



UNIVERSITAS
ATMA JAYA YOGYAKARTA
serviens in lumine veritatis



Dasar Pemrograman (INFT06204)

Minggu 8
Prosedur





Pemrograman Modular

Teknik pemrograman modular :

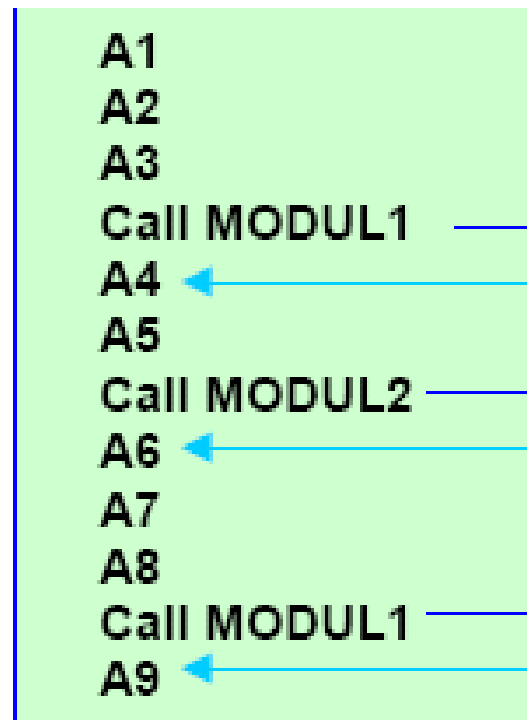
- Program dipecah menjadi beberapa **subprogram** yang lebih kecil.
- Subprogram (**modul**, ***routine***) kadang independen dari program utama sehingga dapat dirancang tanpa mempertimbangkan konteks tempat ia digunakan, bahkan dapat dirancang orang lain.

Modularisasi memberikan dua keuntungan :

- Untuk aktivitas yang harus dilakukan lebih dari satu kali, modularisasi menghindari penulisan teks program yang sama secara berulang kali.
- Kemudahan dalam menulis dan menemukan kesalahan (*debug*) program.



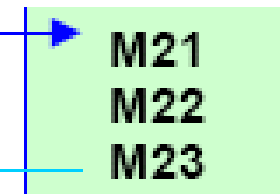
Program Utama



MODUL1



MODUL2





Definisi Prosedur

- **Prosedur** (*subroutine*) adalah modul program yang mengerjakan tugas/aktifitas yang spesifik dan menghasilkan suatu efek netto.
- **Efek netto** diketahui dengan membandingkan keadaan awal (sebelum) dan keadaan akhir (sesudah) pelaksanaan sebuah prosedur.

→ Pada setiap prosedur harus didefinisikan keadaan awal (K.Awal) dan keadaan akhir (K.Akhir).



Definisi Prosedur

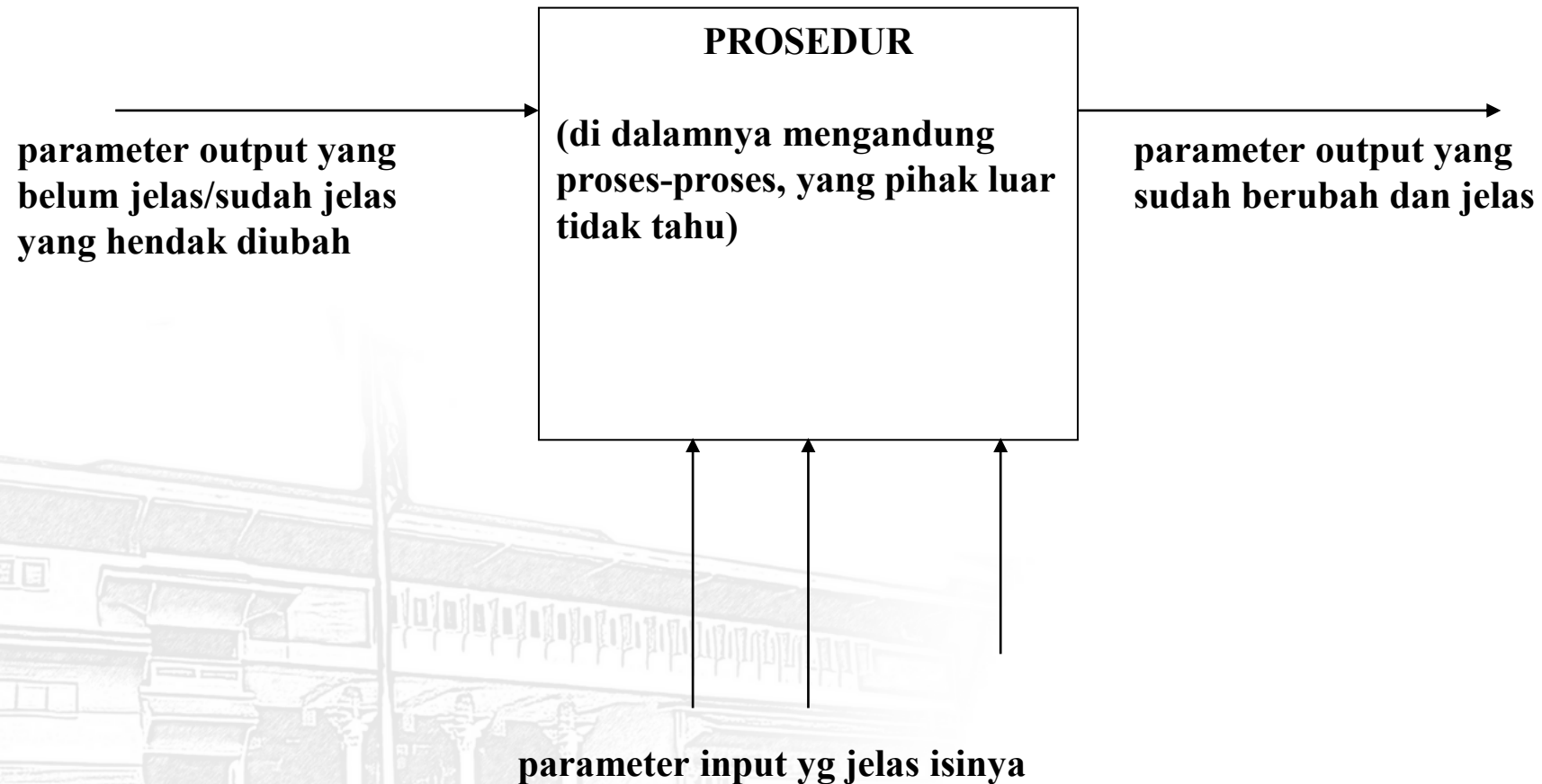
- Pada dasarnya *prosedur membutuhkan input dan output*.
- Input yang standar berasal dari piranti input standar (*standard input device*), yaitu **keyboard**.
- Output yang standar berasal dari piranti output standar (*standard output device*), yaitu **monitor**.
- Kedua piranti I/O standar ini dijembatani penggunaannya dalam C oleh library **stdio.h** (standard I/O), yang memperkenalkan **printf** untuk keperluan output dan **scanf** untuk keperluan input.



- Berdasarkan darimana datangnya input dan output, prosedur dikategorikan ke dalam 4 macam:

input	output	prosedur
Std dev	Std dev	Naïve
Parameter	Std dev	Semi Naïve
Std dev	Parameter	Semi Naïve
Parameter	Parameter	Nett Effect

- Hanya ada satu jenis prosedur yang direkomendasikan, yaitu prosedur yang menghasilkan efek netto (nett effect procedure)





Mendefinisikan Prosedur

- Struktur prosedur sama dengan struktur algoritma, yaitu : bagian judul, bagian deklarasi, dan badan prosedur.
- Setiap prosedur memiliki nama unik (sebaiknya diawali dengan kata kerja, mis : TUKAR, HITUNG_LUAS, CARI_MAKS, dll)





Mendefinisikan Prosedur (2)

Prosedur Naif:

```
void Hitung_Luas
```

```
{
```

```
    /*prosedur untuk menghitung luas segitiga dengan input t dan a */
```

```
}
```

Prosedur Berparameter:

```
void Hitung_Luas(int *Luas, int tinggi, int alas)
```

```
{
```

```
    /*prosedur untuk menghitung luas segitiga dengan input t dan a */
```

```
}
```



Pemanggilan Prosedur

- Prosedur diakses dengan memanggil namanya dari program pemanggil (program utama atau modul program lain).
- Dalam program pemanggil, prototype prosedur harus dideklarasikan supaya program pemanggil mengenal nama prosedur tersebut serta cara mengaksesnya.





Nama Lokal dan Nama Global

Nama Lokal :

- Nama-nama di bagian deklarasi prosedur.
- Bersifat lokal, hanya dapat digunakan di dalam prosedur yang melingkupinya.

Nama Global :

- Nama-nama yang dideklarasikan di program utama.
- Bersifat global, dapat digunakan di bagian manapun dalam program, baik di program utama maupun di prosedur.



Nama Lokal dan Nama Global

Menggunakan Nama Lokal atau Nama Global ?

- Jika nama (peubah, tipe, atau tetapan) digunakan di seluruh bagian program, maka harus dideklarasikan global.
- Jika nama tersebut hanya digunakan di dalam prosedur, maka sebaiknya dideklarasikan lokal.
- Usahakan menggunakan nama global sedikit mungkin!
→ menyulitkan dalam pencarian kesalahan (*debugging*)



Contoh

```
#include <stdio.h>
```

```
int N;
```

```
float rata2;
```

Variabel global

```
void Hit_Rata2();
```

Deklarasi prosedur
(prototype)

```
void main()
```

```
{
```

```
printf("Banyak data: %d"); scanf("%d", &N);
```

```
Hit_Rata2();
```

Pemanggilan prosedur

```
printf("Rata-rata : %f\n", rata2);
```

```
}
```




Contoh

```
void Hit_Rata2()
{
    int i, bil, jumlah;
    jumlah=0;
    for(i=0;i<=N;i++)
    {
        printf("Masukkan angka ke-%d: ", i); scanf("%d", &bil);
        jumlah=jumlah+bil;
    }
    rata2=(float)jumlah/N;
}
```

Variabel lokal

Variabel Global



**UNIVERSITAS
ATMA JAYA YOGYAKARTA**
serviens in lumine veritatis

Question and Answer



Terima kasih



uajy



Universitas Atma Jaya Yogyakarta



www.uajy.ac.id