**Modul 3: Rekursif**

## Tujuan Pembelajaran

Setelah melakukan praktikum dalam bab ini, mahasiswa diharapkan mampu:

1. Memahami mengenai konsep rekursif
2. Mampu memecahkan permasalahan dengan konsep rekursif
3. **Materi**

# 1 Rekursif

**Rekursif** adalah proses ketika sebuah fungsi memanggil dirinya sendiri. Konsep ini mirip dengan **perulangan (looping)**, di mana program akan berjalan terus hingga mencapai **kondisi penghentian (base case)**. Dalam rekursif ada dua komponen utama:

* **Basis (Base Case):** kondisi yang menghentikan rekursif agar tidak berjalan tanpa henti.
* **Rekursif (Recursive Case):** bagian di mana fungsi memanggil dirinya sendiri dengan input yang lebih kecil atau lebih sederhana.

# 2 Proses Rekursif

Tahapan membuat program rekursif di C++ adalah sebagai berikut: Untuk melakukan hitung mundur mulai dari 20 sampai 1.

### Langkah 1 – Deklarasi fungsi rekursif

void write(int n);

### Langkah 2 – Menyediakan variabel parameter

Parameter n akan berfungsi sebagai penghitung mundur.

void write(int n) {

int counter = n; // variabel lokal

### Langkah 3 – Membuat kondisi basis

Jika counter > 0, maka blok perintah rekursif dijalankan. Jika tidak, fungsi berhenti.

if (counter > 0) {

### Langkah 4 – Proses rekursif

Cetak nilai counter, kurangi nilainya, lalu panggil kembali fungsi write(counter-1).

cout << "Saatnya hitung mundur dari " << counter << endl;

write(counter - 1);

}

}

### Langkah 5 – Memanggil fungsi dari main

int main() {

write(20); // mulai dari 20

return 0;

}

**Program Lengkapnya**

#include <iostream>  
using namespace std;  
  
void write(int counter ) {  
 if (counter > 0) {  
 cout << "Saatnya hitung mundur dari " << counter << endl;  
 write(counter - 1);  
 }  
}  
  
int main() {  
 write(20);  
 return 0;  
}

### Output Program:

Saatnya hitung mundur dari 20

Saatnya hitung mundur dari 19

...

Saatnya hitung mundur dari 2

Saatnya hitung mundur dari 1

## C. Praktikum

Tuliskan kode program pada percobaan 1- 3 berikut ini lalu buat ilustrasi proses pemanggilannya! Contoh ilustrasinya dapat dilihat pada percobaan 1

### Percobaan 1 – Faktorial

Persamaan Faktorial pada matimatika

#include <iostream>

using namespace std;

int faktorial(int n) {

if (n == 0 || n == 1)

return 1;

else

return n \* faktorial(n - 1);

}

int main() {

int n = 5;

cout << n << "! = " << faktorial(n) << endl;

return 0;

}

**Berikut ini adalah ilustrasi dari Faktorial di atas!**

Faktorial (4)

=24

4 \* faktorial (3)

6

3 \* faktorial (2)

2

2 \* faktorial (1)

1

faktorial (1)

### Percobaan 2 – Pangkat

#include <iostream>

using namespace std;

int pangkat(int a, int b) {

if (b == 0)

return 1;

else

return a \* pangkat(a, b - 1);

}

int main() {

int a = 3, b = 4;

cout << a << " pangkat " << b << " = " << pangkat(a, b) << endl;

return 0;

}

### Percobaan 3 – Deret Fibonacci

#include <iostream>

using namespace std;

int fibonacci(int n) {

if (n == 0) return 0;

if (n == 1) return 1;

return fibonacci(n-1) + fibonacci(n-2);

}

int main() {

int n = 10;

cout << "Deret Fibonacci hingga " << n << " bilangan:\n";

for (int i = 0; i < n; i++) {

cout << fibonacci(i) << " ";

}

return 0;

}

**D. Tugas**

1. Buat program rekursif untuk menghitung jumlah bakteri. Misalnya, awalnya ada 6 bakteri dan setiap menit jumlahnya berlipat ganda. Hitung jumlah bakteri setelah 10 menit.  
   Petunjuk: gunakan fungsi int bakteri(int menit).
2. Buat program rekursif untuk menghitung saldo dengan bunga tunggal. Misalnya, saldo awal Rp4.000.000 dengan bunga 3% per bulan. Hitung saldo setelah 12 bulan.  
   Petunjuk: gunakan fungsi double saldo(double nilai, int bulan).