

PENDAHULUAN

1. Apa yang dimaksud dengan rekursi?

Rekursif adalah suatu proses atau prosedur dari fungsi yang memanggil dirinya sendiri secara berulang-ulang. Karena proses dalam Rekursif ini terjadi secara berulang-ulang maka harus ada kondisi yang membatasi pengulangan tersebut, jika tidak maka proses tidak akan pernah berhenti sampai memori yang digunakan untuk menampung proses tersebut tidak dapat menampung lagi/penuh.

2. Tuliskan fungsi untuk menghitung nilai factorial

```
#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 fibonanci dari deret fibonanci

```
#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;
}
```

Percobaan

1. Analisa fungsi rekursif untuk menghitung nilai faktorial

Pada program sekuensial ini maka perlu inisialisasi fungsi rekrusifnya di awal sebelum fungsi (main). Pada fungsi main pengguna akan memasukkan nilai dan disimpan pda variabel r nantinya akan dipanggil fungsi faktorial() dengan nilai parameter yang

dibawah adalah nilai r tersebut, lalu akan tersimpan pada variabel hasil. Dan jika nilai yang dimasukkan adalah 1 maka nilai balik (return value) adalah nilai itu sendiri, sementara jika tidak maka akan dihitung menggunakan rumus faktorial yaitu $(A * \text{faktorial}(A-1))$.

2. Analisa fungsi rekursif untuk menampilkan deret fibonanci

Dalam fungsi ini terdapat percabangan if dimana jika nilai m adalah 0 atau 1 maka nilai kembali atau return value pada fungsi tersebut adalah nilai itu sendiri yaitu 0 dan 1. Karena algoritma deret bilangan fibonanci C++ adalah penjumlahan dari dua bilangan penjumlahan sebelumnya maka kita perlu mendapat 2 nilai awal yaitu 0 dan 1 agar dapat dijumlahkan dan menjadi nilai pada deret selanjutnya. Apabila dijalankan program tersebut akan menghasilkan output yang sama dengan program sebelumnya.

3. Analisa fungsi rekursif untuk menentukan bilangan prima atau bukan prima

Program ini akan menentukan bilangan yang dimasukan oleh pengguna apakah bilangan prima atau bukan dengan menggunakan fungsi rekursif. Dengan kata lain kita dapat melakukan pemanggilan berulang-ulang pada fungsi sesuai dengan kondisi yang ditetapkan. Beberapa fungsi selain fungsi utama main () yaitu fungsi ambil () dan cek (). Cara kerjanya adalah pengguna akan memasukan bilangan yang ingin ditentukan apakah bilangan prima atau bukan, kemudian akan di cek bilangan tersebut melalui fungsi cek (). Fungsi cek () berfungsi untuk mengecek apakah bilangan yang dimasukan adalah bilangan yang lebih dari 1 (karena bilangan prima dimulai dari 2) jika kondisi bernilai true maka selanjutnya akan di proses pada fungsi ambil (). Pada fungsi ambil() nilai akan dicek selama nilai i belum sama dengan 1 maka nilai bil akan terus dilakukan modulus pada nilai i dengan catatan nilai i akan ditambahkan 1 jika hasil $\text{bil} \% i \neq 0$, jika kondisi tidak terpenuhi ditambah 0. Fungsi cek akan menghasilkan nilai boolean (true/false) jika true maka akan menampilkan kalimat 'Bilangan Prima' jika tidak menampilkan kalimat 'Bukan Bilangan Prima'.

4. Analisa fungsi rekursi untuk menghitung bilangan pangkat.

Pada contoh fungsi rekursi c++ untuk menghitung nilai pangkat, pengguna akan memasukkan nilai x dan y lalu muncul output nilai x angka yang dipangkatkan dengan nilai y. fungsi dengan nama pangkat rekursif dengan menggunakan 2 parameter yaitu nilai x dan y, jika nilai y yang dimasukan adalah 0 maka akan di set nilai baliknya adalah 1, namun jika tidak maka fungsi tersebut di set nilai balik nya dimana nilai x akan dikalikan dengan nilai y-1.

Analisa

1. Program rekursif untuk menghitung segitiga Pascal

```

#include <iostream> using
namespace std;

long faktorial(int n) {
    long z = 1; int i =
    1;
while(i<=n) {
    z=z*i; i++;
    }
return z;
}

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

cout<<"Masukkan nilai: ";
cin >> a;

for (i=0; i<a; i++) {
    for (j=0; j<a-i-1; j++){
        cout << " ";
    }
}

```

Analisa:

```

#include <iostream> using namespace
std;

```

Merupakan pemberian intruksi untuk preprocessor directive **#include** agar menyertakan file header **iostream** sehingga memungkinkan penggunaan perintah cin, cout, dan endl dalam program ini. *Using namespace std;* berkaitan dengan fungsi dari file header iostream karena berisi tentang perintah seperti cout, cin, dan endl. Using namespace std; merupakan pernyataan (statement) yang selalu diakhiri dengan tanda ; (titik koma).

```

long faktorial (int n){

```

Merupakan pendeklarasian fungsi factorial dengan menggunakan parameter formal fungsi tipe data long variable n dengan tipe data integer. Bentuk umum dari pendeklarasian fungsi seperti tipe_fungsi nama_fungsi (parameter_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi long faktorial..

```
long z=1; int i=1;
```

Merupakan pendeklarasian variabel z adalah 1 dengan tipe data long dan variable i adalah 1 dengan tipe data integer.

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

Merupakan kondisi while akan berjalan ketika variabel counter i kurang dari samadengan variabel n. Ketika nilai *variabel counter* i sudah mencapai n, maka kondisi **while (i <= n)** tidak terpenuhi lagi (**false**), sehingga perulangan berhenti. Tanda { (kurung kurawal buka) merupakan awal dari fungsi while. $Z=z*i$; pendeklarasian rumus variabel z pada fungsi while. $i++$; perulangan bersifat increment atau akan ditambah 1 setiap kali proses hingga sampai batas yang ditentukan maka proses akan berhenti. Tanda } (kurung kurawal tutup) merupakan akhir dari fungsi while.

```
return z;  
}
```

Merupakan *statement* untuk mengembalikan nilai pada variable z ke fungsi pemanggilan.

```
int main(){
```

Merupakan fungsi main yang berisi statement-statement yang selalu diawali dengan tanda { (kurung kurawal buka) dan diakhiri dengan tanda } (kurung kurawal tutup). Fungsi main akan menjalankan statement-statement yang berada di dalamnya secara berurutan. Tanda { (kurung kurawal buka) merupakan awal dari fungsi main.

```
int a,i,j;
```

Merupakan pendeklarasian variabel a, i, dan j dengan tipe data integer.

```
cout<<"Masukan nilai : ";
```

Merupakan perintah untuk menampilkan teks **Masukan nilai** :pada layar.

```
cin>>a;
```

Merupakan perintah untuk menginputkan nilai variable a.

```
for (i=0; i<a; i++) {
```

Merupakan pendeklarasian *statement looping* for. Disini i=0 merupakan keadaan awal dimana i memiliki nilai 0 dengan syarat i kurang dari sama dengan a dengan perubahan dijumlahkan dengan 1.

```
for (j=0; j<a-i-1; j++){
```

Merupakan pendeklarasian *statement looping* for. Disini j=0 merupakan keadaan awal dimana j memiliki nilai 0 dengan syarat i kurang dari sama dengan a-1 dengan perubahan dijumlahkan dengan 1.

```
cout << " ";  
}
```

Merupakan proses menampilkan " " pada layar.

```
for (j=0; j<=i; j++){
```

Merupakan pendeklarasian *statement looping* for. Disini j=0 merupakan keadaan awal dimana j memiliki nilai 0 dengan syarat i kurang dari sama dengan i dengan perubahan dijumlahkan dengan 1

```
cout << faktorial(i) / (faktorial(j) * faktorial(i - j)) << " ";  
}
```

Merupakan proses menampilkan output hasil nilai variabel faktorial(i) / (faktorial(j) * faktorial(i - j)) dengan kalimat Dipangkatkan nilai variabel y yang sama-sama diinputkan oleh keyboard dan menampilkan tanda = beserta dengan nilai variabel pangkatrekursif(x,y) pada hasil program. Tanda } (kurung kurawal tutup) merupakan akhir dari fungsi for.

```
return 0;  
}
```

Menyatakan hasil keluaran fungsi main adalah 0 atau untuk memberikan exit status yang berarti program berakhir dengan normal. Tanda } (kurung kurawal tutup) merupakan akhir dari fungsi main.

2. Program, secara rekursif, masukkan jumlah N karakter dan cetak semua kombinasi

```

#include <bits/stdc++.h> using
namespace std;

void printLebarChar(char set[], string huruf,int n, int k)
{ if (k == 0)
    {
        cout << (huruf) ;
        cout << " ";
        return;
    } for (int i = 0; i < n;
    i++)
    { string newHuruf;
        newHuruf = huruf + set[i]; printLebarChar(set,
        newHuruf, n, k - 1);
    }
}

void inputLebarChar(char set[], int k,int n)

```

{
Analisa:

```

#include <bits/stdc++.h> using namespace
std;

```

Merupakan header yang akan meng-include semua standar library. File header bits/stdc++.h berfungsi agar kita tidak meng-includekan header library lainnya. *Using namespace std;* berkaitan dengan fungsi dari file header iostream karena berisi tentang perintah seperti cout, cin, dan endl. Using namespace std; merupakan pernyataan (statement) yang selalu diakhiri dengan tanda ; (titik koma).

```

void printLebarChar(char set[], string huruf,int n, int k)
{

```

Merupakan pendeklarasian fungsi void dengan menggunakan parameter fungsi tipe data char variabel set[], tipe data string variabel huruf, tipe data int variabel n dan k. Bentuk umum dari pendeklarasian fungsi seperti tipe_fungsi nama_fungsi (parameter_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi void.

Merupakan kondisi jika k sama dengan 0 maka akan mencetak variabel huruf, kemudian nilai pada variabel akan kembali ke nilai awal.

```

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

```

```

        newHuruf = huruf + set[i];

```

Merupakan pendeklarasian *statement looping* for. Disini i=0 merupakan keadaan awal dimana i memiliki nilai 0 dengan syarat i kurang dari sama dengan n dengan perubahan dijumlahkan dengan 1. *String newHuruf*; pendeklarasian variabel newhuruf tipe data string yang berada dalam fungsi for. *newHuruf=huruf + set [i]*; pendeklarasian rumus variabel new huruf pada fungsi for. *printLebarChar(set, newHuruf, n, k - 1);* pendeklarasian variabel lebar tipe data char yang berada dalam fungsi for. Tanda } (kurung kurawal tutup) merupakan akhir

```

void inputLebarChar(char set[], int k,int n)
{

```

Merupakan pendeklarasian fungsi void dengan menggunakan parameter fungsi tipe data char variabel set[], tipe data int variabel n dan k. Bentuk umum dari pendeklarasian fungsi seperti tipe_fungsi nama_fungsi (parameter_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi void.

```

printLebarChar(set, "", n, k);
}

```

Merupakan pendeklarasian variabel lebar tipe data char yang berada dalam fungsi for.

```

int main(){

```

Merupakan fungsi main yang berisi statement-statement yang selalu diawali dengan tanda { (kurung kurawal buka) dan diakhiri dengan tanda } (kurung kurawal tutup). Fungsi main akan menjalankan statement-statement yang berada di dalamnya secara berurutan. Tanda { (kurung kurawal buka) merupakan awal dari fungsi main

```

char set1[] = {'a', 'b', 'c'};

```

Merupakan pendeklarasian variabel set1[] tipe data char yang berada dalam fungsi main. *int k=3* pendeklarasian nilai variabel k tipe data int yang berada dalam fungsi main.

```
int k = 3;
```

Merupakan pendeklarasian variabel k adalah 3 dengan tipe data integer.

```
cout << "Jumlah karakter : " ;
```

Merupakan perintah untuk menampilkan teks Jumlah karakter : pada layar.

```
cin >> k;
```

Merupakan perintah untuk menginputkan nilai variable k.

```
inputLebarChar(set1, k, 3);  
}
```

Merupakan pendeklarasian rumus input variabel lebar tipe data char. Tanda } (kurung kurawal tutup) merupakan akhir dari fungsi void 3.

Program BinarySearch dengan Rekursif! (Data tentukan sendiri)

```
#include <iostream>  
#include <conio.h>  
#include <iomanip>  
using namespace std;  
  
int data[13] = {2, 5, 8, 10, 14, 32, 35, 41, 67, 88, 90, 101, 109};  
int cari;  
  
void selection_sort()  
{ int temp, min, i, j;  
for(i=0; i<7;i++)  
    { min = i; for(j = i+1;  
                  j<7; j++)  
        {  
  
  
            if(data[j]<data[min])
```



```

void binarysearch()
{
    //searching
    int awal, akhir, tengah, b_flag = 0;
    awal = 0; akhir = 7; while (b_flag
    == 0 && awal<=akhir)
    {
        tengah = (awal + akhir)/2; if(data[tengah]
        == cari)
        {
            b_flag = 1; break;
        }
        else if(data[tengah]<cari)
            awal = tengah + 1; else
            akhir = tengah -1;
    }
    if(b_flag == 1) cout<<"\nData ditemukan pada index ke-
        "<<tengah<<endl; else cout<<"\nData tidak ditemukan\n";
} int
main()
{
    cout<<"\t 'BINARY SEARCH'"<<endl;
    cout<<"\t======"<<endl;

    cout<<"\nData          : ";

    //tampilkan data awal

```

for(int x = 0; x<7; x++) Analisa:

```

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

```

Merupakan pemberian intruksi untuk preprocessor directive **#include** agar menyertakan file header **iostream** sehingga memungkinkan penggunaan perintah cin, cout, dan endl

dalam program ini. *Using namespace std;* berkaitan dengan fungsi dari file header *iostream* karena berisi tentang perintah seperti *cout*, *cin*, dan *endl*. *Using namespace std;* merupakan pernyataan (statement) yang selalu diakhiri dengan tanda ; (titik koma). *#include <conio.h>* adalah header yang melibatkan “*getch()*” yang berfungsi agar program tidak tertutup setelah semua program selesai dieksekusi sehingga pengguna dapat melihat hasil dari perhitungan program. *#include <iomanip>* memasukkan perintah untuk meng include kan file header *iomanip* sebelum proses compile berlangsung (preprocessing). File header *iomanip* berfungsi untuk melakukan operasi *setw()* .

```
int data[13] = {2, 5, 8, 10, 14, 32, 35, 41, 67, 88, 90, 101, 109};
```

Merupakan pendeklarasian array (data) [7] tipe data int. *int cari;* pendeklarasian variabel cari tipe data int.

```
int cari;
```

Merupakan pendeklarasian variabel cari dengan tipe data integer.

```
void selection_sort(){
```

Merupakan pendeklarasian fungsi void *selection_sort* kombinasi antara sorting dan searching. Untuk setiap proses, akan dicari elemen-elemen yang belum diurutkan yang memiliki nilai terkecil atau terbesar akan dipertukarkan ke posisi yang tepat di dalam array. Bentuk umum dari pendeklarasian fungsi seperti tipe_fungsi nama_fungsi (parameter_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi void

```
for(i=0; i<7;i++){
```

Merupakan pendeklarasian *statement looping* for. Disini *i=0* merupakan keadaan awal dimana *i* memiliki nilai 0 dengan syarat *i* kurang dari 7 dengan perubahan dijumlahkan dengan 1

```
min = i;
```

Merupakan pendeklarasian variabel min sama dengan 1.

```
for(j = i+1; j<7; j++){
```

Merupakan pendeklarasian *statement looping* for. Disini i+0 merupakan keadaan awal dimana i memiliki nilai ditambah 1 dengan syarat i kurang dari 7 dengan perubahan dijumlahkan dengan 1.

```
    if(data[j]<data[min]){  
                                min=j;  
    }
```

Merupakan kondisi jika nilai variabel data[j] yang dimasukkan lebih kecil dari variabel data[min] maka nilai min=j;

```
    temp = data[i];  
    data[i] =  
    data[min];  
    data[min] = temp;  
    }
```

Merupakan pendeklarasian nilai variabel temp sama dengan variabel data[i], nilai variabel data[i] sama dengan variabel data[min], dan nilai variabel data[min] sama dengan variabel temp.

```
void binarysearch(){
```

Merupakan pendeklarasian fungsi void binarysearch metode pencarian dimana data harus diurutkan terlebih dahulu sebelum dilakukan proses pencarian. Bentuk umum dari pendeklarasian fungsi seperti tipe_fungsi nama_fungsi (parameter_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi void.

```
int awal, akhir, tengah, b_flag = 0;    awal = 0;
```

```
    akhir = 7;
```

Merupakan pendeklarasian variabel awal, akhir, tengah, b_flag dengan tipe data integer. Nilai variabel b_flag sama dengan 0, nilai variabel awal sama dengan 0, dan nilai variabel akhir adalah 7.

```
while (b_flag == 0 && awal<=akhir)
```

Merupakan kondisi while akan berjalan ketika variabel counter `b_flag = 0` dan awal { kurang dari samadengan variabel akhir. Ketika nilai *variabel counter* tidak sesuai maka kondisi while tidak terpenuhi lagi (**false**), sehingga perulangan berhenti. Tanda { (kurung kurawal buka) merupakan awal dari fungsi while..

```
tengah = (awal + akhir)/2;
```

Merupakan pendeklarasian variabel dengan adalah variabel awal ditambah variabel akhir dibagi 2.

```
if(data[tengah] == cari){  
    b_flag =  
    1;
```

Merupakan kondisi jika nilai variabel `data[tengah]` yang dimasukkan lsama sengan nilai dari variabel `cari` maka nilai `b_flag` sama dengan 1. *Break*; pendeklarasian break berfungsi untuk menghentikan pernyataan pengulangan.

```
else if(data[tengah]<cari) awal  
    = tengah + 1;
```

Merupakan kondisi jika nilai variabel `data[tengah]` yang dimasukkan lebih kecil dari variabel `cari` maka nilai awal sama dengan nilai variabel tengah ditambah 1.

```
else  
    akhir = tengah -1;
```

Merupakan kondisi jika semua syarat tidak terpenuhi maka nilai akhir sama dengan nilai tengah dikurang 1.

```
if(b_flag == 1)  
    cout<<"\nData ditemukan pada index ke-"<<tengah<<endl;  
    else  
        cout<<"\nData tidak ditemukan\n";
```

Merupakan kondisi jika nilai variabel `b_flag` sama dengan 1 maka akan mencetak teks Data ditemukan pada index ke- dan jika syarat tidak terpenuhi maka akan mencetak teks Data tidak ditemukan.

```
int main(){
```

Merupakan fungsi main yang berisi statement-statement yang selalu diawali dengan tanda { (kurung kurawal buka) dan diakhiri dengan tanda } (kurung kurawal tutup). Fungsi main akan menjalankan statement-statement yang berada di dalamnya secara berurutan. Tanda { (kurung kurawal buka) merupakan awal dari fungsi main

```
cout<<"\t\t\t' BINARY SEARCH '"<<endl;
```

Merupakan perintah untuk menampilkan teks dilanjutkan dengan perintah untuk menampilkan teks 'BINARY SEARCH' berikutnya di baris baru.

```
cout<<"\nData          : ";
```

Merupakan perintah untuk menampilkan teks dilanjutkan dengan perintah untuk menampilkan teks Data.

```
for(int x = 0; x<7; x++)
```

Merupakan pendeklarasian *statement looping* for. Disini x=0 merupakan keadaan awal dimana i memiliki nilai 0 dengan tipe data integer, dengan syarat x kurang dari 7 dengan perubahan dijumlahkan dengan 1

```
cout<<setw(3)<<data[x];
```

Merupakan perintah untuk mengatur lebar variabel dan nilai dari variabel array data[x].

```
cout<<"\nMasukkan data yang ingin Anda cari : ";
```

Merupakan perintah untuk menampilkan teks dilanjutkan dengan perintah untuk menampilkan teks Masukkan data yang ingin Anda cari :

```
cin>>cari;
```

Merupakan perintah untuk menginputkan nilai variable cari.

```
cout<<"\nData diurutkan : ";
```

Merupakan perintah untuk menampilkan teks dilanjutkan dengan perintah untuk menampilkan teks Data diurutkan :

```
selection_sort();
```

Merupakan (); proses, akan dicari elemen-elemen yang belum diurutkan yang memiliki nilai terkecil atau terbesar akan dipertukarkan ke posisi yang tepat di dalam array.

```
for(int x = 0; x<7;x++)
```

Merupakan pendeklarasian *statement looping* for. Disini x=0 merupakan keadaan awal dimana i memiliki nilai 0 dengan tipe data integer, dengan syarat x kurang dari 7 dengan perubahan dijumlahkan dengan 1

```
cout<<setw(3)<<data[x];
```

Merupakan perintah untuk menampilkan setw(3) yang berfungsi untuk mengatur lebar variabel dan nilai dari variabel array data[x].

```
cout<<endl
```

Merupakan perintah untuk mengakhiri baris.

```
binarysearch();
```

Merupakan metode pencarian dimana data harus diurutkan terlebih dahulu sebelum dilakukan proses pencarian.

```
getche();
```

Merupakan fungsi agar program tidak tertutup setelah semua program selesai dieksekusi sehingga dapat melihat hasil dari perhitungan program.

```
return 0;  
}
```

Menyatakan hasil keluaran fungsi main adalah 0 atau untuk memberikan exit status yang berarti program berakhir dengan normal. Tanda } (kurung kurawal tutup) merupakan akhir dari fungsi main.

4. Program rekursif untuk memecahkan permasalahan menara honai!

```
#include <iostream>  
#include <conio.h> using  
namespace std;  
  
void hanoi(int n, char dari, char bantu, char tujuan)  
{  
    if(n==1)  
        cout<<"Pindahkan piring dari " << dari << " ke " << tujuan << "\n";  
    else{  
        hanoi(n-1, dari, tujuan, bantu);  
        hanoi(1, dari, bantu, tujuan);  
        hanoi(n-1, bantu, dari, tujuan);  
    }  
}  
  
int main(){  
    int jum_piring;
```

Analisa:

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

Merupakan pemberian intruksi untuk preprocessor directive `#include` agar menyertakan file header **`iostream`** sehingga memungkinkan penggunaan perintah `cin`, `cout`, dan `endl` dalam program ini. *Using namespace std;* berkaitan dengan fungsi dari file header `iostream` karena berisi tentang perintah seperti `cout`, `cin`, dan `endl`. *Using namespace std;* merupakan pernyataan (statement) yang selalu diakhiri dengan tanda `;` (titik koma). `#include <conio.h>` adalah header yang melibatkan `"getch()"` yang berfungsi agar program tidak tertutup setelah semua program selesai dieksekusi sehingga pengguna dapat melihat hasil dari perhitungan program.

```
void hanoi(int n, char dari, char bantu, char tujuan){
```

Merupakan pendeklarasian fungsi void dengan menggunakan parameter fungsi tipe data `char` variabel `dari`, `bantu`, `tujuan`, dan tipe data `int` variabel `n`. Bentuk umum dari pendeklarasian fungsi seperti `tipe_fungsi nama_fungsi (parameter_fungsi);` Tanda `{` (kurung kurawal buka) merupakan awal dari fungsi void

```
if(n==1)  
cout<<"Pindahkan piring dari " << dari << " ke " << tujuan << "\n";  
else{  
    hanoi(n-1, dari, tujuan, bantu); hanoi(1,  
    dari, bantu, tujuan); hanoi(n-1, bantu,  
    dari, tujuan);
```

Merupakan kondisi jika nilai variabel `n` yang dimasukkan adalah 1 maka akan mencetak teks Pindahkan piring dari kemudian menuliskan karakter pada variabel `dari` yang sudah diinputkan sebelumnya dan akan disimpan ke dalam *file*. Setelah itu akan menampilkan teks ke kemudian menuliskan karakter pada variabel `tujuan` yang sudah diinputkan sebelumnya dan akan disimpan ke dalam *file*.

```
int main(){
```

Merupakan fungsi main yang berisi statement-statement yang selalu diawali dengan tanda `{` (kurung kurawal buka) dan diakhiri dengan tanda `}` (kurung kurawal tutup).

Fungsi main akan menjalankan statement-statement yang berada di dalamnya secara berurutan. Tanda { (kurung kurawal buka) merupakan awal dari fungsi main

```
int jum_piring;
```

Merupakan pendeklarasian variabel jum_piring dengan tipe data integer.

```
cout<<"Teknik Hanoi Pada C++" <<endl<<endl;
```

Merupakan perintah untuk menampilkan teks dilanjutkan dengan perintah untuk menampilkan teks Teknik Hanoi Pada C++ berikutnya di baris baru.

```
cout<<"Masukkan Jumlah Piring      : ";
```

Merupakan perintah untuk menampilkan teks Masukkan Jumlah Piring :

```
cin>>jum_piring;
```

Merupakan perintah untuk menginputkan nilai variable jum_piring.

```
hanoi(jum_piring, 'A', 'B', 'C');
```

Merupakan pendeklarasian variabel hanoi di dalam fungsi main.

```
getche ();  
}
```

Merupakan fungsi agar program tidak tertutup setelah semua program selesai dieksekusi sehingga dapat melihat hasil dari perhitungan program.

5. Jelaskan proses rekursif untuk program dibawah ini!

```
void decToBin(int num){  
if (num > 0){  
decToBin(num / 2); cout  
<< num % 2;
```

```
}  
}
```

Analisa : pada program diatas mengubah decimal ke binary dengan menggunakan method Rekursif Tail. Prosesnya dengan pemanggilan rekursif diakhir method dan tidak memiliki fase balik. Untuk mengubah non rekursif menjadi rekursif tail diperlukan sebuah parameter tambahan sehingga dapat memanggil rekursif diakhir method.

```
void decToBin(int num){
```

Merupakan pendeklarasian fungsi void dengan menggunakan parameter fungsi tipe int variabel num. Bentuk umum dari pendeklarasian fungsi seperti tipe_fungsi nama_fungsi (parameter_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi void decTobin. if (num > 0).

```
if (num > 0){  
    decToBin(num / 2); cout  
    << num % 2;  
}
```

```
}
```

Merupakan kondisi jika akan berjalan ketika nilai variabel num > 0, maka akan dioperasikan ke dalam decTobin. Tanda { (kurung kurawal buka) merupakan awal dari fungsi if. *Cout << num%2;*}} menampilkan hasil output nilai variabel num modulu 2 pada hasil program. Tanda } (kurung kurawal tutup) merupakan akhir dari fungsi void. Tanda }(kurung kurawal tutup) merupakan aakhir dari fungsi if. Tanda }(kurung kurawal tutup) merupakan aakhir dari fungsi void decTobin.

6. Jelaskan proses untuk program dibawah ini!

```
boolean search(int[] x, int size, int n) {  
    if (size > 0) {  
        if (x[size-1] == n) { return  
            true; } else { return  
                search(x, size-1, n);  
            }  
    }  
}
```

Analisa:

```
boolean search(int[] x, int size, int n) {
```

Merupakan pendeklarasian variable search dengan tipe data boolean variabel x, size dan n dengan tipe data integer. Bentuk umum dari pendeklarasian fungsi seperti tipe_fungsi nama_fungsi (parameter_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi Boolean search.

```
if (size > 0) {
```

Merupakan kondisi jika nilai variabel size yang dimasukkan lebih dari 0.

```
if (x[size-1] ==  
n) { return true;  
} else { return  
search(x, size-1,  
n);  
} } return
```

Merupakan kondisi tidak (else) maka nilai balik (return value) akan dioperasikan ke dalam return search dan bernilai false. Tanda } (kurung kurawal tutup) merupakan akhir dari fungsi if. Tanda } (kurung kurawal tutup) merupakan akhir dari fungsi boolean.

7. Jelaskan proses rekursif untuk program dibawah ini!

```
boolean binarySearch(int[] x, int start, int end, int n) {  
if (end < start) return false;  
  
int mid = (start+end) / 2;  
if (x[mid] == n) { return  
true;  
} else {
```

```
if (x[mid] < n) {
```

Analisa:

```
boolean binarySearch(int[] x, int start, int end, int n) {
```

Merupakan pendeklarasian variable binarySearch dengan tipe data boolean, variabel x, start, end, dan n dengan tipe data integer. Bentuk umum dari pendeklarasian fungsi seperti tipe_fungsi nama_fungsi (parameter_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi boolean binarySearch.

```
if (end < start) return false;
```

Merupakan kondisi jika nilai variabel end yang dimasukkan nilainya lebih kecil dibandingkan variable start maka nilai balik (return value) adalah false.

```
int mid = (start+end) / 2;
```

Merupakan pendeklarasian tipe data int nilai variabel mid adalah (start+end)/2 yang berada di dalam fungsi Boolean binarysearch

```
if (x[mid] == n) { return  
true;  
}  
else { if (x[mid] < n) {  
return search(x, mid+1, end,  
n);
```

Merupakan Kondisi jika nilai variabel x [mid] adalah nilai n atau bernilai benar maka nilai balik (return value) adalah true. Tanda { (kurung kurawal buka) merupakan awal dari fungsi if. } else { if (x[mid] < n) { return search(x, mid+1, end, n); kondisi jika tidak (else if) akan berjalan ketika x[mid] < n maka nilai balik (return value) akan dioperasikan ke dalam return search.

8. Jelaskan proses rekursif dibawah ini dengan memanggil

```
mystery(2, 25) and mystery(3, 11)!  
int mystery(int a, int b) {  
  
if (b == 0) return 0; if  
(b % 2 == 0) return  
mystery(a+a, b/2);
```

Analisa:

```
int mystery(int a, int b) {
```

Merupakan pendeklarasian variable mystery dengan tipe data integer variabel a dan b. Bentuk umum dari pendeklarasian fungsi seperti tipe_fungsi nama_fungsi (parameter_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi int mystery.

```

if (b == 0)
return 0; if (b
% 2 == 0)

return mystery(a+a, b/2);
return mystery(a+a, b/2) + a;

```

Merupakan kondisi jika nilai variabel b yang dimasukkan sama dengan 1 maka nilai balik (return value) adalah nilai 0.

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

```

int mystery(int a, int b) {
if (a == b) cout<<a<<endl;
else {

int m1 = (a + b) / 2; int
m2 = (a + b + 1) / 2;
mystery(a, m1);
mystery(m2, b);

}
}

```

Analisa:

```

int mystery(int a, int b) {

```

Merupakan pendeklarasian variable mystery dengan tipe data integer variabel a dan b. Bentuk umum dari pendeklarasian fungsi seperti tipe_fungsi nama_fungsi (parameter_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi int mystery.

```

if (a == b) cout<<a<<endl;
else { int m1 = (a + b) /
2;

int m2 = (a + b + 1) / 2;
mystery(a, m1);
mystery(m2, b);

}
}

```

Merupakan Kondisi jika nilai variabel a sama dengan nilai variabel b atau bernilai benar maka akan menampilkan nilai variabel a dan berpindah pada baris baru. *else { int m1 = (a+b)/2 ;int m2 = (a+b+1)/2;* kondisi tidak (else) maka nilai akan dioperasikan ke dalam deklarasi variabel m1 dan m2 tipe data int. *mystery(a, m1);mystery(m2, b);}* pendeklarasian variabel mystery di dalam fungsi int mystery. Tanda } (kurung kurawal tutup) merupakan aakhir dari fungsi int mystery.

10. Jelaskan proses rekursif untuk program dibawah ini!

```
int f(int n) {  
    if (n == 0)  
        return 0;  
  
    if (n == 1) return 1; if  
    (n == 2) return 1; return  
    2*f(n-2) + f(n-3);  
}
```

Analisa:

```
int f(int n) {
```

Merupakan pendeklarasian variable f dengan tipe data integer variabel n. Bentuk umum dari pendeklarasian fungsi seperti tipe_fungsi nama_fungsi (parameter_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi int f

```
    if (n == 0)  
        return 0;
```

Merupakan kondisi jika nilai variabel n yang dimasukkan adalah 0 maka nilai balik (return value) adalah nilai 0.

```
    if (n == 1)  
        return 1;
```

Merupakan kondisi jika nilai variabel n yang dimasukkan adalah 1 maka nilai balik (return value) adalah nilai 1.

```
    if (n == 2) return 1; return 2*f(n-2) + f(n-3);
```

Merupakan kondisi jika nilai variabel n yang dimasukkan adalah 2 maka nilai balik (return value) adalah nilai 1 dan dioperasikan ke dalam return $2*f(n-2) + f(n-3)$.

11. Jelaskan proses rekursif untuk program dibawah 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;  
} int cube(int  
n) {
```

```
if (n == 0) return 0;
```

Analisa:

```
int square(int n) {
```

Merupakan pendeklarasian variable square dengan tipe data integer variabel n. Bentuk umum dari pendeklarasian fungsi seperti tipe_fungsi nama_fungsi (parameter_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi int square.

```
if (n == 0) return 0;  
return square(n-1) + 2*n - 1;
```

Merupakan kondisi jika nilai variabel n yang dimasukkan adalah 0 maka nilai balik (return value) adalah nilai 0.

```
int cube(int n) {
```

Merupakan pendeklarasian fungsi cube dengan menggunakan parameter formal fungsi tipe data int variabel n. Bentuk umum dari pendeklarasian fungsi seperti tipe_fungsi nama_fungsi (parameter_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi int cube.

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

Merupakan Kondisi jika nilai variabel n=0 bernilai benar maka nilai balik (return value) adalah 0 dan dioperasikan ke dalam $return\ cube\ (n-1)+3*(square(n))-3*n+1$. Tanda } (kurung kurawal tutup) merupakan akhir dari fungsi int cube.