

Subprogram

Bahan Kuliah SD3105 Algoritma dan Pemrograman

Sevi Nurafni

Fakultas Sains dan Teknologi

Universitas Koperasi Indonesia 2024

Kode yang Berulang

- Semakin besar program, semakin banyak bagian kode yang berulang
- Sangat tidak efisien jika bagian kode yang sama/serupa diketik berulang-ulang, (bahkan kalau di-copy-paste)
- Di samping itu, dalam banyak persoalan, ada berbagai rumus/formula yang berulang-ulang dipakai dalam satu program
- Bagaimana jika ada cara supaya bagian kode tersebut tidak perlu diketik berulang-ulang, tapi tetap dapat digunakan berkali-kali dalam program yang sama

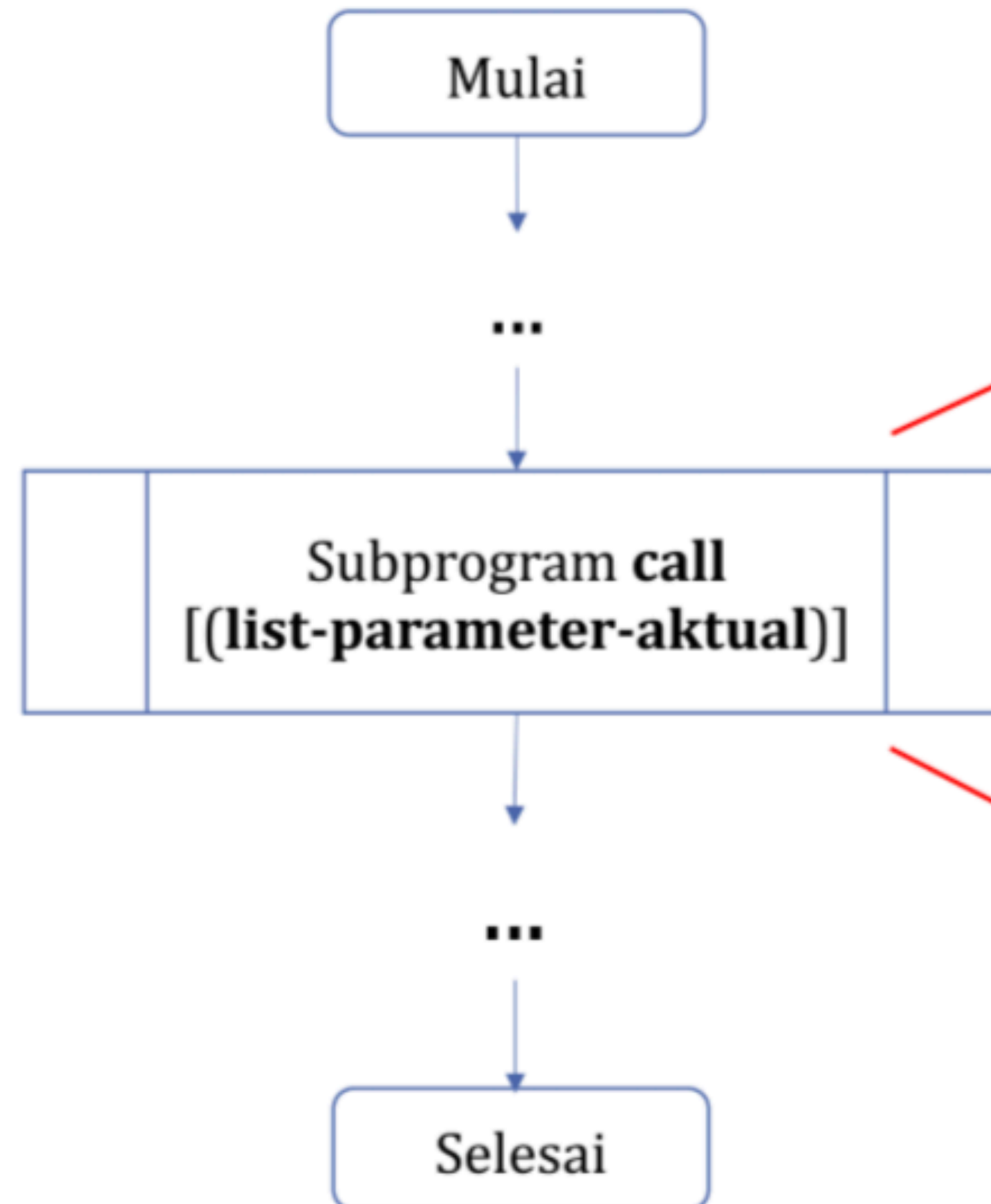
Subprogram



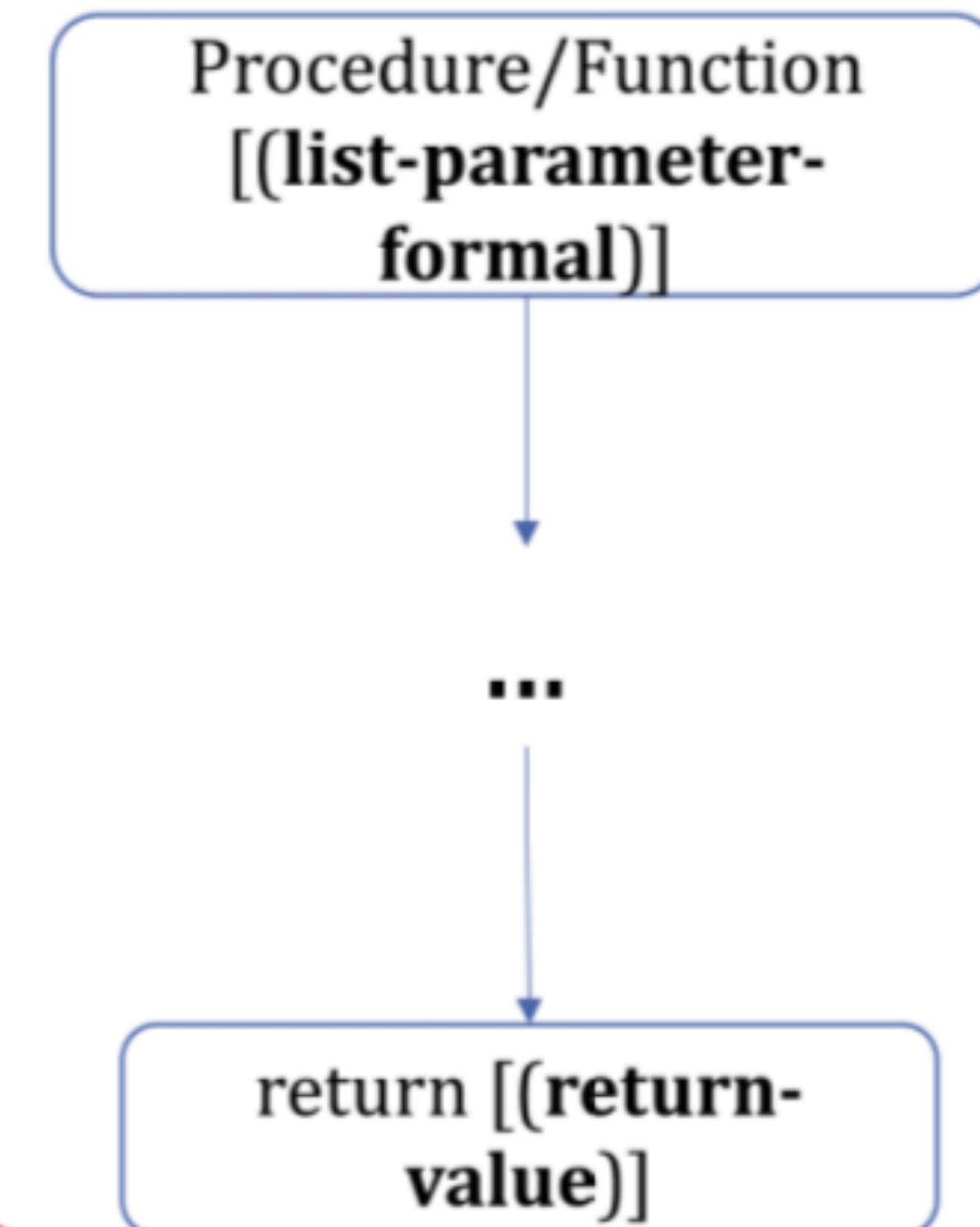
- A set of instructions designed to perform a frequently used operation within a program
- 2 (dua) jenis subprogram:
 - Fungsi: pemetaan suatu nilai domain (input) ke range (output)
 - Hasil dari fungsi dinyatakan dalam sebuah type data yang eksplisit
 - Prosedur: deretan instruksi yang jelas initial state dan final state-nya → mirip seperti program secara umum, namun dalam scope yang lebih kecil

Flowchart Symbol

Pemanggilan subprogram dalam
algoritma program utama



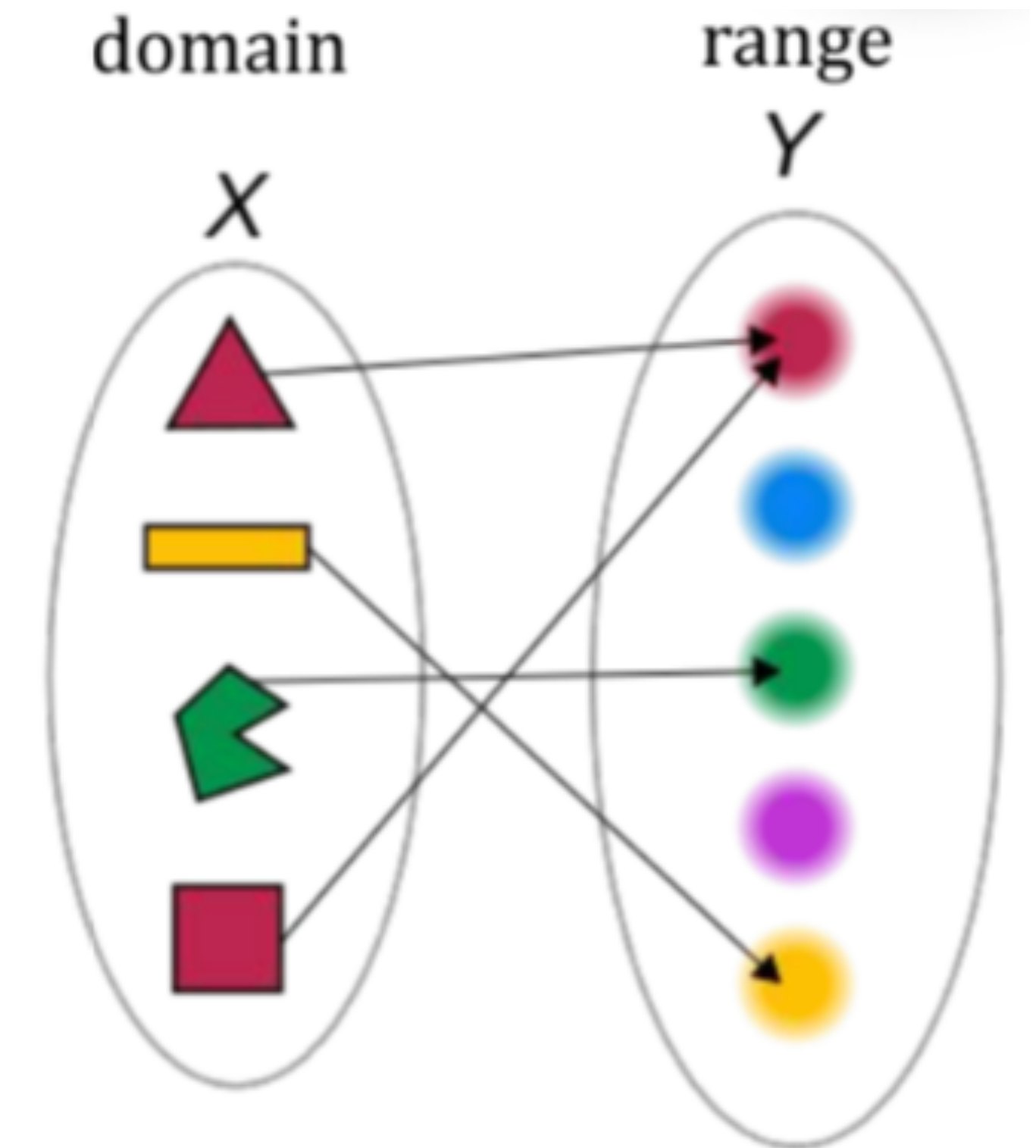
Flowchart terpisah untuk
pendefinisian dan **realisasi**
subprogram



Fungsi

Subprogram

- Konsep fungsi di pemrograman didasari oleh konsep pemetaan dan fungsi di matematika
- Fungsi: asosiasi (pemetaan) antara 2 himpunan nilai yaitu domain dan range
 - Setiap elemen pada himpunan domain dipetakan tepat satu ke sebuah elemen pada himpunan range
- Contoh: $f(X) = X^2$
 - fungsi untuk menghitung kuadrat dari suatu bilangan
 - Domain: bilangan bulat
 - Range: bilangan bulat (0 atau positif)



$$f: X \rightarrow Y$$

Fungsi dalam Pemrograman



- Memrogram fungsi pada dasarnya adalah: merakit isi black box
 - Berangkat dari keadaan awal \rightarrow himpunan nilai yang terdefinisi sebagai input (domain)
 - Menghasilkan nilai-nilai yang mendefinisikan keadaan akhir \rightarrow himpunan nilai yang terdefinisi sebagai output (range)
 - Tugas pemrogram fungsi adalah menentukan langkah-langkah untuk menghasilkan keadaan akhir berdasarkan keadaan awal
- Fungsi didefinisikan sebagai bagian terpisah dari program dan dipanggil dalam program utama

Kegunaan Fungsi



- Program dapat didekomposisi menjadi sub-sub bagian
 - Tiap sub bagian dapat didefinisikan sebagai fungsi yang tinggal dipanggil sebagai 1 baris atau ekspresi dalam program utama
- Code reuse instead of code rewriting
 - Jika task yang harus dikerjakan fungsi banyak dipakai di program, memprogram menjadi jauh lebih sederhana jika task tersebut dibuat dalam bentuk fungsi
 - Contoh: fungsi untuk menghitung akar kuadrat (sqrt) sangat berguna untuk berbagai jenis persoalan → bayangkan kalau setiap kali Anda harus menulis programnya
- Setiap fungsi dapat dites secara mandiri dan tidak tergantung pada bagian program yang lain
 - Dalam C++, fungsi dapat dites dengan membuat program sederhana yang hanya memanggil fungsi tersebut.
 - Jika program besar dan harus dikerjakan oleh lebih dari 1 programmer, hal ini memudahkan pembagian kerja

Mendefinisikan Fungsi dalam C++



- Nama fungsi didefinisikan sebelum kurung buka () dengan tipe data output ditentukan di awal.
- Spesifikasi fungsi dapat dituliskan sebagai komentar di atas definisi fungsi.
- Tipe data input harus dideklarasikan secara eksplisit sebelum nama parameter di dalam tanda kurung.
- Jika terdapat lebih dari satu parameter, setiap parameter dipisahkan dengan tanda koma , .
- Tipe data output didefinisikan secara eksplisit di awal definisi fungsi, sesuai dengan nilai yang dikembalikan melalui perintah `return`.

```
TipeData NamaFungsi (TipeDataParameter Parameter)
    //spesifikasi fungsi

    return NilaiOutput;
}

int Kuadrat(int X) {
    int hasil = X * X;
    return hasil;
}
```

Fungsi dalam Pemrograman

```
#include <iostream>
using namespace std;

// Fungsi Kuadrat
int Kuadrat(int X) {
    // menghitung kuadrat X
    int hasil = X * X;
    return hasil;
}

// ALGORITMA PROGRAM UTAMA
int main() {
    int A = 5;
    int B = Kuadrat(A) + 10;

    cout << "Nilai B: " << B <<
endl;
    return 0;
}
```

Function flow of control:

1) Salah satu baris pada kode program utama memanggil

fungsi: `int B = Kuadrat(A) + 10 # A = 5`

2) Program beralih ke kode fungsi Kuadrat mulai dari baris yang pertama sampai pada baris yang mendefinisikan hasil fungsi (`return`). Parameter input diasosiasikan dengan daftar parameter input pada fungsi.

Fungsi dalam Pemrograman

```
#include <iostream>
using namespace std;

// Fungsi Kuadrat
int Kuadrat(int X) {
    // menghitung kuadrat X
    int hasil = X * X;
    return hasil;
}

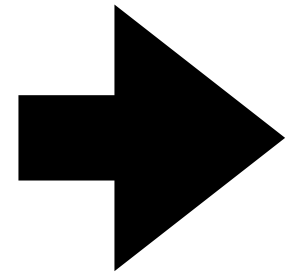
// ALGORITMA PROGRAM UTAMA
int main() {
    int A = 5;
    int B = Kuadrat(A) + 10;

    cout << "Nilai B: " << B <<
endl;
    return 0;
}
```

Function flow of control:

```
int Kuadrat(int X) {
    int hasil = X * X;
    return hasil;
}

int Kuadrat(int 5) {
    int hasil = 5 * 5;
    return hasil; //hasil=25
}
```



3) Program meninggalkan fungsi dengan menyimpan hasil perhitungan dan kembali pada baris terakhir program utama yang ditinggalkannya dan menggantikan hasil perhitungan berdasarkan hasil

fungsi: $B = 25 + 10 \# B = 35$

4) Program melanjutkan ke instruksi berikutnya.

Contoh1

- Buatlah program yang menerima masukan buah nilai jari-jari lingkaran (bilangan riil), misalnya r , dan menuliskan luas lingkaran ke layar
- Perhitungan luas lingkaran → dibuat menjadi fungsi

```
#include <iostream>
using namespace std;

// Fungsi untuk menghitung luas lingkaran
float Luas(float r) {
    float Luas = 3.14 * r * r;
    return Luas;
}

// PROGRAM UTAMA
int main() {
    float r, L;

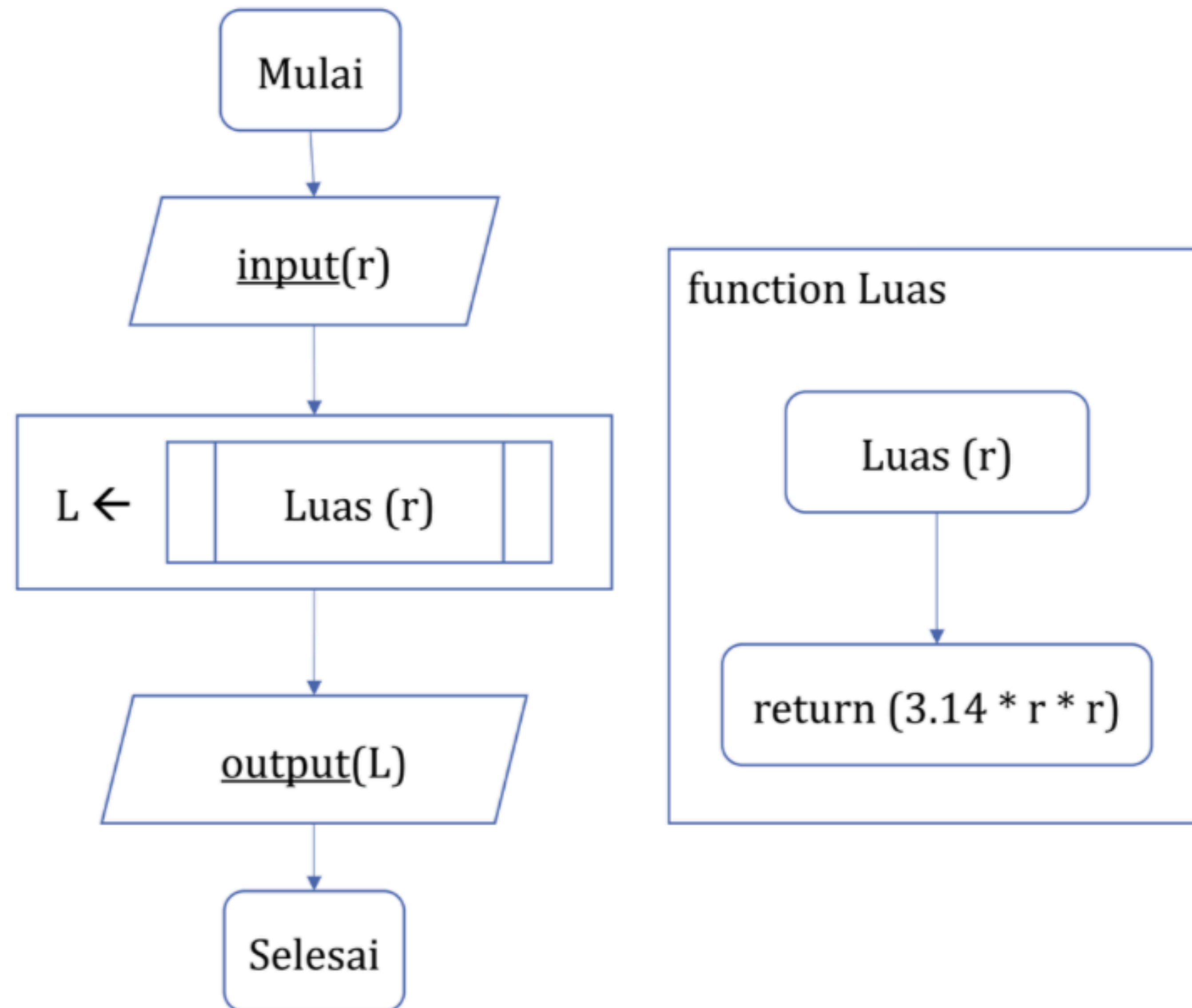
    cout << "Masukkan jari-jari lingkaran: ";
    cin >> r; // Input jari-jari

    L = Luas(r); // Pemanggilan fungsi Luas
    cout << "Luas lingkaran: " << L << endl;

    return 0;
}
```

Contoh1

- Flowchart



Latihan1



- Buatlah definisi dan realisasi fungsi Max3 untuk menghitung nilai maksimum dari 3 bilangan, misalnya A, B, C.
- Contoh: $A = 1, B = -10, C = 5 \rightarrow \text{maksimum} = 5$
- Algoritma:
 - $A \geq B \text{ and } A \geq C : \text{maksimum} = A$
 - $B \geq A \text{ and } B \geq C : \text{maksimum} = B$
 - $C \geq A \text{ and } C \geq B : \text{maksimum} = C$

Menggunakan Fungsi



- Fungsi dipanggil dalam **program utama** atau dalam **fungsi lain** sebagai bagian dari ekspresi atau instruksi.
- Syarat memanggil fungsi:
 - Nama fungsi harus sesuai dengan nama yang dideklarasikan.
 - Jumlah parameter input harus sama, dan tipe data setiap parameter harus sesuai
 - Saat pemanggilan fungsi, terjadi asosiasi satu-ke-satu antara parameter fungsi dan nilai argumen yang diberikan.
 - **Hasil pemanggilan fungsi** harus memiliki tipe data yang sama dengan tipe data output fungsi yang telah dideklarasikan.
 - Pemanggilan fungsi dapat dilakukan sebagai bagian dari sebuah ekspresi

Pemanggilan fungsi sebagai Ekspresi



```
#include <iostream>
using namespace std;

int fungsi1(int a) {
    return a * 2;
}

int fungsi2(int b) {
    return b * b;
}

// Program utama
int main() {
    int x, y, hasil;

    cout << "Masukkan nilai x: ";
    cin >> x;
    cout << "Masukkan nilai y: ";
    cin >> y;

    // Memanggil fungsi sebagai bagian dari ekspresi
    hasil = fungsi1(x) + fungsi2(y);

    cout << "Hasil dari fungsi1(x) + fungsi2(y) adalah: " <<
    hasil << endl;

    return 0;
}
```

Prosedur

Prosedur



- Prosedur: subprogram mengelompokkan instruksi-instruksi yang sering dipakai di program
 - Tidak harus ada parameter input/output
- Prosedur tidak menggunakan return dengan ekspresi/nilai. Jika return digunakan, hanya untuk mengakhiri eksekusi fungsi.
- Prosedur dalam C++ biasanya menggunakan kata kunci `void` untuk menunjukkan bahwa tidak ada nilai yang dikembalikan.

```
void NamaFungsi (TipeDataParameter Parameter)
    //spesifikasi fungsi
}
```

```
void printMessage() {
    cout << "Hello, this is a
    procedure in C++!" << endl;
}
```

```
void printNumbers(int n) {
    for (int i = 1; i <= n; i++) {
        cout << i << " ";
    }
    cout << endl;
}
```

Contoh3

- Buatlah fungsi CetakNama yang menerima masukan sebuah string nama dan mencetak "Hello, " + nama ke layar.
- Tidak ada nilai yang dikeluarkan dari fungsi

```
#include <iostream>
#include <string>
using namespace std;

// Definisi subprogram
void CetakNama(string nama) {
    // Mencetak "Hello" + nama ke layar
    cout << "Hello, " << nama << "!!" << endl;
    return; // Tidak wajib ditulis karena fungsi void
}

// Program utama
int main() {
    // Memanggil prosedur
    string nama = "Dunia";
    CetakNama(nama);
    return 0;
}
```

Memanggil Prosedur



- Karena prosedur tidak menghasilkan nilai, pemanggilannya harus dilakukan sebagai satu baris instruksi mandiri, bukan sebagai bagian dari ekspresi.
- Parameter input dalam prosedur diberikan saat pemanggilan, sama seperti pada fungsi biasa.

Contoh2

- Buatlah program yang menerima masukan sebuah integer > 0 , misalnya N, dan sebuah string, misalnya nama lalu mencetak: "Hello, nama!" sebanyak N kali ke layar

```
#include <iostream>
#include <string>
using namespace std;

void CetakNama(string nama) {
    // Mencetak "Hello" + nama ke layar
    cout << "Hello, " << nama << "!" << endl;
}

int main() {
    string nama;
    int N;

    cout << "Masukkan nama: ";
    cin >> nama;

    cout << "Berapa kali diulang? ";
    cin >> N;

    // Melakukan pengulangan untuk mencetak nama
    for (int i = 0; i < N; i++) {
        CetakNama(nama);
    }

    return 0;
}
```

Fungsi Standar Python



- Dalam C++ didefinisikan sangat banyak fungsi standar yang tersedia dan tinggal digunakan → jadi tidak perlu di-coding lagi
- Fungsi-fungsi standar ini didefinisikan dalam header file.
- Contoh library standar yang sering dipakai adalah cmath
- Fungsi-fungsi yang didefinisikan dalam library math:
 - sqrt → mencari akar kuadrat suatu bilangan
 - sin → mencari sinus
 - cos → mencari cosinus
 - pow → pangkat suatu bilangan
 - dll.
- Memanggil header cmath dengan menambahkan instruksi pada bagian awal program: `#include <cmath>`
- Informasi lebih lanjut: <https://cplusplus.com/reference/cmath/>

Latihan2



- Buatlah sebuah fungsi bernama HitungJarak, yang menerima masukan: v : kecepatan (dalam m/s, bilangan riil) dan t : waktu tempuh (dalam s, bilangan riil) dan menghasilkan jarak tempuh s dengan rumus: $s = v * t$.
- Asumsikan nilai $t \geq 0$ dan $s \geq 0$.
- Selanjutnya, buatlah program utama yang menggunakan fungsi HitungJarak tersebut (bebas).

Latihan3



- Masih ingat program untuk mencari nilai maksimum array?
- Buatlah fungsi MaxArray yang menerima masukan sebuah array of integer, misalnya T, dan panjang array, misalnya N, dan menghasilkan nilai terbesar yang disimpan dalam array tersebut.
- Asumsikan $N > 0$.
- Contoh: $T = [5, 4, 3, 2, 1]$; $N = 5$
maka nilai maksimum = 5

**SELAMAT
BELAJAR**