

PROCEDURE DAN FUNCTION

- Seolah-olah terpisah dari program utama, namun sebenarnya mrpkn **bagian dari program utama** yg kemudian dipanggil (dapat berulang) oleh program utama tsb.
- Modul atau blok program yang digunakan untuk membantu program utama dalam **menyelesaikan submasalah-submasalahnya**.
- Di dalam suatu Prosedur atau Fungsi dimungkinkan adanya prosedur atau fungsi lain (**nested**).
- Prosedur dan fungsi ada dua yaitu **pustaka** (disiapkan oleh bahasa pemrograman) dan **dibuat sendiri**.
- Nama prosedur atau fungsi **dinyatakan dua kali**, pertama pada **bagian deklarasi** atau kamus dan kedua di **bagian tubuh program utama** (untuk memanggilnya).
- Semua kamus/deklarasi dalam prosedur disebut sebagai **kamus/deklarasi lokal**.
- Procedure pada Algoritma
 - Prosedur **Tanpa Parameter**
 - Prosedur **Dengan Parameter Masukan**
 - Prosedur **Dengan Parameter Keluaran**
 - Prosedur **Dengan Parameter Masukan/Keluaran**

Prosedur Tanpa Parameter pada Algoritma

- *Cara memanggil (dari algoritma/program utama)*

```
KAMUS
    Procedure namaprosedur
ALGORITMA
    namaprosedur
```

- *Cara mendefinisikan:*

```
Procedure namaprosedur
{ kondisi awal : ... }
{ kondisi akhir : ... }

KAMUS LOKAL
    

ALGORITMA
    
```

Prosedur Dengan Parameter Masukan Pada Algoritma

- *Cara memanggil:*

KAMUS

Procedure namaprosedur(input namaparameter : type)

ALGORITMA

namaprosedur(namaparameter)

- *Cara mendefinisikan:*

Procedure namaprosedur(input namaparameter : type)

{ *kondisi awal* : ... }

{ *kondisi akhir* : ... }

KAMUS LOKAL

ALGORITMA

Prosedur Dengan Parameter Keluaran Pada Algoritma

- *Cara memanggil:*

KAMUS

Procedure namaprosedur(output namaparameter : type)

ALGORITMA

namaprosedur(namaparameter)

- *Cara mendefinisikan:*

Procedure namaprosedur(output namaparameter : type)

{ *kondisi awal* : ... }

{ *kondisi akhir* : ... }

KAMUS LOKAL

ALGORITMA

Prosedur Dengan Parameter Masukan/Keluaran Pada Algoritma

- *Cara memanggil:*

KAMUS

Procedure namaprosedur(input/output namaparameter:type)

ALGORITMA

namaprosedur(namaparameter)

- *Cara mendefinisikan:*

Procedure namaprosedur(input/output namaparameter : type)

{ *kondisi awal* : ... }

{ *kondisi akhir* : ... }

KAMUS LOKAL

ALGORITMA

Fungsi Pada Algoritma

- *Cara memanggil:*

KAMUS

Function namafungsi(Input daftarnamaparameter) → typehasil

ALGORITMA

namavar ← namafungsi(daftarnamaparameteraktual)

atau

output(namafungsi(daftarnamaparameteraktual))

- *Cara mendefinisikan:*

Function namafungsi(Input daftarnamaparameter) →
typehasil

{ *kondisi awal* : ... }

{ *kondisi akhir* : ... }

KAMUS LOKAL

ALGORITMA

...
return hasil

Fungsi Pada Bahasa C++

- Pada bahasa C++, tidak ada blok atau bagian atau modul program yang disebut prosedur, semua disebut fungsi.
- Fungsi yang tidak memiliki nilai yang dikembalikan ke program utama disebut void.

Cara penulisan deklarasi fungsi pada C++ yaitu:

```
tipehasil namafungsi();
```

atau

```
tipehasil namafungsi(daftar_tipe_parameter_formal);
```

```
tipehasil namafungsi(daftar_tipedannama_parameter_formal);
```

Cara pemanggilan fungsi pada C++ yaitu:

(pemanggilan fungsi pada C++ berada pada main() {...})

- Cara tak langsung (untuk fungsi yang mengembalikan nilai)

```
type varhasil;
```

```
deklarasi parameter aktual;
```

```
varhasil=namafungsi(daftar_parameter_aktual);
```

```
cout << "..." << varhasil;
```

- Cara langsung (untuk fungsi bertipe void/ tidak mengembalikan nilai)

```
deklarasi parameter aktual;
```

```
namafungsi(daftar_parameter_aktual);
```

```
cout << "..." << namafungsi(daftar_parameteraktual);
```

Cara pendefinisian fungsi pada C++:

```
tipe_hasil namafungsi()  
{  
    deklarasi variabel lokal  
    daftar_pernyataan;  
}
```

- dilakukan bila tidak ada parameter yang dilewatkan fungsi ini
- input data ada pada program yang memanggil

atau

```
tipe_hasil namafungsi(daftar_nama_parameter_formal);  
deklarasi parameter formal  
{  
    deklarasi variabel lokal  
    daftar_pernyataan;  
    return(...);  
}
```

- dilakukan bila ada parameter yang dilewatkan fungsi ini

atau

```
tipe_hasil namafungsi(daftar_tipe_parameter_formal)  
{  
  deklarasi variabel lokal  
  daftar_pernyataan;  
  return(...);  
}
```

atau

```
tipe_hasil namafungsi(daftar_tipedannama_parameter_formal)  
{  
  deklarasi variabel lokal  
  daftar pernyataan;  
  return(...);  
}
```

Contoh-contoh :

Algoritma

Prosedur Tanpa Parameter

Algoritma LuasSegi4

{untuk menghitung luas segiempat}

KAMUS

Procedure HitungLuas

ALGORITMA

HitungLuas

Procedure HitungLuas

KAMUS LOKAL

Luas, panjang, lebar : real

ALGORITMA

Input(panjang)

Input(lebar)

Luas \leftarrow panjang * lebar

Output(Luas)

Prosedur Dengan Parameter Masukan

Algoritma LuasSegi4

{untuk menghitung luas segiempat}

KAMUS

panjang, lebar : real

Procedure HitungLuas(input pj,lb : real)

ALGORITMA

Input(panjang)

Input(lebar)

HitungLuas(panjang,lebar)

Procedure HitungLuas(input pj,lb : real)

KAMUS LOKAL

Luas : real

ALGORITMA

Luas \leftarrow pj * lb

Output(Luas)

Prosedur Dengan Parameter Keluaran

Algoritma LuasSegi4

{untuk menghitung luas segiempat}

KAMUS

Luas : real

Procedure HitungLuas(output L : real)

ALGORITMA

HitungLuas(Luas)

Output(Luas)

Procedure HitungLuas(output L : real)

KAMUS LOKAL

panjang, lebar : real

ALGORITMA

Input(panjang)

Input(lebar)

L \leftarrow panjang * lebar

Prosedur Dengan Parameter Masukan/Keluaran

Algoritma Tukar

{untuk menukar 2 bilangan bulat}

KAMUS

x, y : integer

Procedure Tukar(input/output x,y: integer)

ALGORITMA

Input(x,y)

Output(x,y)

Tukar(x,y)

Output(x,y)

Procedure Tukar(input/output x,y: integer)

KAMUS LOKAL

temp : integer

ALGORITMA

temp \leftarrow x

x \leftarrow y

y \leftarrow temp

Procedure Pada Bahasa C++ (fungsi tidak mengembalikan nilai)

Tanpa Parameter

```
#include <iostream.h>
void HitungLuas();
main()
{
    HitungLuas();
}

void HitungLuas()
{
    float Luas, panjang, lebar;

    cout << "panjang = "; cin >> panjang;
    cout << "lebar = "; cin >> lebar;
    Luas = panjang * lebar;
    cout << "Luas = " << Luas;
}
```

Dengan Parameter Masukan

```
#include <iostream.h>
void HitungLuas(float pj, float lb);
main()
{
    float panjang, lebar;

    cout << "panjang = "; cin >> panjang;
    cout << "lebar = "; cin >> lebar;
    HitungLuas(panjang, lebar);
}

void HitungLuas(float pj, float lb)
{
    float Luas;

    Luas = pj * lb;
    cout << "Luas = " << Luas;
}
```

Dengan Parameter Keluaran

```
#include <iostream.h>
void HitungLuas(float *Luas);
main()
{
    float Luas;

    HitungLuas(&Luas);
    cout << "Luas = " << Luas;
}

void HitungLuas(float *Luas)
{
    float panjang, lebar;

    cout << "panjang = "; cin >> panjang;
    cout << "lebar = "; cin >> lebar;
    *Luas = panjang * lebar;
}
```

Dengan Parameter Masukan/Keluaran

```
#include <iostream.h>
void Tukar (int *x, int *y);
main()
{
    int x,y;

    cout << "x = "; cin >> x;
    cout << "y = "; cin >> y;
    Tukar(&x,&y);
    cout << "x = " << x << endl;
    cout << "y = " << y;
}

void Tukar (int *x, int *y)
{
    int temp;

    temp = *x; *x = *y; *y = temp;
}
```

Fungsi Pada Algoritma

Algoritma LuasSegi4

{untuk menghitung luas segiempat}

KAMUS

p, lb : real

Function Luas(input panjang,lebar: real) → real

ALGORITMA

Input(p)

Input(lb)

Output(Luas(p,lb))

Function Luas(input panjang,lebar: real) → real

KAMUS LOKAL

L : real

ALGORITMA

L ← panjang * lebar

Return L

Fungsi Pada Bahasa C

(fungsi mengembalikan nilai)

```
#include <iostream.h>
float Luas(float panjang, float lebar);
main()
{
    float p, lb;

    cout << "panjang = "; cin >> p;
    cout << "lebar = "; cin >> lb;
    cout << "Luas segiempat = " << Luas(p,lb);
}

float Luas(float panjang, float lebar)
{
    float L;

    L = panjang * lebar;
    return (L);
}
```