

PROSEDUR (Lanjutan)

Parameter

MINGGU KE: 3

TUJUAN:

- Mahasiswa dapat memahami penggunaan prosedur dengan parameter.
- Mahasiswa dapat membuat prosedur dengan parameter.

TEORI PENGANTAR:

Penggunaan parameter menawarkan mekanisme pertukaran informasi antara prosedur dan titik di mana ia dipanggil.

Cara pemanggilan prosedur dengan parameter adalah:

NamaProsedur(parameter aktual)

Ketika prosedur dipanggil, parameter aktual berkoresponden dengan parameter formal. Parameter aktual adalah parameter yang disertakan pada waktu pemanggilan. Parameter formal adalah parameter yang dideklarasikan pada bagian header prosedur.

Aturan yang harus diperhatikan:

- Jumlah parameter aktual pada pemanggilan prosedur harus sama dengan jumlah parameter formal pada deklarasi prosedurnya.
- Tiap parameter aktual harus bertipe sama dengan tipe parameter formal yang bersesuaian.
- Tiap parameter aktual harus diekspresikan dalam cara yang taat-asas dengan parameter formal yang bersesuaian, bergantung pada jenis parameter formal.

Ada tiga jenis parameter formal dalam prosedur:

1. Parameter masukan, yaitu parameter yang nilainya berlaku sebagai masukan untuk prosedur.
2. Parameter keluaran, yaitu parameter yang menampung keluaran yang dihasilkan oleh prosedur.
3. Parameter masukan/keluaran, yaitu parameter yang berfungsi sebagai masukan sekaligus keluaran bagi prosedur tersebut.

Parameter Masukan

Perhatikan Algoritma 3 pada Materi Minggu ke 2. Jika program utama ingin mengomunikasikan panjang alas dan tinggi segitiga ke prosedur `HitungLuasSegitiga`, maka peubah alas dan tinggi harus dideklarasikan sebagai parameter formal di bagian header prosedur:

procedure `HitungLuasSegitiga` (input alas, tinggi: real)

```
{ Menghitung luas segitiga dengan rumus Luas = (alas x tinggi)/2}
{K.Awal: alas dan tinggi sudah terdefinisi nilainya}
{K.Akhir: luas segitiga tercetak}
DEKLARASI
```

```
    luas : real
ALGORITMA
    luas ← (alas * tinggi)/2
    write (luas)
```

Algoritma 1

Program utama yang memanggil prosedur **HitungLuasSegitiga** harus mendeklarasikan prosedur ini dan memanggilnya dengan parameter aktual yang bersesuaian.

```
PROGRAM Segitiga
{Menghitung luas N buah segitiga}
DEKLARASI
    i, N: integer
    alas, tinggi: real
    procedure HitungLuasSegitiga(input alas, tinggi: real)
    {Menghitung luas segitiga dengan rumus L = (alas x tinggi)/2}
ALGORITMA:
    read(N)
    for i ← 1 to N do
        read(alas, tinggi)
        HitungLuasSegitiga(alas, tinggi)
    endfor
```

Algoritma 2

Karena yang dipentingkan adalah nilainya, maka nama parameter aktual tidak harus sama dengan nama parameter formal yang bersesuaian asalkan tipenya tetap sama. Pada algoritma berikut prosedur **HitungLuasSegitiga** dipanggil dengan menyertakan nilai alas dan tinggi di dalam peubah a dan t:

```
PROGRAM Segitiga
{Menghitung luas N buah segitiga}
DEKLARASI
    i, N: integer
    a, t: real
    procedure HitungLuasSegitiga(input alas, tinggi: real)
    {Menghitung luas segitiga dengan rumus L = (alas x tinggi)/2}
ALGORITMA:
    read(N)
    for i ← 1 to N do
        read(a, t)
        HitungLuasSegitiga(a, t)
    endfor
```

Algoritma 3

Karena yang dipentingkan adalah nilainya, maka parameter aktual boleh berupa ekspresi atau konstanta, contoh:

HitungLuasSegitiga(a*0.2, t*0.1)

atau

HitungLuasSegitiga (12,6)

Parameter Keluaran

Parameter keluaran dideklarasikan di dalam header prosedur dengan kata kunci output. Pada Algoritma 1 prosedur HitungLuasSegitiga, luas segitiga disimpan di dalam peubah luas dan nilainya dicetak di dalam prosedur tersebut. Jika luas dicetak di dalam program pemanggil, maka luas harus dinyatakan sebagai parameter keluaran.

```
procedure HitungLuasSegitiga(input alas, tinggi: real, output luas: real)
{menghitung luas segitiga dengan rumus Luas=(alas x tinggi)/2}
{K.Awal: alas dan tinggi sudah terdefinisi nilainya}
{K.Akhir: luas berisi luas segitiga}
DEKLARASI
    {tidak ada}
ALGORITMA:
    luas ← (alas * tinggi)/2
```

Algoritma 4

Keadaan akhir prosedur adalah kondisi di mana luas berisi luas segitiga. Program utama yang memanggil prosedur HitungLuasSegitiga harus mendeklarasikan prosedur ini dan memanggilnya dengan parameter aktual yang bersesuaian.

```
PROGRAM Segitiga
{Menghitung luas N buah segitiga}
DEKLARASI
    i, N: integer
    a, t, L: real      {alas, tinggi, dan luas segitiga}
    procedure HitungLuasSegitiga(input alas, tinggi: real, output luas: real)
    {Menghitung luas segitiga dengan rumus L = (alas x tinggi)/2}
ALGORITMA:
    read(N)
    for i ← 1 to N do
        read(alas, tinggi)
        HitungLuasSegitiga(a, t, L)
        write(L)
    endfor
```

Algoritma 5

Perhatikan korespondensi satu-satu antara parameter aktual dengan parameter formal yang berjenis keluaran.

Parameter aktual harus berupa peubah, tidak boleh berupa konstanta atau ekspresi.

Parameter Masukan/Keluaran

Parameter masukan/keluaran dideklarasikan di dalam header prosedur dengan kata kunci input/output. Jenis parameter masukan/keluaran adalah parameter aktual dan harus berupa peubah, tidak boleh berupa konstanta atau ekspresi.

Berikut ini adalah prosedur Inc yang berkoresponden dengan $x=x+1$.

```
procedure Inc(input/output x: integer)
{Menaikan nilai x sebesar 1}
{K.Awal: x sudah terdefinisi nilainya}
{K.Akhir: nilai x bertambah 1}
DEKLARASI
ALGORITMA:
     $x \leftarrow x+1$ 
```

Program pemanggilnya adalah

```
PROGRAM Cetak0Sampai10
{Mencetak nilai dari 0 sampai 10}
DEKLARASI
    x: integer
    procedure Inc(input/output x: integer)
    {Menaikkan nilai x sebesar 1}
ALGORITMA:
     $x \leftarrow 0$ 
    repeat
        write(x)
        Inc(x)
    until  $x>10$ 
```

Analisa efek penggunaan parameter masukan dan perbandingan dengan parameter masukan/keluaran berikut:

Prosedur dengan parameter masukan:

```
procedure TambahEmpat(input x, y: integer)
{Menambahkan nilai x dan y masing-masing dengan 4}
{K.Awal: x dan y sudah berisi nilai}
{K.Akhir: nilai x dan y masing-masing bertambah 4, lalu dicetak}
DEKLARASI
    {tidak ada}
ALGORITMA:
     $x \leftarrow x + 4$ 
     $y \leftarrow y + 4$ 
    output ('nilai x dan y di akhir prosedur Empat')
    write(' x = ', x)
```

write(' y = ', y)

Program utama:

PROGRAM XYZ

{Program yang memperlihatkan efek penggunaan parameter masukan}

DEKLARASI

a, b: integer

procedure TambahEmpat(input x, y: integer)

{Menambahkan nilai x dan y masing-masing dengan 4}

ALGORITMA:

a ← 15

b ← 10

write('nilai a dan b sebelum pemanggilan prosedur TambahEmpat:')

write(' a = ', a)

write(' b = ', b)

TambahEmpat(a,b)

write('nilai a dan b sesudah pemanggilan prosedur TambahEmpat:')

write(' a = ', a)

write(' b = ', b)

Prosedur dengan parameter masukan/keluaran

procedure TambahEmpat(input/output x, y: integer)

{Menambahkan nilai x dan y masing-masing dengan 4}

{K.Awal: x dan y sudah berisi nilai}

{K.Akhir: nilai x dan y masing-masing bertambah 4, lalu dicetak}

DEKLARASI

{tidak ada}

ALGORITMA:

x ← x + 4

y ← y + 4

write('nilai x dan y di akhir prosedur Empat')

write(' x = ', x)

write(' y = ', y)

Program utama:

PROGRAM XYZ

{Program yang memperlihatkan efek penggunaan parameter masukan/keluaran}

DEKLARASI

a, b: integer

procedure TambahEmpat(input x, y: integer)

{Menambahkan nilai x dan y masing-masing dengan 4}

ALGORITMA:

a ← 15

b ← 10

write('nilai a dan b sebelum pemanggilan prosedur TambahEmpat:')

write(' a = ', a)

write(' b = ', b)

```
TambahEmpat(a,b)
write('nilai a dan b sesudah pemanggilan prosedur TambahEmpat:')
write('  a = ', a)
write('  b = ', b)
```

KESIMPULAN:

- Sangat dianjurkan menulis program yang modular. Program yang dipecah menjadi beberapa prosedur (atau fungsi) menunjukkan teknik pemrograman yang baik dan terstruktur.
- Sangat dianjurkan menulis prosedur dengan parameter. Parameter berfungsi sebagai media komunikasi antara modul dengan program pemanggil. Parameter dapat mengurangi kebutuhan penggunaan peubah global.
- Bila prosedur menghasilkan keluaran yang digunakan oleh program pemanggil, gunakan parameter keluaran untuk menampung keluaran tersebut. Bila prosedur tidak menghasilkan keluaran, atau menghasilkan keluaran dan ternyata keluaran tersebut hanya digunakan di dalam prosedur itu saja, gunakan parameter masukan. Bila prosedur menerima masukan sekaligus keluaran pada parameter yang sama, gunakan parameter masukan/keluaran.

PRAKTIKUM:

1. Buat program delphi dengan prosedur yang menerima nomor menu program dan menuliskan pesan nomor menu yang dipilih.

```
procedure AktifkanMenu(input NomorMenu: integer)
{Mencetak nomor menu yang dipilih}
{K.Awal: NomorMenu sudah berisi nomor menu yang dipilih}
{K.akhir: Pesan yang bersesuaian dengan NomorMenu tercetak.}
DEKLARASI
    {tidak ada}
```

ALGORITMA:

```
    case NomorMenu
        1: write('Anda memilih menu nomor 1')
        2: write('Anda memilih menu nomor 2')
        3: write('Anda memilih menu nomor 3')
        4: write('Anda memilih menu nomor 4')
        5: write('Anda memilih menu nomor 5')
    endcase
```

Pernyataan `write('Anda memilih menu nomor ...')` dapat diganti dengan pemanggilan prosedur/fungsi yang berkaitan dengan menu yang dipilih.

2. Buat program dalam Pascal yang berisi prosedur untuk menghitung rata-rata N buah bilangan bulat. Data bilangan bulat dibaca dari piranti masukan. Masukan prosedur adalah N dan keluarannya adalah nilai rata-rata.

Procedure HitungRataRata(input N: integer, output n:real)
{Menghitung rata-rata N buah bilangan bulat yang dibaca dari piranti masukan.}
{K.Awal: N sudah berisi banyaknya bilangan bulat, $N > 0$.}
{K.Akhir: u berisi rata-rata seluruh bilangan}

DEKLARASI

x: integer {data bilangan bulat yang dibaca dari papan kunci}
k: integer {pencacah banyak bilangan}
jumlah: integer {jumlah seluruh bilangan}

ALGORITMA:

jumlah $\leftarrow$ 0 {inisialisasi}
for k $\leftarrow$ 1 to N do
 read(x)
 jumlah $\leftarrow$ jumlah + x
endwhile
u $\leftarrow$ jumlah/N

3. Buat program dengan menggunakan prosedur dengan parameter untuk menambah suatu jam dengan satu detik.

TUGAS:

Tuliskan prosedur yang menerima sebuah tanggal dalam bentuk dd-mm-yyyy dan memberikan keluaran dd NAMA_BULAN yyyy. Parameter tanggal berjenis masukan/keluaran.