Transformation Towards the Ultimate

Kampus Merdeka INDONESIA JAYA

TIF1101 – Dasar-Dasar Pemrograman

HO 04 - Jenis Data

Opim Salim Sitompul

Department of Information Technology
Universitas Sumatera Utara

Outline I

Outline II

- Jenis Data **double**

8 Jenis Data Buatan Pemrogram
- Enum
- typedef

9 Konversi jenis data
- Promosi
- Demosi
- Casting

Pendahuluan

- Peranan data dalam sebuah program adalah sangat besar sekali.
- Jika pemberian dan penggunaan data tidak dilakukan secara benar, maka hasil pengolahan yang diperoleh juga tidak akan benar.
  - GIGO (*garbage in, garbage out*)

Kata Kunci (*Keyword*)

- Komponen pembentuk pernyataan (*statement*).
- Memiliki arti khusus bagi sebuah bahasa pemrograman.
- Digunakan untuk maksud-maksud yang telah tertentu pula.
- Penggunaannya harus sesuai dengan yang dimaksudkan, tidak boleh diubah.

Kata Kunci (*Keyword*)

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Pengenal (*Identifier*)

- Kata yang digunakan oleh pemrogram untuk maksud-maksud tertentu yang diperlukan dalam menyusun program.
- Mengacu ke nama variabel, konstanta bernama, nama fungsi, dll.
- Terdiri dari kombinasi antara huruf, angka, dan garis bawah yang karakter pertamanya harus huruf atau garis bawah.

Pengenal (*Identifier*)

- Contoh Yang Sah:

```
1 indeks     /* nama variabel */
2 yN     /* nama variabel */
3 PI       /* konstanta bernama */
4 cariHuruf() /* nama fungsi */
5 maksArr[100]  /* nama array */
```

Pengenal (*Identifier*)

- Contoh Yang Tidak Sah:

```
1 luas lingkaran    /* tidak boleh ada spasi
    */
2 nomor#       /* Ada karakter khusus # */
3 2kali      /* Dimulai dengan angka */
```

Konstanta (*Constants*)

- Data objek yang memiliki nilai tertentu yang tidak dapat diubah selagi program dijalankan.
- Ada 2 jenis:
  - Konstanta buatan pemrogram (*programmer-defined constant*).
    - Harus memiliki nama:
    - Contoh: PI, MAX, MIN
  - Konstanta literal (*literal constant*)
    - Representasi tertulis dari nilainya
    - Contoh: 5 100 10.2 4.3 ‘A’ “Lapar”

Konstanta (*Constants*)

- Konstanta juga memiliki jenis data tertentu:
  - karakter
  - integer
  - floating point
  - string
- Konstanta integer dibagi lagi atas tiga kelompok:
  - desimal
  - oktal
  - heksadesimal
- Konstanta floating point terbagi atas:
  - float
  - double

Konstanta Karakter

- Diapit oleh tanda petik tunggal (*single quote*)
  - Contoh: ’a’ ’A’ ’0’ ’&‘ ‘ ‘
- Karakter-karakter yang tidak dapat dicetak (karakter *escape* - yaitu karakter yang diawali oleh *backslash*)
  - Contoh: ’\r‘ ’\a‘ ’\n‘ ’\0‘ ’\\‘ ’\“ ’\“‘

Konstanta Karakter

- C membagi karakter atas 5 kelompok:
  1. Huruf:
     'a', 'b', ..., 'z', 'A', 'B', ..., 'Z'
  2. Angka:
     '0', '1', ..., '9'
  3. Karakter khusus:
     ',' '.' ';' '?' '$' '!' '%' ...
  4. Karakter *escape*:
     '\r', '\n', '\0', '\b'
  5. Karakter *whitespaces*:
     spasi, tabulator, *carriage return*, *newline*, *form feed*

Konstanta Integer

- Konstanta Integer Desimal:
  1708 +2610 +612 -92
- Konstanta Integer Oktal:
  057 011 040 0463
- Konstanta Integer Hexadesimal:
  0x37 0xFFFF 0x9000 0Xabef

Konstanta Floating Point

- Direpresentasikan dengan sebuah tanda positif atau negatip, diikuti oleh bagian integer, titik desimal, bagian fraksional, dan bagian eksponensial.
- Contoh:
  1.5 235.67 .001 2.1E5 1000.1F 49822.0L
- Konstanta floating-point terdiri dari dua jenis:
  float dan **double**.
- 4.15F /* **float**: floating point ketelitian tunggal */
  7.78L /* **double**: floating point ketelitian ganda, long */

Konstanta String

- Untaian karakter yang diapit oleh tanda kutip ganda
- Dapat berupa huruf, angka, karakter khusus, dan/atau spasi
- Pada akhir string, kompiler akan menambahkan karakter nol ('\0') untuk menandai akhir string.
  - Contoh:
    "September" "Komputer\n" "a" " " "!"

Konstanta Bernama

- Konstanta bernama dapat dideklarasikan dengan dua cara:
  - Pendefinisian melalui pengarah **#define**:
    - Contoh:
      #define MAKS_DATA 1000
      #define NAMAFILE “databaru.txt”
      #define EXP 2.71828
  - Pendeklarasian melalui kata kunci **const**:
    - Contoh:
      const int MAKS_DATA = 1000;
      const double EXP = 2.71828;

Variabel

- Objek data yang didefinisikan dan dinamai oleh pemrogram secara eksplisit.
- Menempati satu lokasi memori yang disisihkan untuk satu jenis data tertentu dan diberi nama untuk memudahkan pengacuannya.
- Menyatakan lokasi memori yang dapat menampung jenis data tertentu dengan nilai yang berbeda-beda.

Deklarasi Variabel

- Pendeklarasian variabel berguna untuk:
  - Memberitahu kompiler tentang jenis variabel yang digunakan.
  - Memberikan daftar nama variabel yang digunakan.
  - Menyediakan lokasi memori tempat menampung nilai yang diberikan pada variabel.
- Bentuk Umum:
  jenis_data nama_variabel;

Daftar Variabel

Jenis	Byte	Bit	Range
short int	2	16	-32,768 s.d. +32,767 (32kb)
unsigned short int	2	16	0 s.d. +65,535 (64Kb)
unsigned int	4	32	0 s.d. +4,294,967,295 (4Gb)
long int (int)	4	32	-2,147,483,648 s.d. +2,147,483,647 (2Gb)
long long int	8	64	-2^{63} s.d. $(2^{63} - 1)$
Signed char	1	8	-128 s.d. +127
Unsigned char	1	8	0 s.d. +255
Float	4	32	3.4E-38 s.d. 3.4E+38
double	8	64	1.7E-308 s.d. 1.7E+308
long double	12	96	3.4E-4932 s.d. +1.1E+4932 (19 angka ketelitian)

Jenis Data **char**

- Digunakan untuk menyajikan karakter.
- Menempati memori sebesar 1 byte.
  - Satu byte terdiri dari 8 bit:
  - Menampung $2^8 = 256$ nilai yang berbeda.
- Sesuai dengan jumlah karakter ASCII (0 hingga 255)
- Contoh:
  char ch;
  unsigned char huruf1, huruf2;
  unsigned char tblKar[80];

Jenis Data **char**

```
1  /* Nama file: AscKar.c
2     Konversi karakter ke kode ASCII  */
3  #include <stdio.h>
4
5  int main()
6  {
7    char c;
8
9    /* Minta sebuah karakter */
10   printf("Berikan sebuah karakter: ");
11   scanf("%c", &c);
12   /* Tampilkan Kode ASCII */
13   printf("Kode ASCII %c = %d.\n", c, c);
14
15   return 0;
16 }
```

Jenis Data **int**

- Menyajikan nilai integer (bilangan bulat)
- Tersimpan di dalam memori sebesar 4 bytes (32 bit)
- Terbagi lagi atas tiga jenis:
  1. **short int** (2 bytes)
  2. **int** dan **long** (4 bytes)
  3. **long long** (8 bytes)
- Dibagi dalam dua kelompok: **signed** dan **unsigned**.
  - Bilangan yang dapat ditampung oleh **unsigned int** adalah $2^{32} = 4,294,967,295$ (4GB)
  - Bilangan yang dapat ditampung oleh **signed int** terbagi dalam *range*:
    -2,147,483,648 s/d -2,147,483,647

Jenis Data **int**

- Contoh:
  int i, j, k ; /* signed integer */
  short a, b, c ; /* short integer */
  long fibonacci ; /* long integer */
  long long kel_dunia; /* long long integer */
  unsigned umur; /* integer tak bertanda */
  unsigned short usia; /* short tak bertanda */
  unsigned long fibo; /* long int tak bertanda */

Jenis Data **int**

```
1 /* Nama file: konversi1.c
2    Konversi suhu dari Celcius ke Fahrenheit
       */
3 #include <stdio.h>
4 int main()
5 {
6   int Celcius, Fahrenheit;
7   /* Minta temperatur celcius */
8   printf("Berikan temperatur Celcius: ");
9   scanf("%d", &Celcius);
```

Jenis Data **int**

```
10    /* Hitung temperatur Fahrenheit */
11    Fahrenheit = (9 * Celcius) / 5 + 32;
12    /* Tampilkan temperatur */
13    printf("%d Celcius = %d Fahrenheit.\n",
          Celcius, Fahrenheit);
14
15    return 0;
16  }
```

Jenis Data **float**

- Menampung bilangan-bilangan riil.
- Bilangan riil adalah bilangan yang memiliki bagian desimal, titik, bagian fraksional dan bagian eksponensial.
- Contoh:
  float volume_bola;
  float laba1, laba2;

Jenis Data **float**

```
1  /* Nama file: konversi2.c
2     Program mengkonversi suhu dari Fahrenheit
          ke Celcius */
3  #include <stdio.h>
4
5  int main()
6  {
7    float Celcius, Fahrenheit;
8
9    /* Minta temperatur fahrenheit */
10   printf("Berikan temperatur Fahrenheit: ");
11   scanf("%f", &Fahrenheit);
```

Jenis Data **float**

```
13    /* Hitung temperatur Celcius */
14    Celcius = (5.0F/9.0F) * (Fahrenheit - 32);
15    /* Tampilkan temperatur*/
16    printf("%f Fahrenheit = %f Celcius.\n",
          Fahrenheit, Celcius );
17
18    return 0;
19  }
```

Jenis Data **double**

- Kelompok *floating point*, tetapi memiliki angka ketelitian ganda (*double precision*)
- Menempati 8 byte memori
- Dapat pula berjenis **long double**, menempati 10 byte memori.
- Contoh:
  double luas;
  long double sinus_x;

Jenis Data **double**

```
1  /* Nama file: radioactive.c
2     Menghitung sisa peluruhan radio aktif */
3  #include <stdio.h>
4  #include <math.h>
5
6  int main()
7  {
8    double JlhAwal, ParuhHidup, Waktu, Sisa;
9
10   printf("Berikan jumlah awal (mg) zat, ");
11   printf(" paruh-hidupnya (hari) dan ");
12   printf("waktu (hari) untuk mencari jumlah
        yang tersisa: ");
```

Jenis Data **double**

```
14    scanf("%lf %lf %lf", &JlhAwal, &ParuhHidup
          , &Waktu);
15    Sisa = JlhAwal * pow(0.5, (Waktu /
          ParuhHidup));
16    printf("Jumlah yang tersisa = %lf mg.\n",
          Sisa);
17
18    return 0;
19  }
```

Jenis Data Buatan Pemrogram

- Jenis data yang didefinisikan oleh pemrogram (*user defined types*) berdasarkan jenis data dasar.
  - **enum**
  - **typedef**

Enum

- Bentuk Umum:
  enum *pengenal* {daftar enumerator};
- Contoh:
  enum *bln* {jan, feb, mar, apr, mei, jun, jul, agt, sep, okt, nop, des};
- Enumerasi dimulai dari 0, jadi jan=0, feb=1, ...

Enum

- Nilai *default* enumerator dapat diubah sesuai kebutuhan.
- Contoh:
  enum *bln* {jan=1, feb, mar, apr, mei, jun, jul, agt, sep, okt, nop, des};
- Dengan demikian enumerasi dimulai dari 1, jadi jan=1, feb=2, . . .
- Variabel berjenis enum dapat dideklarasikan sbb.:
  enum *bln* bln_sekarang, bln_lalu;

Enum

```
1  /* Nama file: enumerasi.c
2     Contoh penggunaan jenis data enum */
3
4  #include <stdio.h>
5  /* Definisi jenis data enum */
6  enum bln {jan=1,feb,mar,apr,mei,jun,jul,agt,
       sep,okt,nop,des};
7
8  int main()
9  {
10   /* deklarasi variabel berjenis enum */
11   enum bln bln_sekarang, bln_kemarin;
```

Enum

```
12    bln_sekarang = okt;
13    bln_kemarin = bln_sekarang - 1;
14    printf("Sebelum bulan ke-%d adalah bulan
          ke-%d.\n", bln_sekarang, bln_kemarin);
15    return 0;
16  }
```

typedef

- Memberikan nama baru pada jenis data dasar.
- Bentuk umum:
  typedef *jenis_data* pengenal;
- Contoh:
  typedef *float* bilangan;
  typedef *double* Resistance;

typedef

```
1  /* Nama file: circuit.c
2     Menganalisa sirkuit sederhana */
3  #include <stdio.h>
4
5  typedef double Resistance;
6
7  int main()
8  {
9    Resistance R1, R2, R3;
10   double TotalR, Voltase, Arus;
11
12   /* Minta tiga resistan dan voltase */
13   printf("Berikan tiga resistan (Ohm): ");
14   scanf("%lf %lf %lf", &R1, &R2, &R3);
```

typedef

```
15    printf("Voltase yang digunakan (volt): ");
16    scanf("%lf", &Voltase);
17    /* Hitung Total Resistan dan Arus */
18    TotalR = 1.0/(1.0/R1 + 1.0/R2 + 1.0/R3);
19    Arus = Voltase / TotalR;
20    /* Tampilkan Arus */
21    printf("Arusnya = %lf amps.\n", Arus);
22
23    return 0;
24 }
```

Konversi jenis data

- Menggunakan jenis data yang berbeda-beda di dalam sebuah program memberikan potensi terjadinya kesalahan, antara lain:
  - Kesalahan pembulatan (*round-off error*)
    - Perbedaan antara aproksimasi perhitungan pada sebuah bilangan dan nilai eksak matematisnya.
    - Rounding:
      $23.5 \rightarrow 24$
      $-23.5 \rightarrow -23$
  - Kesalahan pemotongan *(truncation error*)
    - Membatasi jumlah digit di sebelah kanan titik desimal dengan membuang *least significant digit*.
    - Contoh: pemotongan ke 4 digit desimal:
      $5.634143 \rightarrow 5.63$
      $32.438191 \rightarrow 32.43$
      $-6.344444 \rightarrow -6.3$

Promosi

- Konversi dari jenis data yang lebih kecil ke jenis data yang lebih besar.
  - int $\rightarrow$ float
  - float $\rightarrow$ double

Promosi

```
1  #include <stdio.h>
2
3  int main()
4  {
5    int x;
6    float y;
7
8    x = 100;
9    y = x;
10
11   printf("x=%d, y=%f\n", x, y);
12
13   return 0;
14 }
```

Demosi

- Konversi dari jenis data yang lebih besar ke jenis data yang lebih kecil:
  - float $\rightarrow$ int
  - double $\rightarrow$ float

Demosi

```
1  #include <stdio.h>
2
3  int main()
4  {
5    int x;
6    float y;
7
8    y = 7.6;
9    x = y;
10
11   printf("x=%d, y=%f\n", x, y);
12
13   return 0;
14 }
```

Casting

- Konversi jenis data yang dilakukan secara sengaja dengan mengubah nilai yang akan dikonversi ke jenis yang diinginkan.
- Bentuk umum:
  (*jenis_data*) pengenal;

Casting

```
1  #include <stdio.h>
2
3  int main()
4  {
5    int x;
6    float y;
7
8    x = 100;
9    y = (float) x;
10
11   printf("x=%d, y=%f\n", x, y);
12
13   return 0;
14 }
```

Casting

```
1  #include <stdio.h>
2
3  int main()
4  {
5    int x;
6    float y;
7
8    y = 7.6;
9    x = (int) y;
10
11   printf("x=%d, y=%f\n", x, y);
12
13   return 0;
14 }
```