TUGAS ANALISA PRAKTIK
ALGORITMA & PEMPROGRAMAN

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 terhenti

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	: Ammar Abdullah Al-Zahid
NIM	: 2303015012
JUDUL PERCOBAAN	:

SYNTAX PROGRAM | **HASIL PROGRAM**

Percobaan ke- 1

```
#include <iostream>
#include <cstdlib>
using namespace std;
unsigned int pangkat (int x, int y) {
int hasil = 1;
for (i=1; i<= y; i++){
hasil = hasil*x; }
return hasil; }
int main(int argc, char** argv) {
int x,y;
x = 6;
y = 2;
cout << x << " pangkat " << y << " ";
cout << pangkat (x,y) <<endl;
system ("PAUSE");
return 0;
}
```

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

ANALISA PRAKTIKUM

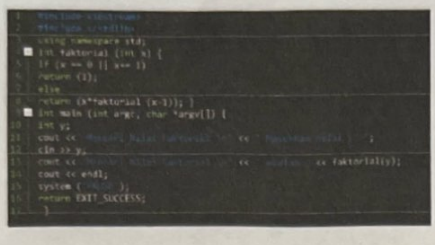
1. mengimport library iostream dan cstdlib untuk mendukung program
2. mendeklarasikan fungsi pangkat() yang menerima 2 parameter (int x dan int y) dan mengembalikan tipe data unsigned int
3. fungsi pangkat() bekerja dengan melakukan perulangan dari 1 sampai y. Didalam perulangan variable hasil dikalikan dengan x sebanyak y kali sehingga menghasilkan nilai x pangkat y.
4. Di fungsi main(), deklarasi 2 variabel x dan y yang nilainya ditentukan masing-masing 6 dan 2
5. Tampilkan x dan y sebagai string "x pangkat y"
6. Panggil fungsi pangkat() dan kirim isi variabel x dan y sebagai parameternya. Hasilnya ditampung dalam variabel bernama hasil.
7. Tampilkan hasil perhitungan pangkat ke layar.
8.

LAPORAN PRAKTIKUM

B. ANALISA PERCOBAAN

NAMA	: Ammar Abdullah Al-Zahid
NIM	: 2303015012
JUDUL PERCOBAAN	:

SYNTAX PROGRAM	HASIL PROGRAM
Percobaan ke-	



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

ANALISA PRAKTIKUM

1. Mengimport library iostream dan cstdlib

2. Mendeklarasikan fungsi rekursif faktorial() yang menerima satu parameter bertipe integer (x) dan mengembalikan nilai integer hasil perhitungan faktorial

3. fungsi faktorial bekerja dengan prinsip rekursif yaitu memanggil dirinya sendiri. Basisnya adalah jika x bernilai 0 atau 1 maka nilai faktorialnya adalah 1

4. Jika x lebih besar dari 1, fungsi akan memanggil dirinya sendiri dengan nilai x-1, dan dilakukan perkalian x dengan hasil pemanggilan fungsi faktorial(x-1)

5. pada fungsi main(), masukkan nilai y diminta dari user.

6. kemudian memanggil fungsi faktorial() dengan argumen y yang nilainya diinputkan user tadi

7. Hasil pemanggilan fungsi faktorial disimpan dalam variabel dan ditampilkan ke user.

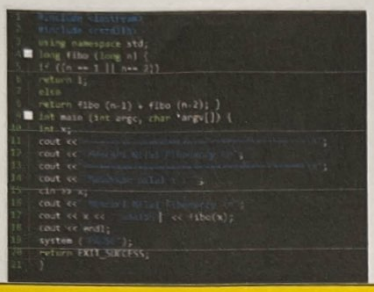
LAPORAN PRAKTIKUM

B. ANALISA PERCOBAAN

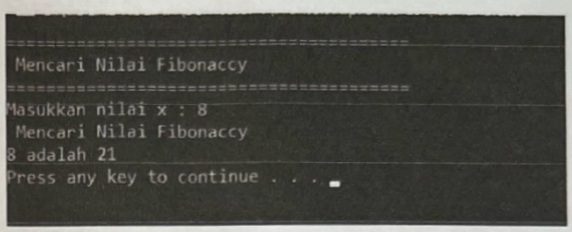
NAMA	: Ammar Abdullah Al-Zahid
NIM	: 2303015012
JUDUL PERCOBAAN	:

SYNTAX PROGRAM | **HASIL PROGRAM**

Percobaan ke-



```
#include <iostream>
#include <cstdlib>
using namespace std;
long fibo (long n) {
if ((n == 1 || n== 2))
return 1;
else
return fibo (n-1) + fibo (n-2); }
int main (int argc, char *argv[]) {
int x;
cout <<
cout <<
cout <<
cout <<
cin>> x;
cout <<
cout << x <<  << fibo(x);
cout << endl;
system (
return EXIT_SUCCESS;
}
```



```
Mencari Nilai Fibonaccy
Masukkan nilai x : 8
 Mencari Nilai Fibonaccy
8 adalah 21
Press any key to continue . . .
```

ANALISA PRAKTIKUM

1. Program mengimport library iostream dan cstdlib

2. Di deklarasikan fungsi rekursif fibo() yang menerima satu parameter bertipe long (n) dan mengembalikan nilai long hasil perhitungan fibonacci.

3. fungsi fibo() bekerja secara rekursif dengan basis bahwa fibo dari 1 atau 2 adalah 1

4. jika n>2. nilai fibo(n) dihitung dengan memanggil fibo(n-1) dan fibo(n-2) kemudian menjumlahkannya.

5. pada fungsi main(). User diminta memasukkan nilai x

6. kemudian memanggil fungsi fibo() dengan argumen x untuk menghitung fibonacci ke-x

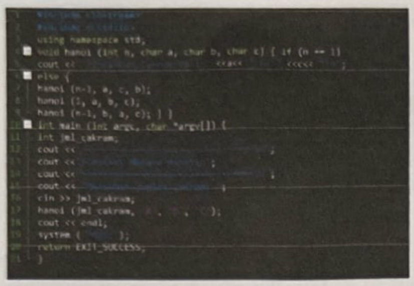
7. Hasilnya disimpan dan ditampilkan ke user.

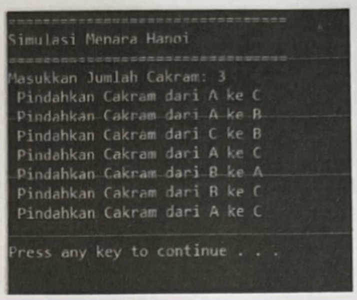
LAPORAN PRAKTIKUM

B. ANALISA PERCOBAAN

NAMA	: Ammar Abdullah Al-Zahid
NIM	: 2303015012
JUDUL PERCOBAAN	:

SYNTAX PROGRAM	HASIL PROGRAM
Percobaan ke-	





ANALISA PRAKTIKUM

1. Program mengimport library iostream dan cstdlib
2. Dideklarasikan fungsi rekursif hanoi() dengan 4 parameters (n,a,b,c)
   - n = jumlah cakram
   - a, b, c = nama tiang
3. Basis : jika n = 1, cetak instruksi pindah cakram dari a ke c
4. Rekursi :
   - Pindahkan n-1 cakram dari a ke b dengan bantuan c.
   - pindahkan 1 cakram dari a ke c
   - pindahkan n-1 cakram dari b ke c dengan bantuan a.
5. Pada main(), diminta input jumlah cakram dari user.
6. Memanggil fungsi hanoi() dengan argumen jumlah cakram dan nama tiang A, B, dan C.

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!