BAB III
SISTEM BILANGAN KOMPUTER DAN DASAR-DASAR MATEMATIKA

- Bahasan :
  - representasi bilangan dalam komputer dan implikasinya terhadap nilai perhitungan.
  - Sumber-sumber berbagai jenis galat (error) dan perambatannya dalam proses perhitungan.

BILANGAN BULAT

Desimal

- Sistem bilangan yang sering digunakan manusia, (simbol yg digunakan 0, 1 … 9)
- Juga disebut sistem basis 10 (basis 10 ; 10 sebagai basis)
- Bentuk umum

$$d_k d_{k-1} \ldots d_2 d_1 d_0 = d_k \times 10^k + d_{k-1} \times 10^{k-1} + \ldots + d_2 \times 10^2 + d_1 10^1 + d_0 \times 10^0 \quad (1.1)$$

Contoh :

$132 = 1 \times 100 + 3 \times 10 + 2 \times 1$

$= 1 \times 10^2 + 3 \times 10^1 + 2 \times 10^0$

Biner

- Sistem bilangan yang digunakan komputer (mengacu 2 keadaan : ada arus listrik, tidak ada arus listrik), simbol yang digunakan 0 dan 1
- Juga disebut sistem basis 2 (basis 2 ; 2 sebagai basis)
- Bentuk umum

$$b_k b_{k-1} \ldots b_2 b_1 b_0\ _{dua} = b_k \times 2^k + b_{k-1} \times 2^{k-1} + \ldots + b_2 \times 2^2 + b_1\ 2^1 + b_0 \times 2^0 \quad (1.2)$$

Contoh :

$$10101_{dua} = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0.$$
$$= 16 + 4 + 1 = 21 \text{ (dua puluh satu).}$$

Konversi Bilangan

- Diperlukan untuk menjembatani perhitungan sistem bilangan desimal (yang sering digunakan manusia) oleh komputer (yang hanya mengenal sistem biner)

- Persamaan (1,2) memperlihatkan cara mengubah bilangan biner ke representasi desimal, yaitu dengan melakukan perhitungan aritmetik pada ruas kanan persemaam tersebut.

- Persamaan (1,2) juga dapat kita gunakan untuk menyusun suatu algoritma yang effisien untuk mengubah bilangan bulat positif N dalam representasi biner, misal :

  $N = b_k \times 2^k + b_{k-1} \times 2^{k-1} + \ldots + b_1 \times 2^1 + b_0 \times 2^0$, maka

  $N/2 = b_k \times 2^{k-1} + b_{k-1} \times 2^{k-2} + \ldots + b_1 \times 2^0 + b_0/2$

  $N/2 = H_o + b_0/2$;

- $H_0 = b_k \times 2^{k-1} + b_{k-1} \times 2^{k-2} + ... + b_1 \times 2^0$ adalah bilangan bulat hasil pembagian N dengan 2. Dengan demikian, b_o adalah sisa pembagian tersebut.

- Sehingga
  $H_0/2 = b_k \times 2^{k-2} + b_{k-1} \times 2^{k-3} + ... + b_1/2 = H_1 + b_1/2$
- $H_1 = b_k \times 2^{k-2} + b_{k-1} \times 2^{k-3} + ... + b_1/2$ adalah bilangan bulat hasil pembagian H_0 dengan 2.

- Secara umum diperoleh :

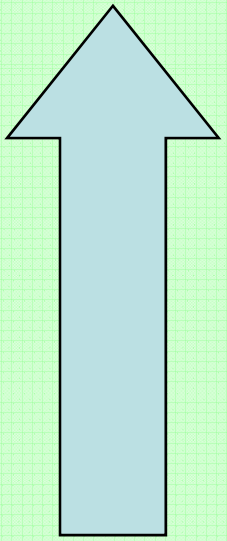
$$N = 2H_0 + b_0$$
$$H_0 = 2H_1 + b_1$$
$$H_1 = 2H_2 + b_2$$
$$\vdots$$
$$Hk\text{-}2 = 2Hk\text{-}1 + bk\text{-}1$$
$$Hk\text{-}1 = 2Hk + bk \quad (Hk = 0) \quad (1.3)$$

Contoh :

Ubahlah 132 ke dalam representasi binernya

Jawab : gunakan persamaan (1.3)

132	=	2 x 66 + 0	b0 = 0
66	=	2 x 33 + 0	b1 = 0
33	=	2 x 16 + 1	b2 = 1
16	=	2 x 8 + 0	b3 = 0
8	=	2 x 4 + 0	b4 = 0
4	=	2 x 2 + 0	b5 = 0
2	=	2 x 1 + 0	b6 = 0
1	=	2 x 0 + 1	b7 = 1.



Jadi 132 = 10000100_{dua}. ▮

Latihan 1.1:

1. Ubahlah bilangan bulat berikut ke dalam representasi desimal :
   a. 11011_{dua}
   b. 1110011_{dua}
   c. 10001000101_{dua}
2. Ubahlah bilangan desimal berikut ke dalam representasi biner :
   a. 21
   b. 103
   c. 2314

Bilangan Pecahan

- $x \in R$ disebut bilangan pecahan desimal jika $0<x<1$
- Bentuk umum pecahan desimal :

  $x = d_1 . 10^{-1} + d_2 . 10^{-2} + d_3 . 10^{-3} + ... + d_n . 10^{-n} + ...$ (1.4),

  $0 \leq di \leq 9$ untuk setiap i
- Biasanya kita menuliskan (1.4) sebagai

  $x = 0.d_1d_2d_3...$
- Jika ada bilangan bulat k sehingga $d_j=0$ untuk semua $j > k$, maka pecahan desimal itu kita katakan *berakhir*. Sebagai contoh,

  $1/2= 0.5 = 5 \times 10^{-1}$

  adalah pecahan desimal berakhir

- sedangkan,
  = 0.666... = $6 \times 10^{-1} + 6 \times 10^{-2} + 6 \times 10^{-3} + \ldots$ *bukan* pecahan desimal berakhir
- Kita juga dapat merepresentasikan $x \in (0,1)$ sebagai suatu pecahan biner :
  $x = b_1 . 2^{-1} + b_2 . 2^{-2} + b_3 . 2^{-3} + \ldots + b_n . 2^{-n} + \ldots . \quad (1.5)$, dimana semua $bi \in \{0,1\}$.
- Kita juga biasa menuliskan (1.5) sebagai,
  $x = 0.b1b2b3\ldots_{dua}$
- Algoritma untuk representasi biner dari suatu bilangan real $x \in (0,1)$. :
  - kita kalikan kedua ruas (1.5) dengan 2 sehingga diperoleh,
    $2x = b_1 + (b_2 . 2^{-1} + b_3 . 2^{-2} + \ldots + b_n . 2^{-n+1} + \ldots) \quad (1.6)$

perhatikan bahwa suku yang dikurung di ruas kanan (1.6) adalah pecahan antara 0 dan 1. Dengan demikian kita dapat mengambil bagian bulat dari kedua ruas (1.6) dan kita peroleh

$b_1 = int(2x)$ (1.7), int(2x) adalah bagian bulat dari bilangan real 2x

– Untuk mendapatkan b_2, kita misalkan,

$p_1 = b_2 . 2^{-1} + b_3 . 2^{-2} + ... + b_n . 2^{-n+1} + ...$ (1.8) yakni bagian pecahan dari 2x

Kalikan kedua ruas (1.8) dengan 2 sehingga kita peroleh,

$2p_1 = b_2 + (b3 . 2^{-1} + ... + b_n . 2^{-n+2} + ...)$ (1.9)

Dengan demikian, $b_2 = int(2p_1)$

– Proses ini harus dilanjutkan untuk mendapatkan b_3, b_4, dan seterusnya, diperoleh

$b_1 = \text{int}(2x) \rightarrow p_1 = 2x - b_1$

$b_2 = \text{int}(2p_1) \rightarrow p_2 = 2p_1 - b_2$

$b_3 = \text{int}(2p_2) \rightarrow p_3 = 2p_2 - b_3$

.

.

$b_n = \text{int}(2p_{n-1}) \rightarrow p_n = 2p_{n-1} - b_n$ (1.10)

- Carilah representasi biner dari x = 0.125

  Jawab

  Kita gunakan (1.10) :

  b1 = int(2x0.125) = int(0.25) = 0 → p1 = 0.25

  b2 = int(2x0.25) = int(0.5) = 0 → p2 = 0.5

  b3 = int(2x0.5) = int(1.0) = 1 → p3 = 0

  bj = 0 untuk setiap j > 3.

  Jadi, $0.125 = 0.001_{dua}$. ∎

- Contoh
  Carilah representasi biner dari x = 0.1

- Jawab
  b1 = int(2x0.1) = int(0.2) = 0 $\rightarrow$ p1 = 0.2
  b2 = int(2x0.2) = int(0.4) = 0 $\rightarrow$ p2 = 0.4
  b3 = int(2x0.4) = int(0.8) = 0 $\rightarrow$ p3 = 0.8
  b4 = int(2x0.8) = int(1.6) = 1 $\rightarrow$ p4 = 0.6
  b5 = int(2x0.6) = int(1.2) = 1 $\rightarrow$ p5 = 0.2

  Ternyata p5 = p2 sehingga kita tahu bahwa terjadi pengulangan angka $b_k = b_{k+4}$ untuk k = 2,3, … dalam representasi biner. Dengan demikian,
  $0.1 = 0.000110\underline{0011}\ldots_{dua}$. ▌ (Lambang berarti kelompok angka itu berulang).

Latihan 1.2

1. Ubahlah bilangan pecahan biner berikut ke dalam pecahan desimal :
   a. 0.110011_{dua}
   b. 0.001110001_{dua}
2. Ubahlah bilangan pecahan desimal berikut ke dalam pecahan biner :
   a. 0.214
   b. 0.856
3. Carilah bilangan biner yang menghampiri $\prod$ dengan selisih kurang dari 10^{-2}.

ARITMETIKA TITIK MENGAMBANG

- Dari uraian sebelumnya : dalam melakukan perhitungan menggunakan komputer seringkali nilai-nilai yang kita gunakan tidak dapat direpresentasikan secara pasti dan nilai yang dimanipulasi dalam komputer hanyalah nilai hampirannya. Penyebab utama adalah terbatasnya tempat penyimpanan bilangan dalam komputer. Besarnya perbedaan antara nilai hampiran dan nilai sesungguhnya disebut galat *(error)*.

- Pengertian Galat Mutlak dan Galat Nisbi

  Jika bilangan χ digunakan sebagai hampiran bagi bilangan pasti x, maka

  (1) yang dimaksud dengan galat mutlak adalah

  $e_m = |x - \chi|$

  $e_n = \left|\frac{x - \chi}{x}\right|$

  (2) yang dimaksud dengan galat relatif adalah:

- Perhitungan ilmiah dalam komputer biasanya dilakukan dalam aritmetika titik mengambang (*floating-point*). Suatu bilangan titik-mengambang yang terdiri dari n angka dengan basis β memiliki bentuk,

  $x = \pm (0.a_1a_2a_3...a_n).\beta^e$

  dimana

  $0.a_1a_2a_3...a_n$ (dalam basis β) disebut mantisa dan e (bilangan bulat dalam basis β) disebut eksponen.
- Bilangan titik-mengambang tersebut dikatakan dinormalkan jika $a_1 = 0$ atau jika $a_1 = a_2 = ... = a_n = 0$ (yakni jika mantisa = 0).
- Beberapa contoh bilangan berbentuk titik-mengambang adalah :

  (a) $(0.1100110011).2^{101110011}$ ($\beta = 2$, dinormalkan).

  (b) $(0.027934556).10^{13}$ ($\beta = 10$, tak dinormalkan).

  (c) $(0.FBAC).16^{BA}$ ($\beta = 16$, dinormalkan).

- Ketelitian (*precission*) atau panjang n bilangan titik-mengambang pada suatu komputer biasanya tergantung dari panjang word komputer tersebut dan berbeda dari satu merek komputer ke merek komputer lainnya. Komputer yang menerima program dalam bahasa FORTRAN biasanya menyediakan dua jenis ketelitian, yakni ketelitian tunggal (*single precision*) dan ketelitian ganda (*double precision*) yang panjangnya kira-kira dua kali lipat dari ketelitian tunggal.
- Besarnya eksponen e terbatas dalam selang

  $$m < e < M$$

  untuk nilai m dan M tertentu (tergantung dari merek komputer). Biasanya m = -M.
- Terdapat dua cara yang biasanya dilakukan dalam komputer untuk mengubah suatu bilangan real x ke bentuk titik-mengambangnya fl(x) dengan ketelitian n, yakni yang dikenal sebagai pembulatan (*rounding*) dan pemenggalan (*chopping*).
- Pada proses pembulatan, fl(x) ditentukan sebagai bilangan titik-mengambang yang dinormalkan yang terdekat dengan x.
- Pada proses pemenggalan, fl(x) ditentukan sebagai bilangan titik mengambang dinormalkan yang terdekat antara x dan 0.

Contoh

- Angggaplah bahwa kita menggunakan komputer dengan ketelitian 4 dan $\beta = 10$, maka

$$fl(\frac{5}{3}) = \begin{cases} (0.1667).10^1 & \text{jika dibulatkan} \\ (0.1666).10^1 & \text{jika dipenggal} \end{cases}$$

$$fl(\text{-}236.35) = \begin{cases} -(0.2364).10^3 & \text{jika dibulatkan} \\ -(0.2363).10^3 & \text{jika dipenggal} \end{cases}$$

- Cara manapun yang digunakan untuk mendapatkan fl(x), akan diperoleh nilai hampiran yang berbeda dari nilai sebenarnya. Galat yang ditimbulkan oleh proses ini disebut galat pembulatan (*round-off error*) yang biasanya diukur dengan konsep galat nisbi.

- Pada beberapa komputer perhitungan fl(x) ini diubah dalam kasus $|x| \geq \beta^M$ (disebut *overflow*). Pada kebanyakan jenis komputer terjadinya kasus *overflow* dianggap sebagai suatu kesalahan fatal dan biasanya program langsung dihentikan. Tetapi, kasus *underflow* tidak dianggap sebagai suatu kesalahan; bilangan titik-mengambang biasanya diganti dengan 0 dan program tetap dilanjutkan.

- Bila dilakukan operasi aritmetik misalnya penjumlahan terhadap dua bilangan titik-mengambang, maka biasanya bilangan hasil operasi tersebut tidak memiliki panjang mantisa yang sama dengan panjang bilangan-bilangan yang dioperasikan. Hal ini akan menimbulkan masalah tersendiri yang cukup serius. Untuk mendapatkan gambaran tentang apa yang terjadi, perhatikanlah contoh berikut ini.

Contoh

- Misalkan kita diberikan dua bilangan titik-mengambang dengan panjang mantisa masing-masing 2; $x = (0.11).10^2$ dan $y = (0.43).10^{-3}$. Maka

$x + y = (0.1100043).10^2$

- Bila penjumlahan ini kita lakukan dengan komputer yang ketelitiannya 4, maka hasil yang kita peroleh adalah :

  $$fl(x+y) = (0.1100).10^2$$

  Perhatikanlah, ternyata hasil ini sama dengan x; jadi seakan-akan kita sama sekali tidak melakukan penjumlahan $x + y$.
- Hasil penjumlahan akan benar bila komputer yang kita gunakan memiliki ketelitian sekurang-kurangnya 7.

- perlu ditekankan : komputer memiliki tempat terbatas untuk merepresentasikan bilangan, tidak semua bilangan nyata dapat direpresentasikan dalam komputer. Sebagai contoh, komputer yang menggunakan 32 bit (misalnya IBM AT 386) untuk merepresentasikan bilangan nyata dengan ketelitian tunggal menggunakan 8 bit untuk eksponen dan 24 bit untuk mantisa.

  Besarnya bilangan nyata yang dapat direpresentasikan ada dalam selang
  $(2.93876).10^{-39}$ sampai $(1.701412).10^{38}$, yakni 2^{-128} sampai 2^{127}.

- LATIHAN

1. Lakukanlah perhitungan di bawah ini menggunakan komputer dengan mantisa 4 bit. Apa yang terjadi ?
   a) (1/5 +1/3) +1/6
   b) (1/9+1/17) +3/7
   c) (7/10+1/9) + 1/7
2. Bilangan-bilangan berikut diberikan dalam komputer desimal dengan mantisa 4 angka yang dinormalkan :
   $x=0.4523.10^{4}$, $y=0.2115.10^{-3}$, $z=0.2583.10^{1}$

Lakukanlah perhitungan berikut dan hitung galat pada hasil akhir bila dilakukan pembulatan :

a. x + y + z
b. x/y
c. x – y
d. x – y – z
e. xy/z
f. (y/z)x

3. Tulislah algoritma untuk mengubah bilangan bulat desimal ke dalam representasi oktal (bilangan dengan basis 8) .
4. Tulislah algoritma untuk mengubah bilangan pecahan desimal ke dalam representasi pecahan oktal.

KEHILANGAN ANGKA SIGNIFIKAN

- Satu hal yang dapat memperbesar peranan galat dalam perhitungan menggunakan komputer adalah *kehilangan angka signifikan*. Jika x* adalah nilai hampiran bagi *x*, maka kita katakan x* menghampiri *x* sampai *k* angka signifikan asalkan galat mutlak paling besar pada angka signifikan ke-*k* dari *x*. Sebagai teladan, x* 3 menghampiri x= π sampai satu angka signifikan, sedangkan x*= 22/7 sesuai dengan x= π sampai 3 angka signifikan.

- Kehilangan angka signifikan sering terjadi bila kita mengurangkan dua bilangan yang hampir sama baik besarnya maupun tandanya.

  Perhatikanlah teladan berikut

  Anggaplah bahwa

  $X^*=(0.22345683).10^1$ dan $y^*=(0.22345512).10^1$ masing-masing adalah nilai hampiran bagi *x* dan *y* sampai 7 angka signifikan. Maka, dengan menggunakan komputer yang memiliki ketelitian 8, kita peroleh

  $Z^*=x^*-y^* =0.000001710.10^{!} = (0.171).10^{-4}$

yang menghampiri $z = x - y$ hanya sampai 2 angka yang signifikan karena angka ke-3 pada mantisa berasal dariangka ke-8 dari *x* dan *y* yang bukan angka signifikan (mungkin mengandung galat) .

- Sebagai akibat kehilangan sebanyak 5 angka signifikan di atas, galat nisbi dalam z* mungkin 10^5 kali galat nisbi dalam *x** atau y*, walaupun galat mutlaknya paling besar adalah jumlah galat dalam x* dan *y**.
- Dari Contoh di atas, agar galat nisbi kecil, maka kita harus mencegah terjadinya kehilangan angka signifikan.
- Kehilangan angka signifikan seringkali dapat dicegah dengan mengubah bentuk rumus pengurangan menjadi bentuk lain yang senilai.
- Contoh 1.7

  Misalkan kita akan menghitung

Untuk nilai *x* yang sangat besar (misalkan *x* = 12346) . Bila kita menggunakan komputer dengan ketelitian 6 secara langsung dengan rumus tersebut, maka

$$y = \sqrt{12346} - \sqrt{12345}$$

= $(0.111113) \cdot 10^3$ - $(0.111108) \cdot 10^3$
= $(0.5) \cdot 10^{-2}$

padahal hasil sesungguhnya adalah : *y* = 0.00450003262627 75

Dengan demikian, galat nisbi dari hasil pengurangan secara langsung di atas adalah 10 persen.

- Cara yang jauh lebih baik untuk mencegah kehilangan angka signifikan (dan mengurangi galat nisbi) adalah dengan mengubah rumus tersebut sebagai berikut :

$$y = \sqrt{x} - \sqrt{x-1}$$

$$= \left(\sqrt{x} - \sqrt{x-1}\right) \cdot \frac{\sqrt{x} + \sqrt{x-1}}{\sqrt{x} + \sqrt{x-1}} \quad = \frac{1}{\sqrt{x} + \sqrt{x-1}}$$

untuk x = 12346, maka kita peroleh :

$$y = \frac{1}{\sqrt{12346} + \sqrt{12345}} \quad = \frac{1}{(0.222221) \cdot 10^3}$$

$$= \left(0.450002\right) \cdot 10^2$$

Dengan cara ini, galat nisbi hanya 0.0003 persen.