

TUGAS ANALISA PRAKTIK ALGORITMA & PEMROGRAMAN



Tugas Minggu ke :
Tanggal Praktek :
Kelompok : 3
Nama Anggota :

1. Ammar Abdullah Al-zahid (2303015012)
2. Fajar Ramadan (2303015032)
3. Dafa Wibisena (2303015079)
4. Fikri Raftanjani (2303015007)

**TEKNIK INFORMATIKAFAKULTAS TEKNOLOGI
INDUSTRI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH PROF. DR. HAMKA
JAKARTA
2023**

LAPORAN PRAKTIKUM

A. RINGKASAN TEORI

REKURSIF VS ITERASI

Rekursi adalah proses pengulangan sesuatu dengan cara kesamaan-diri, sebagai contohnya, saat dua cermin berada paralel antara satu dengan yang lain, gambar yang tertangkap adalah suatu bentuk rekursi tak-terbatas

Bentuk Umum Larik: Rekursif adalah suatu proses yang bisa memanggil dirinya sendiri

Perulangan rekursif merupakan salah satu metode didalam pemrograman yang mana dalam sebuah fungsi terdapat instruksi yang memanggil fungsi itu sendiri, atau lebih sering disebut memanggil dirinya sendiri

Perulangan iteratif merupakan perulangan yang melakukan proses perulangan terhadap sekelompok instruksi

Perulangan dilakukan dalam batasan syarat tertentu. Ketika syarat tersebut tidak terpenuhi lagi maka perulangan akan berhenti

Dengan rekursif dapat menentukan nilai terkecil untuk dimasukan ke nilai yang lebih besar dengan memanggil dirinya sendiri untuk mendapatkan nilai terkecil

Rekursif :

- Sebuah fungsi yang memanggil dirinya sendiri. Contoh klasik adalah fungsi faktorial
- Lebih sederhana untuk dipahami dan diimplementasikan untuk beberapa kasus
- Membutuhkan tambahan overhead karena fungsi harus dipanggil berulang kali
- Dapat menyebabkan stack overflow jika rekursif terlalu dalam

Iterasi :

- Menggunakan perulangan seperti for dan while untuk mengulangi kode
- Lebih efisien dalam kebanyakan kasus, tidak ada overhead dari pemanggilan fungsi berulang
- Lebih sulit dipahami dalam beberapa kasus
- Tidak berisiko stack overflow

LAPORAN PRAKTIKUM

B. ANALISA PERCOBAAN

NAMA

•

NIM

⋮

JUDUL PERCOBAAN

• •

SYNTAX PROGRAM

HASIL PROGRAM

Percobaan ke-

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 unsigned int pangkat (int x, int y) {
5     int hasil = 1, i;
6     for (i=1; i<=y; i++) {
7         hasil = hasil * x;
8     }
9     return hasil; }
10 int main(int argc, char** argv) {
11     int x,y;
12     x = 6;
13     y = 2;
14     cout << x << " Pangkat " << y << " = ";
15     cout << pangkat(x,y) << endl;
16     system ("PAUSE");
17     return 0;
18 }

```

```
D:\TUGAS KULIAH\Semester 1\Praktek Alpro\Iterasi\latihan1.exe
6 Pangkat 2 = 36
Press any key to continue . . .
```

ANALISA PRAKTIKUM

[illegible]

LAPORAN PRAKTIKUM

B. ANALISA PERCOBAAN

NAMA

• •

NIM

⋮

JUDUL PERCOBAAN

•

SYNTAX PROGRAM

HASIL PROGRAM

Percobaan ke-

```

1 #include <iostream>
2 #include <cstdlib>
3 using namespace std;
4 int faktorial(int x) {
5     if (x == 0 || x == 1)
6         return 1;
7     else
8         return (x*faktorial(x-1));
9 }
10 int main (int argc, char *argv[]) {
11     int i;
12     cout << "Mencari Nilai Faktorial \n" << "Masukkan nilai y : ";
13     cin >> i;
14     cout << "Mencari Nilai Faktorial \n" << "adalah " << faktorial(y);
15     cout << endl;
16     system ("PAUSE");
17     return EXIT_SUCCESS;
18 }

```

```
Mencari Nilai Faktorial
Masukkan nilai y: 4
Mencari Nilai Faktorial
adalah 24
Press any key to continue . . .
```

ANALISA PRAKTIKUM

[illegible]

LAPORAN PRAKTIKUM

C. KESIMPULAN

Rekursif itu seperti cermin yang memantulkan bayangannya sendiri. Fungsi rekursif memanggil dirinya sendiri berulang kali sampai mencapai kasus dasar. Contoh sederhananya adalah menghitung faktorial suatu bilangan

Iterasi seperti berjalan lingkaran mengelilingi lapangan. Kita berulang mengerjakan hal yang sama dengan pola tertentu menggunakan perulangan for atau while. Contoh sederhananya adalah mencetak bilangan dari 1 sampai 10.

Kelebihan rekursif:

1. Lebih simpel dan intuitif untuk kasus-kasus tertentu
2. Kode lebih singkat

Kelebihan iterasi:

1. Lebih efisien dan cepat untuk kasus besar
2. Menghindari resiko stack overflow

Jadi, rekursif cocok untuk masalah sederhana. Iterasi lebih baik untuk menangani data dalam jumlah besar. Kadang juga bisa dipakai dengan kombinasi. Pilih mana yang paling sesuai kebutuhan!