = 0,371 \times 10^{-4}$ dan $\tilde{y} = 0,371 \times 10^4$.
   a) Berapa besar galat $|x - \tilde{x}|$ yang dapat terjadi?
   b) Berapa besar galat $|y - \tilde{y}|$ yang dapat terjadi?
   c) Berapa besar kesalahan relatif yang dapat terjadi?

Petunjuk Jawaban Latihan

1. a $x_t = \sqrt{2} = 1,414214$

$x_a = 1,414$

$$\varepsilon = \frac{|x_t - x_a|}{x_t} = \ldots.$$

b) Kerjakan seperti a.

2.a $e = 2,718282\ldots$

$$\frac{19}{7} \simeq 2,714285714285...$$

$$\varepsilon = |e - 19/7| = |2,718282\ldots - 2,714285...| = \ldots$$

$$\frac{1}{2}\cdot 10^{-k+1} \leq \varepsilon \leq \frac{1}{2}\cdot 10^{-k} \qquad \Rightarrow \qquad k = \ldots$$

3.a $|x - \tilde{x}| = \left|0,371\times 10^{-4} - 0,367\times 10^{-4}\right| = \ldots$

b $|y - \tilde{y}| = \left|0,371\times 10^{4} - 0,367\times 10^{4}\right| = \ldots$

c $\dfrac{|x - \tilde{x}|}{x} = \ldots$

$\dfrac{|y - \tilde{y}|}{y} = \ldots$

RANGKUMAN

Galat pada suatu perhitungan numerik adalah selisih antara nilai sebenarnya (yang dicari) dengan nilai pendekatan (yang didapat),

Galat = nilai sebenarnya – nilai pendekatan

Galat pembulatan adalah galat karena pembatasan banyaknya digit/angka yang harus kita tuliskan. Galat pembulatan dapat dilakukan dengan cara *chopping* atau *rounding*. Selain galat pembulatan, pada perhitungan numerik dapat pula terjadi galat pemenggalan, yaitu galat yang terjadi karena pemenggalan suku-suku dalam suatu deret.

Galat absolut dinyatakan dengan,

$$\varepsilon = |x - \tilde{x}|$$

dan galat relatif dinyatakan dengan,

$$\varepsilon_r = \frac{|x - \tilde{x}|}{|x|}$$

Pada perangkat lunak (*software*) untuk perhitungan numerik, pada umumnya bilangan-bilangan dinyatakan dalam bentuk titik kambang. Dalam bentuk titik kambang, suatu bilangan x dituliskan,

$$x = \sigma\ 0, b_1 b_2 b_3 \cdots b_n\ \beta^e$$

dimana $\sigma = +$ atau $-$ adalah tanda bilangan, $b_1, b_2, \cdots, b_n$ adalah digit/angka, β adalah bilangan dasar (basis) dan $\beta \geq 2$, dan e menyatakan pangkat. Pada penulisan di atas, n menunjukkan tingkat presisi dan $0, b_1 b_2 b_3 \cdots b_n$ disebut mantisa.

Suatu komputer yang menggunakan sistem bilangan dengan bilangan dasar β dan mempunyai akurasi sampai n digit dalam representasi titik kambangnya, akan mempunyai galat pembulatan ε yang besarnya,

$-\beta^{-n+1} \leq \varepsilon \leq 0$, jika menggunakan *chopping*

dan,

$-\frac{1}{2}\beta^{-n+1} \leq \varepsilon \leq \frac{1}{2}\beta^{-n+1}$, jika menggunakan *rounding*.

Dengan adanya galat pembulatan ini maka sebenarnya representasi bilangan x pada komputer mempunyai nilai,

$$\text{float}\ x = 1 + \varepsilon\ x$$

dengan float x menyatakan representasi dengan titik kambang, dan ε adalah galat pembulatan.

Perambatan galat yang terjadi karena operasi penjumlahan dapat dituliskan sebagai,

$$\left| \frac{x + y - \tilde{x} \oplus \tilde{y}}{x + y} \right| \approx |\rho| + |\phi|$$

dengan ρ adalah galat relatif pada operasi penjumlahan dan ϕ adalah galat terbesar dari bilangan yang dijumlahkan.

Perambatan galat yang dihasilkan oleh operasi perkalian dapat ditulis sebagai,

$$\left| \frac{x.y - \tilde{x} * \tilde{y}}{x.y} \right| \approx |\sigma| + |\rho| + |\tau|$$

dengan ρ adalah galat relatif dari operasi perkalian (*) pada komputer, sedangkan σ dan τ adalah galat relatif dari titik kambang $\tilde{x}$ dan $\tilde{y}$ terhadap x dan y.

Pada operasi pembagian, galat relatifnya dapat dituliskan sebagai,

$$\left|\frac{x:y-\tilde{x}/\tilde{y}}{x:y}\right| \approx |\sigma|+|\tau|+|\gamma|$$

dimana γ adalah galat relatif karena operasi pembagian (/) pada komputer.

TES FORMATIF 2

Pilihlah satu jawaban yang paling tepat!

1) Bilangan real $x = 0,5412873...$ didekati oleh $\tilde{x} = 0,5412865...$ akurat sampai ....
   A. 4 angka desimal
   B. 5 angka desimal
   C. 6 angka desimal
   D. 7 angka desimal

2) Bilangan real $x = 0,2784$ didekati oleh $\tilde{x} = 0,2697$ menghasilkan galat relatif sebesar ....
   A. 0,009
   B. 0,018
   C. 0,031
   D. 0,062

3) Bilangan desimal $x = 91,25$ jika dituliskan dalam bentuk titik kambang sistem heksadesimal adalah ....
   A. (0,B54) 16^{-1}
   B. (0,54B) 16
   C. (0,5B4) 16^{2}
   D. (0,54B) 16^{3}

4) Pendekatan $x = 0,251 \times 10^{-3}$ oleh $\tilde{x} = 0,249 \times 10^{-3}$ menghasilkan kesalahan relatif sebesar ....
   A. 0,2%
   B. 0,4%
   C. 0,6%
   D. 0,8%

Cocokkanlah jawaban Anda dengan Kunci Jawaban Tes Formatif 2 yang terdapat di bagian akhir modul ini. Hitunglah jawaban yang benar. Kemudian, gunakan rumus berikut untuk mengetahui tingkat penguasaan Anda terhadap materi Kegiatan Belajar 2.

$$\text{Tingkat penguasaan} = \frac{\textit{Jumlah Jawaban yang Benar}}{\textit{Jumlah Soal}} \times 100\%$$

Arti tingkat penguasaan: 90 - 100% = baik sekali
80 - 89% = baik
70 - 79% = cukup
< 70% = kurang

Apabila mencapai tingkat penguasaan 80% atau lebih, Anda dapat meneruskan dengan modul selanjutnya. **Bagus!** Jika masih di bawah 80%, Anda harus mengulangi materi Kegiatan Belajar 2, terutama bagian yang belum dikuasai.

Kunci Jawaban Tes Formatif

Tes Formatif 1

1) B
2) C
3) A
4) D
5) C
6) A
7) D
8) C

Tes Formatif 2

1) B
2) C
3) C
4) D

Daftar Pustaka

Chapra, Steven C. & Raymond P. Canale. (1988). *Numerical Methods for Engineers, 2nd ed*. Mc Graw – Hill Book Company.

Djauhari, Maman A. (1991). *Analisis Numerik I*. Buku Materi Pokok. Jakarta: Universitas Terbuka.

Hoffman, Joe D. (1992). *Numerical Methods for Engineers and Scintists*. Mc Graw-Hill Inc.

King, J. Thomas. (1984). *Introduction to Numerical Computation*. Mc Graw – Hill Book Company.

Ledermann, Walter & Robert F. Churchhouse. (1981). *Handbook of Applicable Mathematics, Volume III : Numerical Methods*. John Willey and Sons.