Komentar, Type Data, Variable dalam Pemrograman C

Komentar

- Komentar berfungsi untuk memudahan seorang programmer memebaca alur sebuah program.
- Komentar diawali /* dan diakhiri */, kadang kala diawali //→ hanya untuk 1 baris komentar
- Macam-macam dan contoh komentar:
  - /*ini komentar 1 baris*/
  - // ini koemtar 1 baris
  - int /*ini komentar sisipan*/a;
  - /* ini adalah
    komentar 2 baris */

Mengenal Tipe Data[1]

- Tipe data menentukan secara internal data itu dapat disimpan. → mendefinisikan pola struktur informasi.
- Tipe data menentukan data yang dapat disimpan, contoh:
  - Bilangan bulat disimpan pada type data int
  - Bilangan real disimpan pada type data float / double
  - Tulisan/Teks atau kata-kata disimpan pada type data char

Mengenal Tipe Data[2]

- Mendefinisikan TYPE berarti :
  - menentukan **nama type** dalam kamus,
  - definisi **domain harga** yang dapat dipunyai oleh nama tersebut,
  - konvensi atau perjanjian tentang penulisan **konstanta** bertype tersebut,
  - **operator** yang dapat dioperasikan terhadap objek bertype tersebut.
- Ada dua macam tipe data:
  - Tipe data **dasar**
  - Tipe data **bentukan**

Tipe Data Dasar pada C

Tipe	Keterangan
char	Menyatakan karakter, contoh: A, e, h, J, 1, atau *
double	Menyatakan bilangan real dengan tingkat ketelitian 15 digit atau 10^{-308} sampai 10^{-308}
float	Menyatakan bilangan real dengan tingkat ketelitian 7 digit atau 10^{-38} sampai 10^{-38}
int	Menyatakan bilangan bulat dari -32768 sampai +32767
long int	Menyatakan bilangan bulat dari -217483648 sampai +217483647
bool	Menyatakan nilai logika: **true** atau **false**

Untuk detil: Diktat_Prosedural.pdf (pp: 33-39)

Tulisan VS Bilangan

- Tipe data yang berisi sebuah tulisan/teks/kata adalah tipe data karakter (char) atau String.
  - Penulisanya **diapit** oleh tanda " "
  - Contoh: "ini adalah tulisan"
- Tipe data bilangan merupakan tipe data yang berisi angka/ bilangan bulat dan bilangan real.
  - Penulisanya **tidak diapit** tanda ""
  - Contoh: 10

Mengenal Pengenal/Identifier[1]

- Pengenal / Identifier adalah nama yang digunakan untuk variable, fungsi, dst.
- Dalam bahasa C, pemberian nama / pengenal / identifier memiliki beberapa ketentuan, yaitu:
  - Harus diawali dengan huruf atau simbol garis bawah "_"
  - Sisanya dapat berupa huruf, angka, atau simbol garis bawah "_"
  - Huruf kecil dan huruf besar dibedakan / case-sensitive
    - Harga ≠ harga ≠ HARGA

Mengenal Pengenal/Identifier [2]

- Contoh nama pengenal yang benar:
  - n
  - n_buah
  - kuartal2
  - jumlahPenduduk
  - _n
- Contoh nama pengenal yang salah:
  - 2buah → tidak boleh diawali angka
  - jumlah penduduk → tidak boleh ada tanda spasi
  - jumlah* → tidak boleh mengandung tanda bintang "*"

Variable dan Variable Statis

- Variable menyatakan suatu nama yang dapat digunakan untuk memberikan nilai didalamnya dan nilai didalamnya dapat diubah sewaktu-waktu.
  - Jika variable tersebut berupa string maka ditulis dengan diapit tanda (")
  - Contoh : char nama[30]="Amir";
- Variable Statis merupakan variable yang **menyimpan nilai secara permananen**.
  - Kata kuncinya : **static**
  - Contoh penulisan dalam bahasa C: static float PI=3.14;

Lingkup Variable

Variable Global

- Adalah variable yang didklarasikan diluar fungsi, baik fungsi utama, maupun fungsi pendukung.

Variable Lokal

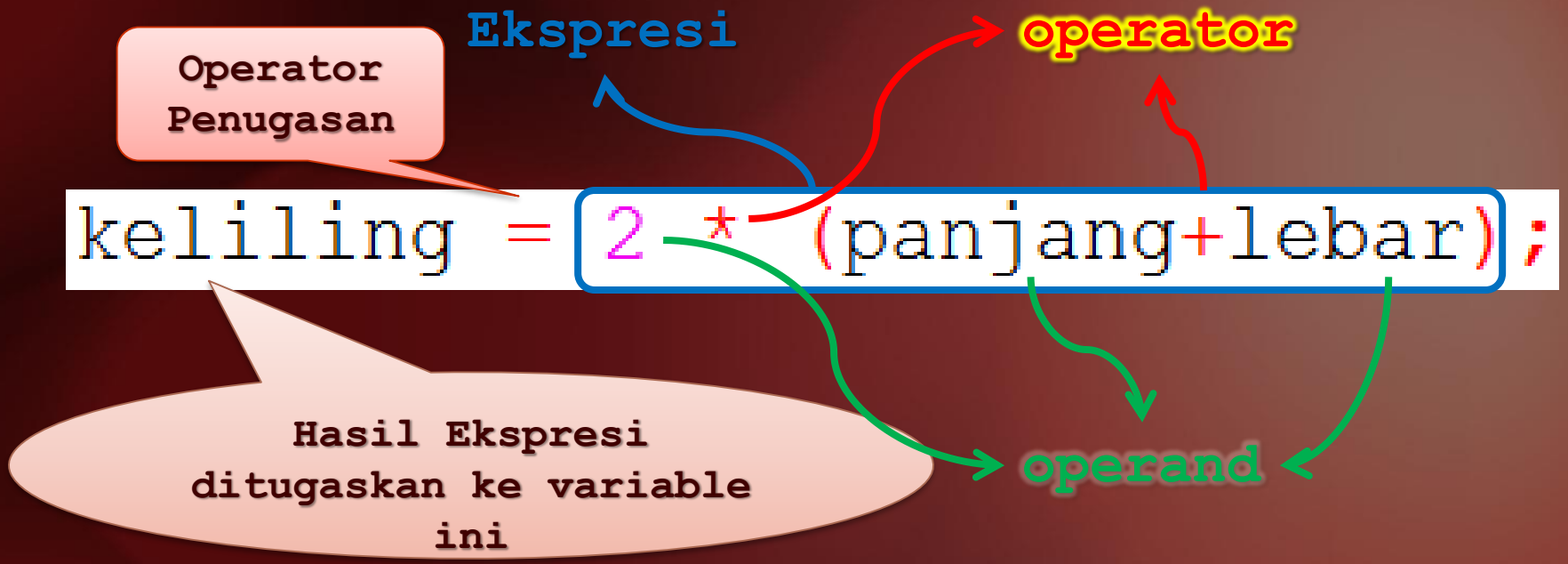
- Adalah variable yang dideklarasi dalam fungsi, jadi hanya dapat diakses oleh fungsi itu sendiri.

Contoh Lingkup Variable

vGlobalLokal.c

```
7   Spesifikasi      : menampilkan variable global dari
8                       fungsi utama dan fungsi pendukung
9   ------------------------------------------------*/
10  #include<stdio.h>
11  /* KAMUS (Global)*/
12  int x=10; //variable Global
13  //fungi pendukung --> ini adalah komentar
14  void TampilNilaiX () {
15      /* KAMUS (lokal-fungsi TampilNilaiX)*/
16      /* ALGORITMA (lokal-fungsi TampilNilaiX)*/
17      printf ("X dipanggil dari fungsi TampilNilaiX :%d\n", x);
18  }
19  //fungi utama --> ini adalah komentar
20  int main() {
21      /* KAMUS (lokal-fungsi utama)*/
22      int y=10;//variable Lokal
23      /* ALGORITMA (lokal-fungsi utama)*/
24      TampilNilaiX (); // memanggi fungsi TampilNilaiX ()
25      printf ("X dipanggil dari fungsi utama :%d\n", x);
26      printf ("y merupakan variable lokal fungsi main :%d\n", y);
27      return 0;
28  }
```

Ekspresi, Operator, Operand, Operator Penugasan



Mengenal Operator

Operator	Keterangan	Contoh	Prioritas
-	Tanda minus	-2	1
+	Tanda plus	+45	1
*	Perkalian int*double=double	7*3→21 7*3.0→21.0	2
/	Pembagian int/double=double	9/2→4 9/2.0→4.5	2
%	Sisa pembagian (khusus untuk int)	7%2→1 8%3→2	2
+	Penjumlahan int+double=double	7+2→9 7+2.0→9.0	3
-	Pengurangan int-double=double	7-2→5 7-2.0→5.0	3

Prioritas Operator

Prioritas Operator [2]

Increment VS Decrement

- Increment → penambahan nilai sebanyak 1, contoh: x++;
  - Pernyataan diatas sebanding dengan x=x+1;
- Decrement → pengurangan nilai sebanyak 1, contoh: x--;
  - Pernyataan diatas sebanding dengan x=x-1;
- **Pre Incremen/Decrement** : menaikan / menurunkan nilai yang terdapat pada sebuah variable **sebelum nilai diproses**.
- **Post Incremen/Decrement** : menaikan / menurunkan nilai yang terdapat pada sebuah variable **setelah nilai diproses**.

Pre/Post Increment

- Lihat contoh dibawah ini

```
nilai dari i :3
nilai dari j :3
nilai dari i (j=i++) :4
nilai dari k :5
nilai dari i (k=++i) :5
```

```c
#include<stdio.h>
int main() {
    /* KAMUS*/
    int i,j,k;
    /* ALGORITMA*/
    i=3; // i diisi 3
    printf ("nilai dari i :%d\n", i); // menampilkan nilai i
    j=i++;//j diisi nilai i, i ditambah/diincrement
    printf ("nilai dari j :%d\n", j);//menampilkan nilai j
    printf ("nilai dari i (j=i++) :%d\n", i); // menampilkan nilai i
    k=++i;// k diisi nilai i yang telah diincrement
    printf ("nilai dari k :%d\n", k); // menampilkan niali k
    printf ("nilai dari i (k=++i) :%d\n", i); // menampilkan nilai i
    return 0;
}
```

Post Increment

Pre Increment

Pre/Post Decrement

- Lihat contoh dibawah ini

```
nilai dari i :3
nilai dari j :3
nilai dari i (j=i++) :2
nilai dari k :1
nilai dari i (k=++i) :1
```

```c
#include<stdio.h>
int main() {
    /* KAMUS*/
    int i,j,k;
    /* ALGORITMA*/
    i=3; // i diisi 3
    printf ("nilai dari i :%d\n", i); // menampilkan nilai i
    j=i--;//j diisi nilai i, i ditambah/didecrement
    printf ("nilai dari j :%d\n", j);//menampilkan nilai j
    printf ("nilai dari i (j=i++) :%d\n", i); // menampilkan nilai i
    k=--i;// k diisi nilai i yang telah didecrement
    printf ("nilai dari k :%d\n", k); // menampilkan niali k
    printf ("nilai dari i (k=++i) :%d\n", i); // menampilkan nilai i
    return 0;
}
```

Post Decrement

Pre Decrement

Tipe Enumerasi

- Merupakan tipe yang nilainya telah didefinisikan secara pasti saat pemebentukan.
- Menggunakan kata kunci **enum**, berikut adalah bentuk umumnya:
  - enum nama_enumerasi {nilai1, nilai2, ...}
- Conntoh:
  - enum jenisKelamin {pria, wanita};
  - enum warnaPrimer {merah, hijau, biru};
  - enum namaHari {minggu, senin, selasa, rabu, kamis, jumat, sabtu};

Notasi Tipe Enumerasi

Notasi Algoritmik	Bahasa C
KAMUS **{ Definisi type }** **type** hari : (senin, selasa, rabu, kamis, jumat, sabtu) **{ Deklarasi variable }** H : Hari	**/* KAMUS */** **/* Definisi type */** typedef enum { senin, selasa, rabu, kamis, jumat, sabtu } hari; /* senin=1, selasa=2, rabu=3, dst. */ **/* Deklarasi variable */** Hari H;
ALGORITMA **{ Assignment }** H ← senin	**/* ALGORITMA */** **/* Assignment */** H = senin; /* H = 1 */

enumerasi.c

- Untuk mendefinisikan enumerasi bernama JenisKelamin

```c
#include<stdio.h>
#include<string.h> // utk memasukan fungsi strcpy
enum JenisKelamin {pria, wanita}; // pria=0, wanita=1
int main() {
    /* KAMUS*/
    char nama[10];
    enum JenisKelamin gender;
    /* ALGORITMA*/
    //nama="Mira"; tidak bisa digunakan
    strcpy (nama,"Mira");// fungsi utk mengisi nama dengan "Mira"
    gender = wanita;// dapat diisi dengan 1
    printf ("nama :%s\n", nama); // menampilkan nilai nama
    printf ("gender :%d\n", gender); // menampilkan nilai nama
    return 0;
}
```

Tipe Bentukan

- Type bentukan adalah suatu TYPE yang dirancang/dibentuk (dan diberi nama) dari beberapa komponen bertype tertentu.
- Type bentukan dibuat/didefinisikan karena keseluruhan (hasil komposisi) komponen type tersebut mempunyai sebuah makna semantik.
- Type bentukan seringkali disebut sebagai type komposisi, agregat.
- Notasi:

```
type namatype : < elemen1 : type1,
                  elemen2 : type2,
                  ...     >
```

Contoh: Tipe Bentukan

- Type Point
  - Type point menyatakan absis dan koordinat real pada sumbu kartesian

```
type Point : < X : real,      { absis }
               Y : real       { ordinat }
             >
```

- Type JAM
  - Type JAM menyatakan representasi "jam" dalam notasi HH:MM:SS dengan HH bernilai [0..23]; MM bernilai [0..59] dan SS bernilai [0..59

```
type Jam : <  HH  : integer [0..23],      { jam }
              MM  : integer [0..59], { menit }
              SS  : integer [0..59]  { detik }
           >
```

Notasi Type Bentukan

Notasi Algoritmik	Bahasa C
KAMUS **{ Definisi Type }** **type** namatype : < elemen1 : type1, elemen2 : type2, ... > **{ Deklarasi Variable }** nmvar1 : namatype nmvar2 : type1 {misal}	**/* KAMUS */** **/* Definisi Type */** typedef struct { type1 elemen1; type2 elemen2; ... } namatype; **/* Deklarasi Variabel */** namatype nmvar1; type1 nmvar2; /*misal*/
ALGORITMA **{ Akses Elemen }** nmvar2 ← nmvar1.elemen1 nmvar1.elemen2 ← <ekspresi>	**/* ALGORITMA */** **/* Akses Elemen */** nmvar2 = nmvar1.elemen1; nmvar1.elemen2 = <ekspresi>;

Contoh Notasi Type Bentukan

Notasi Algoritmik	Bahasa C
KAMUS **{ Definisi Type }** **type** Point : < X : integer, Y : integer > **{ Deklarasi Variable }** P : Point Bil : integer {misal}	**/* KAMUS */** **/* Definisi Type */** typedef struct { int X; int Y; } Point; **/* Deklarasi Variabel */** Point P; int Bil; /* misal */
ALGORITMA **{ Akses Elemen }** Bil ← P.X P.Y ← 20	**/* ALGORITMA */** **/* Akses Elemen */** Bil = P.X; P.Y = 20;

Operasi Inputan

- Dalam bahasa C terdapat fungsi yang digunakan untuk **menerima inputan** dari keyboard lalu **menampung dalam sebuah variable**, yang disebut dengan **fungsi scanf()**.
- Contoh:

Variable harus sesuai dengan tipe data

Tipe data

```
scanf ("%d", &a);
```

Variable (diawali tanda '&')

Referensi

- Inggriani Liem, IF-ITB, *Diktat Pemrograman Prosedural* (2007)
- FN/IF2030/Sem. 1 2009-2010, ITB
- I Made Joni, Budi Raharjo, *Pemrograman C dan Implementasinya* (2011)
- Abdul Kadir, *Mudah menjadi programmer C++* (2009)

THANKS