

Nama : Alvin Febrianto
NIM : 21091397031
Prodi / Kelas : Manajemen Informatika / A

C. TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan rekursi?

Rekursi adalah fungsi yang memanggil dirinya sendiri secara berulang-ulang. Saat sebuah fungsi melakukan rekursi, fungsi tersebut membutuhkan percabangan atau perulangan untuk menghentikan rekursi tersebut. Rekursi bisa menjadi seperti "perulangan bercabang" yang memiliki parameter. Jika tidak, maka program akan mengalami error karena kekurangan memori.

2. Tuliskan fungsi untuk menghitung nilai faktorial!

$0! = 1$ Untuk $n = 0$ atau $n = 1$

$n! = n * (n-1)!$ Untuk $n > 1$

Secara pemrograman dapat ditulis sebagai:

Faktorial(0) = 1 (1)

Faktorial(N) = N * Faktorial(N-1) (2)

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

long int faktorial(int A);
int main() {
    int r, hasil;
    cout << "MENGHITUNG NILAI FAKTORIAL DENGAN REKURSIF" << endl;
    cout << endl;
    cout << "Masukan Nilai = ";
    cin >> r;
    hasil = faktorial(r);
    cout << "Faktorial " << r << " != " << hasil << endl;
}

long int faktorial(int A) {
    if (A == 1)
        return (A);
    else
        return (A * faktorial(A - 1));
}
```

3. Tuliskan fungsi untuk menampilkan nilai fibonacci dari deret fibonacci!

$$F(n) = \begin{cases} 0, & \text{jika } n = 0; \\ 1, & \text{jika } n = 1; \\ F(n-1) + F(n-2) & \text{jika tidak.} \end{cases}$$

Fibonacci(1) = 1 (1)

Fibonacci(2) = 1 (2)

Fibonacci(N) = Fibonacci(N-1) + Fibonacci(N-2) (3)

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

int fibonacci(int m) {
    if (m == 0 || m == 1) {
        return m;
    } else {
        return (fibonacci(m - 1) + fibonacci(m - 2));
    }
}

int main() {
    int n, m = 0;
    cout << "Masukan Batas Deret Bilangan Fibonacci: ";
    cin >> n;
    cout << "Deret Fibonacci: ";
    for (int i = 1; i <= n; i++) {
        cout << fibonacci(m) << " ";
        m++;
    }
    return 0;
}
```

D. PERCOBAAN

1. Percobaan 1: Fungsi rekursif untuk menghitung nilai faktorial

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

#include merupakan preprocessor directive yang berfungsi sebagai intruksi untuk memanggil file header yang digunakan. Sedangkan iostream adalah file header yang digunakan untuk memanggil fungsi yang ada pada library file header tersebut seperti fungsi input “cin” dan output “cout”. Using namespace std berarti menggunakan C++ standard library, seperti cin, cout, endl tanpa perlu menulis std::cout, std::cin.

```
long int faktorial (int A);
```

Merupakan pendeklarasian fungsi faktorial menggunakan parameter formal fungsi tipe data int variabel A. Bentuk umum dari pendeklarasian fungsi seperti tipe_fungsi, nama_fungsi, dan parameter_fungsi.

```
int main(){
```

Merupakan fungsi yang berisi statement diawali dengan tanda {}. Fungsi main akan menjalankan statement yang berada di dalamnya secara berurutan.

```
int r, hasil;
```

Merupakan pendeklarasian variabel r dan hasil menggunakan tipe data integer.

```
cout<<"MENGHITUNG NILAI FAKTORIAL DENGAN REKURSIF"<<endl;  
cout<<"Masukan Nilai = ";
```

Merupakan perintah untuk menampilkan teks.

```
cin>>r;
```

Merupakan perintah untuk menginputkan nilai variable r.

```
hasil=faktorial(r);
```

Merupakan pendefinisian variabel hasil yang merujuk pada fungsi faktorial variabel r.

```
cout<<"Faktorial "<<r<<"!= "<<hasil<<endl;  
}
```

Merupakan perintah untuk menampilkan output dalam bentuk kalimat faktorial dengan menampilkan variabel r yang telah diinputkan oleh keyboard dan menampilkan tanda != beserta dengan variabel hasil pada hasil program. Kemudian berpindah pada baris baru. Tanda } merupakan akhir dari fungsi main.

```
long int faktorial (int A){
```

Merupakan fungsi long int faktorial menggunakan parameter formal. Fungsi tipe data int variabel A yang berisi statement diawali dengan tanda {}. Fungsi long int faktorial akan menjalankan statement yang berada di dalamnya secara berurutan.

```
if (A==1)  
return(A);  
else  
return (A*faktorial(A-1));  
}
```

Merupakan kondisi jika nilai yang dimasukkan adalah 1 maka return value adalah nilai itu sendiri. Else return (A*faktorial(A-1)); kondisi jika tidak (selain) maka return value yang dimasukkan akan dihitung menggunakan rumus faktorial tersebut. Tanda } merupakan akhir dari fungsi long int faktorial.

2. Percobaan 2: Fungsi rekursif untuk menampilkan deret fibonacci

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

#include merupakan preprocessor directive yang berfungsi sebagai intruksi untuk memanggil file header yang digunakan. Sedangkan iostream adalah file header yang

digunakan untuk memanggil fungsi yang ada pada library file header tersebut seperti fungsi input “cin” dan output “cout”. Using namespace std berarti menggunakan C++ standard library, seperti cin, cout, endl tanpa perlu menulis std::cout, std::cin.

```
int fibonacci(int m){
```

Merupakan pendeklarasian fungsi fibonacci menggunakan parameter formal fungsi tipe data int variabel A. Bentuk umum dari pendeklarasian fungsi seperti tipe_fungsi, nama_fungsi, dan parameter_fungsi.

```
if (m == 0 || m ==1) {  
    return m;  
}  
else {  
    return (fibonacci(m-1) + fibonacci(m-2));  
}  
}
```

Pada kondisi ini jika nilai variabel m yang diinput adalah 0 atau 1 maka return valuenya adalah nilai itu sendiri. Namun, jika kondisi else maka return value yang diinput akan dihitung menggunakan rumus fibonacci tersebut.

```
int main(){
```

Merupakan fungsi yang berisi statement diawali dengan tanda {}. Fungsi main akan menjalankan statement yang berada di dalamnya secara berurutan.

```
int n, m= 0;
```

Merupakan pendeklarasian variabel n dan m yang bernilai 0 dengan tipe data integer.

```
cout << "Masukan Batas Deret Bilangan Fibonacci: ";  
cin >> n;  
cout << "Deret Fibonacci: ";
```

Merupakan perintah untuk menampilkan teks dan menginputkan nilai variable n.

```
for (int i = 1; i <= n; i++){
```

Merupakan perintah untuk memberi nilai awal perulangan yang dimulai dari I = 1. Perulangan akan berhenti jika nilai I <= (lebih besar sama dengan) inputan N. I++ adalah perulangan yang bersifat increment atau akan ditambah 1 setiap kali proses hingga sampai batas yang ditentukan maka proses akan berhenti.

```
cout << fibonacci(m) <<" ";  
m++;  
}  
return 0;  
}
```

Merupakan perintah untuk menampilkan output fungsi fibonacci variabel m dan memberikan spasi. M++ adalah perulangan yang bersifat increment atau akan ditambah 1 setiap kali proses hingga sampai batas yang ditentukan maka proses akan berhenti.

3. Percobaan 3: Fungsi rekursif untuk menentukan bilangan prima atau bukan prima

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

#include merupakan preprocessor directive yang berfungsi sebagai intruksi untuk memanggil file header yang digunakan. Sedangkan iostream adalah file header yang digunakan untuk memanggil fungsi yang ada pada library file header tersebut seperti fungsi input “cin” dan output “cout”. Using namespace std berarti menggunakan C++ standard library, seperti cin, cout, endl tanpa perlu menulis std::cout, std::cin.

```
int ambil(int bil, int i){
```

Merupakan pendeklarasian variabel “ambil” dengan tipe data integer variabel bil dan i.

```
if (i == 1) {
    return 1;
} else if (bil % i == 0) {
    return 1 + ambil(bil, --i);
} else {
    return 0 + ambil(bil, --i);
}
}
```

Merupakan kondisi jika nilai variabel i yang diinput adalah 1 maka return valuenya adalah nilai 1. Jika kondisi (else if) maka return value yang dimasukkan akan dihitung menggunakan rumus bilangan prima tersebut. Jika tidak (else) maka return value yang dimasukkan akan dihitung menggunakan rumus bilangan prima tersebut.

```
int cek(int bil) {
if (bil > 1) {
return (ambil(bil, bil) == 2);
} else
return false;
}
```

Merupakan kondisi jika nilai variabel bil yang dimasukkan adalah >1 atau bernilai benar maka return value adalah nilai 2. Jika kondisi (else) maka return value yang dimasukkan akan bernilai salah.

```
int main(){
int bil;
cout<<"Masukan Bilangan : ";
cin>>bil;
if (cek(bil)){
    cout<<"Bilangan Prima"<<endl;
} else {
    cout<<"Bukan Bilangan Prima"<<endl;
}
return 0;
}
```

Merupakan pendeklarasian variabel bil, menampilkan teks dan menginput nilai variabel bil. Selanjutnya merupakan pengkondisian yang jika nilai yang diinput bernilai benar maka akan menampilkan output dalam bentuk kalimat Bilangan Prima pada hasil program. Jika kondisi (else) maka akan menampilkan output dalam bentuk kalimat Bukan Bilangan Prima.

4. Percobaan 4: Fungsi rekursif untuk menghitung pangkat

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

#include merupakan preprocessor directive yang berfungsi sebagai intruksi untuk memanggil file header yang digunakan. Sedangkan iostream adalah file header yang digunakan untuk memanggil fungsi yang ada pada library file header tersebut seperti fungsi input “cin” dan output “cout”. Using namespace std berarti menggunakan C++ standard library, seperti cin, cout, endl tanpa perlu menulis std::cout, std::cin.

```
long int pangkatrekursif(int x, int y);
int main(){
int x,y;
```

Merupakan pendeklarasian fungsi long integer pangkat rekursif menggunakan parameter formal fungsi tipe data int variabel A. Bentuk umum dari pendeklarasian fungsi seperti tipe_fungsi, nama_fungsi, dan parameter_fungsi.

```
cout<<"FUNGSI REKURSIF UNTUK MENGHITUNG PANGKAT"<<endl;
cout<<"Masukan Nilai X = "; cin>>x;
cout<<"Masukan Nilai Y = "; cin>>y;
cout<<x<<" Dipangkatkan "<<y<<" = "<<pangkatrekursif(x,y)<<endl;
}
long int pangkatrekursif(int x, int y)
```

Merupakan fungsi long integer dengan menggunakan parameter formal fungsi tipe data integer variabel x dan y yang berisi statement dan diawali dengan tanda { }. Fungsi long int pangkat rekursif akan menjalankan statement yang berada di dalamnya secara berurutan.

```
{
if (y==0)
return 1 ;
else
return x * pangkatrekursif(x,y-1); }
```

Merupakan kondisi jika nilai variabel y yang dimasukkan adalah 1 maka return valuenya adalah nilai 1. Jika kondisi (else) maka return value yang diinput akan dihitung menggunakan rumus pangkat tersebut.

E. LATIHAN

1. Buatlah program rekursif untuk menghitung segitiga pascal!

```
#include <iostream>
#include <conio.h>
using namespace std;
```

#include merupakan preprocessor directive yang berfungsi sebagai intruksi untuk memanggil file header yang digunakan. Sedangkan iostream adalah file header yang digunakan untuk memanggil fungsi yang ada pada library file header tersebut seperti fungsi input “cin” dan output “cout”. Using namespace std berarti menggunakan C++ standard library, seperti cin, cout, endl tanpa perlu menulis std::cout, std::cin.

```
long faktorial(int n) {
    long z = 1;
    int i = 1;
```

Long int merupakan variasi tipe data yang berfungsi untuk variabel yang nilainya berjumlah sangat banyak, dan memiliki fungsi bernama faktorial yang memiliki argument bertipe ‘int’ dan bervariasi n. Merupakan pendeklarasian variabel ‘z’ yang bertipe data ‘long’ dan ‘i’ yang bertipe data ‘int’ dan bernilai 1.

```
while(i<=n){
    z=z*i;
    i++;}
return z;
```

Merupakan proses perulangan dengan nilai i akan terus bertambah sampai dengan nilai maks sesuai nilai ‘n’ yang diinputkan, lalu akan memiliki nilai balik z, yang akan diproses dengan $z=z*i$.

```
int main() {
    int i, j;
```

Merupakan pendeklarasian fungsi main yang bertipe data ‘int’ dan pendeklarasian variabel ‘i’ dan ‘j’ yang bertipe data ‘int’.

```
cout<<" ===== " <<endl;
cout<<"|      SEGITIGA PASCAL      |" <<endl;
cout<<" ===== " <<endl<<endl;
```

Menampilkan output yang berupa teks.

```
for (i=0; i<6; i++) {
for (j=0; j<6-i-1; j++){
cout<<" ";
}
for (j=0; j<=i; j++){
cout<<faktorial(i)/(faktorial(j)*faktorial(i-j))<<" ";
}
    cout << endl;
}
```

Merupakan program pengulangan bersarang nested loop, dimana nilai akan terus bertambah sampai <6, kemudian akan dioutputkan sesuai rumus faktorial(i)/(faktorial(j)*faktorial(i-j)), dan diakhiri dengan <<endl untuk mengakhiri line.

2. Buatlah program secara rekursif, masukkan jumlah N karakter dan cetak dalam semua kombinasi!

```
#include <iostream>
#include <conio.h>
using namespace std;
```

#include merupakan preprocessor directive yang berfungsi sebagai intruksi untuk memanggil file header yang digunakan. Sedangkan iostream adalah file header yang digunakan untuk memanggil fungsi yang ada pada library file header tersebut seperti fungsi input “cin” dan output “cout”. Using namespace std berarti menggunakan C++ standard library, seperti cin, cout, endl tanpa perlu menulis std::cout, std::cin.

```
void printAllKLengthRec(char set[], string prefix, int n, int k){
    if (k == 0){
        cout<<(prefix)<<endl;
        return;
    }

    for (int i = 0; i < n; i++){
        string newPrefix;
        newPrefix = prefix + set[i];
        printAllKLengthRec(set, newPrefix, n, k - 1);
    }
}
```


Merupakan program fungsi dari rekursi kombinasi, yang di dalamnya terdapat nilai yang akan digunakan untuk membuat rangkaian rekursif kombinasi.

```
void printAllKLength(char set[], int k,int n)
{
    printAllKLengthRec(set, "", n, k);
}
```

Merupakan pendeklarasian fungsi dan beberapa variabel, dan juga program untuk mengatur panjang rangkaian kombinasi tersebut.

```
int main(){
char ch[] = {'a','b','c','d','e','f','g','h','i','j','k',
'l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};
int N;
cout<<"\njumlah karakter : "<<' ';
```

Pendeklarasian fungsi main dan variabel ‘ch’ yang bertipe data ‘char’ dan berisi abjad a-z serta variabel ‘N’ yang bertipe data ‘int’. Selanjutnya menampilkan output teks.

```
while(cin >> N){
    if(N != -1){
        printAllKLength(ch,N, N);
    } else if(N>26){
        cout<<"karakter yang anda masukkan salah!";
        break;
    } else{
        break;
    }
}
```

Merupakan keterangan dari sebuah program kombinasi. Di dalamnya terdapat jumlah karakter yang akan digunakan dalam suatu program kombinasi. Apabila jumlahnya lebih dari 26 maka akan muncul output tersebut.

3. Buat program binary search dengan rekursif! (data tentukan sendiri)

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#define size 10
using namespace std;
```

- #include merupakan preprocessor directive yang berfungsi sebagai intruksi untuk memanggil file header yang digunakan. Sedangkan iostream adalah file header yang digunakan untuk memanggil fungsi yang ada pada library file header tersebut seperti fungsi input “cin” dan output “cout”. Using namespace std berarti

menggunakan C++ standard library, seperti cin, cout, endl tanpa perlu menulis std::cout, std::cin.

- Script #define size 10 artinya membuat konstanta bernama size yang diisi nilai 10.

```
int binsearch(int[], int, int, int);
int binsearch(int a[], int x, int low, int high) {
    int mid;
    if (low > high)
        return -1;
    mid = (low + high) / 2;
    if (x == a[mid]) {
        return (mid);
    } else if (x < a[mid]) {
        binsearch(a, x, low, mid - 1);
    } else {
        binsearch(a, x, mid + 1, high);
    }
}
```

Merupakan algoritma fungsi binary search dan menggunakan if else, dimana akan dibagi 2 elemen yaitu low dan high. Jika nilai $x=a[mid]$ akan dikembalikan ke nilai mid. Jika $x < a[mid]$ maka akan dimasukkan ke (a, x, low, mid-1). Selain itu akan dimasukkan ke (a, x, mid + 1, high).

```
int main() {
    int num, i, key, position;
    int low, high, list[size];
```

Merupakan pendeklarasian fungsi 'main' bertipe data 'int' dan pendeklarasian beberapa variabel bertipe data 'int'.

```
cout<<"\nMasukan Jumlah Besar data:"; cin>>num;
cout<<"\nMasukan Data:\n";
for (i=0; i<num; i++) {
    cin>>list[i];
}
```

Merupakan beberapa output dan input untuk memasukkan data-data yang nantinya bisa dicari.

```

low = 0; high = num - 1;
cout<<"\nMasukan data yang ingin di cari : "; cin>>key;
position = binsearch(list, key, low, high);
if (position != -1) {
    cout<<"\nData "<<key<<" berada pada Index ke - "<<position<<endl;
} else cout<<"\nData tidak di temukan"<<endl;
return (0);
}

```

Pada script ini akan diinput data yang akan dicari, kemudian akan diproses posisinya dengan command `position = binsearch(list, key, low, high)`, kemudian akan ditampilkan datanya jika ditemukan dan akan ditampilkan `cout<<"\nData tidak di temukan"<<endl;` jika tidak ada datanya.

4. Buatlah program rekursif untuk memecahkan permasalahan Menara Hanoi!

```

#include <stdio.h>
#include <iostream>
using namespace std;

```

`#include` merupakan preprocessor directive yang berfungsi sebagai intruksi untuk memanggil file header yang digunakan. Sedangkan `iostream` adalah file header yang digunakan untuk memanggil fungsi yang ada pada library file header tersebut seperti fungsi input “cin” dan output “cout”. Using namespace std berarti menggunakan C++ standard library, seperti cin, cout, endl tanpa perlu menulis `std::cout`, `std::cin`.

```

void Hanoi(int n, char dari_pasak, char ke_pasak, char pasak_tmp){
    if (n == 1){
        cout<<"\nPindahkan disc 1 dari pasak "<<dari_pasak<<" ke pasak
"<<ke_pasak;
        return;
    }
    Hanoi(n-1, dari_pasak, pasak_tmp, ke_pasak);
    cout<<"\nPindahkan disc "<<n<<" dari pasak "<<dari_pasak<<" ke
pasak "<<ke_pasak;
    Hanoi(n-1, pasak_tmp, ke_pasak, dari_pasak);
}

```

Script ini merupakan pengaturan dari sebuah program menara hanoi agar disc dapat berpindah dari pasak ke pasak lainnya sesuai dengan yang diperintahkan.

```

int main(){
    int n;
    cout<<"Masukan Jumlah Disc : ";cin>>n;
    Hanoi(n, 'A', 'C', 'B');
    return 0;
}

```

Script ini merupakan program untuk memasukkan jumlah disc yang akan dipindahkan dari pasak ke pasak tersebut.

5. Jelaskan proses rekursif untuk program di bawah ini!

```

void decToBin(int num) {    /*menjelaskan tentang fungsi konversi
                             dari desimal ke biner*/
    if (num > 0) {          /*jika data yang disimpan pada variabel num
                             lebih besar dari 0 maka akan dibagi dengan 2
                             dan ditampilkan sisa hasil pembagiannya*/
        decToBin(num / 2);
        cout << num % 2;
    }
}

```

6. Jelaskan proses rekursif untuk program di bawah ini!

```

boolean search(int[] x, int size, int n) {    /*menjelaskan tentang
                                                ukuran dari sebuah
                                                persamaan n dengan
                                                boolean*/
    if (size > 0) {    //menjelaskan ukuran dari sebuah bilangan n
        if (x[size - 1] == n) {
            return true;
        } else {
            return search(x, size - 1, n);
        }
    }
    return
    false;
}

```

7. Jelaskan proses rekursif untuk program di bawah ini!

```
boolean binarySearch(int[] x, int start, int end, int n) {  
    //menghitung angka biner menggunakan boolean  
    if (end < start)  
        return false;  
    int mid = (start + end) / 2;    /*hasil awal dan akhir dijumlah dan  
                                   dibagi 2*/  
  
    if (x[mid] == n) {  
        return true;  
    } else {  
        if (x[mid] < n) {  
            return search(x, mid + 1, end, n);  
        } else {  
            return search(x, start, mid - 1, n);  
        }  
    }  
}
```

8. Jelaskan proses rekursif untuk program di bawah ini dengan memanggil mystery(2, 25) and mystery(3,11)!

```
int mystery(int a, int b) {    //deklarasi  
    if (b == 0)  
        return 0;  
    if (b % 2 == 0)  
        return mystery(a + a, b / 2);    /*jumlah nilai akan dijumlah  
                                           dan dibagi 2*/  
    return mystery(a + a, b / 2) + a;  
}
```

9. Jelaskan proses rekursif untuk program di bawah ini dengan memanggil mystery(0, 8)!

```
int mystery(int a, int b) {  
    if (a == b) cout << a << endl;  
    else {  
        int m1 = (a+b)/2;    //nilai a dan b akan dijumlah dan dibagi 2  
        int m2 = (a+b+1)/2;    //nilai a dan b dijumlah dengan 1 dibagi 2  
        mystery(a, m1);  
        mystery(m2, b);  
    }  
}
```

10. Jelaskan proses rekursif untuk program di bawah ini!

```
int f(int n) {                                //program nilai n
    if (n == 0)
        return 0;
    if (n == 1)
        return 1;
    if (n == 2)
        return 1;
    return 2 * f(n-2) + f(n-3);                //nilai akan dijumlahkan
}
```

11. Jelaskan proses rekursif untuk program di bawah ini dengan memanggil square(5), cube(5), cube(123)!

```
int square(int n) {
    if (n == 0)
        return 0;
    return square(n-1)+2*n-1;                  /*nilai akan dikurang 1 ditambah 2
                                                dan nilai lainnya akan dikurang 1*/
}

int cube(int n) {
    if (n == 0) return 0;
    return cube(n-1)+3*(square(n))-3*n+1;
}
```