

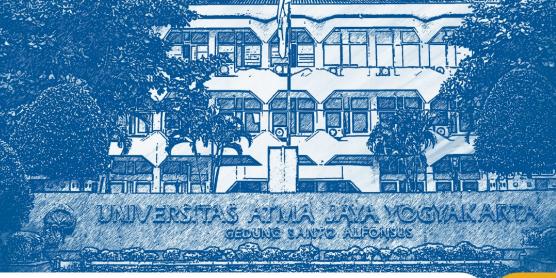


Dasar Pemrograman (INFT06204)

Minggu 10

Prosedur 2







• Kebanyakan program memerlukan pertukaran informasi antara prosedur (atau fungsi) dan titik di mana ia dipanggil.

→ Parameter berfungsi sebagai media komunikasi antara modul dengan program pemanggil.

- Tiap item data ditransferkan antara parameter aktual dan parameter formal.
 - Parameter aktual: parameter yang disertakan pada waktu pemanggilan.
 - Parameter formal: parameter yang dideklarasikan di bagian header prosedur itu sendiri.



- Saat prosedur dipanggil, parameter aktual menggantikan parameter formal.
- Tiap parameter aktual berpasangan dengan parameter formal yang bersesuaian.
- Aturan penting 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 sesuai dengan parameter formal yang bersesuaian, bergantung pada jenis parameter formal.



```
#include <stdio.h>
void HitungLuasLingkaran(float *hasil, float jejari);
void main()
     float Luas, radius;
     radius=9.8;
     HitungLuasLingkaran(&Luas, radius);
     printf("%f\n",Luas);
void HitungLuasLingkaran(float *hasil, float jejari)
     *hasil = 3.14*jejari*jejari;
```



- Jenis parameter formal yang disertakan dalam prosedur:
 - Parameter masukan (input parameter)
 - → nilainya berlaku sebagai masukan untuk prosedur.
 - Parameter keluaran (output parameter)
 - → menampung nilai keluaran yang dihasilkan prosedur.
 - Parameter masukan/keluaran (input/output parameter)
 - → berfungsi sebagai masukan sekaligus keluaran dari prosedur.



Pada bahasa pemrograman:

- Parameter masukan
 - → value parameter, parameter by value
- Parameter keluaran & parameter masukan/keluaran
 - → reference parameter, parameter by reference





Parameter Masukan

- Nilai parameter aktual diisikan (assign) ke dalam parameter formal yang bersesuaian untuk digunakan dalam badan prosedur yang bersangkutan.
- Nilai yang dinyatakan oleh parameter masukan tidak dapat dikirim ke arah sebaliknya.
- Perubahan nilai parameter di dalam prosedur <u>tidak mengubah</u> nilai parameter aktual.
- Nama parameter aktual boleh berbeda dengan nama parameter formal.



Parameter Masukan/Keluaran

- Parameter By Reference



Contoh Parameter Input/Output

```
#include <stdio.h>
void Tukar(int *A, int *B)
void main()
  int nilai1, nilai2;
  scanf("%d", &nilai1);
   scanf("%d", &nilai2);
   printf("Nilai 1 sebelum pertukaran = %d",nilai1);
   printf("Nilai 2 sebelum pertukaran = %d",nilai2);
   Tukar(&nilai1,&nilai2);
   printf("Nilai 1 sebelum pertukaran = %d",nilai1);
   printf("Nilai 2 sebelum pertukaran = %d",nilai2);
```

```
void Tukar(int *A, int *B)
{
    int temp;

    temp = *A;
    *A = *B;
    *B = temp;
}
```



Contoh Parameter Input/Output

```
#include <stdio.h>
                                                  Deklarasi prosedur
void Hit_Rata2(int M, float *rata);
                                                  (prototype)
void main()
                          Variabel lokal
   int N;
   float rata2;
   printf("Banyak data: %d"); scanf("%d", &N);
   Hit_Rata2(N, &rata2);
                                          Pemanggilan prosedur
   printf("Rata-rata: %f\n", rata2);
```



Contoh Parameter Input/Output (2)

```
Parameter input
                                    Parameter output
void Hit_Rata2(int M) (loat *rata)
  int i, bil, jumlah;
  jumlah=0;
  for(i=1;i<=M;i++)
           printf("Masukkan angka ke-%d: ", i); scanf("%d", &bil);
           jumlah=jumlah+bil;
   *rata=(float)jumlah/M;
```



Passing Parameter

```
#include <stdio.h>
void Hit_Rata2(int M, float *rata);
void main()
    int N;
    float rata2;
  printf("Banyak data: %d"); scanf("%d", &N);
  Hit_Rata2(N, &rata2);
  printf("Rata-rata : %f\n", rata2);
```

```
void Hit_Rata2(int M, float *rata)
   int i, bil, jumlah;
   jumlah=0;
   for(i=0;i\leq=M;i++)
          printf("Masukkan angka ke-%d:", i);
          scanf("%d", &bil);
          jumlah=jumlah+bil;
   *rata=(float)jumlah/M;
```



Latihan (1)

• Buat prosedur untuk mempertukarkan 2 buah bilangan integer menggunakan prosedur semi naïve input.





Latihan (2)

- Buat program untuk menghitung biaya yang harus dibayar oleh pengguna warnet dengan ketentuan:
 - Tanpa modul prosedur
 - Operator memasukkan waktu mulai dan selesai penggunaan (diumpamakan jam selesai lebih besar dari jam mulai → tidak perlu dibuatkan kondisi)
 - Satuan waktu penggunaan terkecil adalah 30 menit
 - Biaya penggunaan per 30 menit adalah Rp 1500.



Latihan (3)

- Kembangkan program sebelumnya dengan membuat prosedur untuk:
 - Mengubah hitung selisih waktu
 - Menghitung biaya pemakaian





Question and Answer



Terima kasih

