

### C. TUGAS PENDAHULUAN

Jawablah pertanyaan berikut ini:

1. Apa yang dimaksud dengan rekursi?  
Rekursi adalah suatu proses atau prosedur dari fungsi yang memanggil dirinya sendiri secara berulang-ulang.
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

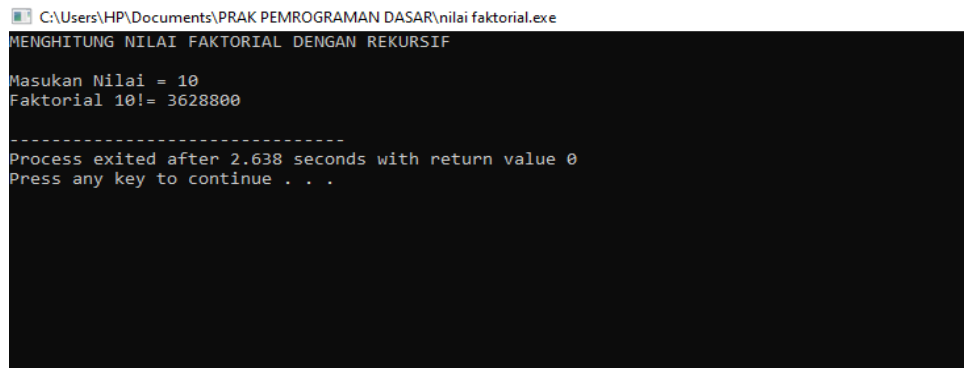
```
#include <iostream>
using namespace std;
int fibonacci(int m) {
    if (m == 0 || m ==1)
    {
        return m;
    }
    else
    {
        return (fibonacci(m-1) + fibonacci(m-2));
    }
}
int main() {
    int n, m= 0;
    cout << "Masukan Batas Deret Bilangan Fibonacci : ";
    cin >> n;
    cout << "Deret Fibonacci: ";
    for (int i = 1; i <= n; i++){
        cout << fibonacci(m) <<" ";
        m++;
    }
    return 0;
}
```

## D. PERCOBAAN

### Percobaan 1: Fungsi rekursif 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));
}
```

### Hasil percobaan 1



```
C:\Users\HP\Documents\PRAK PEMROGRAMAN DASAR\nilai faktorial.exe
MENGHITUNG NILAI FAKTORIAL DENGAN REKURSIF
Masukan Nilai = 10
Faktorial 10!= 3628800
-----
Process exited after 2.638 seconds with return value 0
Press any key to continue . . .
```

### 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 int faktorial (int A);
```

Merupakan pendeklarasian fungsi faktorial dengan menggunakan parameter formal fungsi tipe data int variabel A. Bentuk umum dari pendeklarasian fungsi seperti `tipe_fungsi nama_fungsi (parameter_fungsi);`

```
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 r, hasil;
```

Merupakan pendeklarasian variabel r dan hasil dengan tipe data integer.

```
cout<<"MENGHITUNG NILAI FAKTORIAL DENGAN REKURSIF"<<endl;
```

Merupakan perintah untuk menampilkan teks **MENGHITUNG NILAI FAKTORIAL DENGAN REKURSIF** dilanjutkan dengan perintah untuk menampilkan teks berikutnya di baris baru.

```
cout<<"Masukan Nilai = ";
```

Merupakan perintah untuk menampilkan teks **Masukan Nilai**.

```
cin>>r;
```

Merupakan perintah untuk menginputkan nilai variabel r.

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

Merupakan pendefinisian variabel hasil yang merujuk pada fungsi factorial 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 } (kurung kurawal tutup) merupakan akhir dari fungsi main.

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

Merupakan fungsi long int factorial dengan menggunakan parameter formal fungsi tipe data int variabel A yang berisi statement-statement yang selalu diawali dengan tanda { (kurung kurawal buka) dan diakhiri dengan tanda } (kurung kurawal tutup). Fungsi long int faktorial akan menjalankan statement-statement yang berada di dalamnya secara berurutan. Tanda { (kurung kurawal buka) merupakan awal dari fungsi long int faktorial.

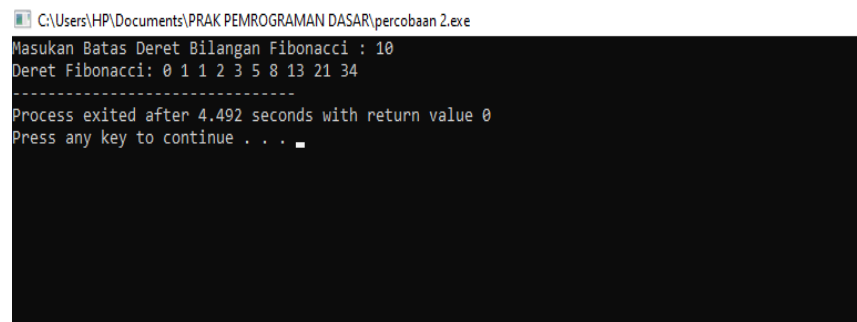
```
if (A==1)
return(A);
else
return (A*faktorial(A-1));
}
```

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

## Percobaan 2: Fungsi untuk menampilkan deret fibonaci

```
#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;
}
```

## Hasil percobaan 2



```
C:\Users\HP\Documents\PRAK PEMROGRAMAN DASAR\percobaan 2.exe
Masukan Batas Deret Bilangan Fibonacci : 10
Deret Fibonacci: 0 1 1 2 3 5 8 13 21 34
-----
Process exited after 4.492 seconds with return value 0
Press any key to continue . . .
```

## 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).

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

Merupakan pendeklarasian fungsi Fibonacci dengan menggunakan parameter formal fungsi tipe data integer variable `m`. Bentuk umum dari pendeklarasian fungsi seperti `tipe_fungsi nama_fungsi (parameter_fungsi)`; Tanda `{` (kurung kurawal buka) merupakan awal dari fungsi `int fibonacci`.

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

Kondisi jika nilai variabel `m` yang dimasukkan adalah 0 atau 1 maka nilai balik (return value) adalah nilai itu sendiri. }Else{ `return (fibonacci(m-1) + fibonacci(m-2));` kondisi tidak (else) maka nilai balik yang dimasukkan akan dihitung menggunakan rumus fibonacci tersebut. Tanda `}` (kurung kurawal tutup) merupakan akhir dari fungsi `int fibonacci`.

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

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

```
cout << "Masukan Batas Deret Bilangan Fibonacci : ";
```

Merupakan perintah untuk menampilkan teks **Masukan Batas Deret Bilangan Fibonacci :**

```
cin >> n;
```

Merupakan perintah untuk menginputkan nilai variable `n`.

```
cout << "Deret Fibonacci: ";
```

Merupakan perintah untuk menampilkan teks **Deret Fibonacci**:

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

Merupakan perintah untuk memberi nilai awal perulangan yang dimulai dari  $i=1$ . Pendeklarasian variabel  $i$  tipe data `int` digunakan untuk perulangan. Perulangan akan berhenti jika nilai  $i \leq$  (lebih besar sama dengan) inputan nilai  $n$ .  $i++$  perulangan bersifat increment atau akan ditambah 1 setiap kali proses hingga sampai batas yang ditentukan maka proses akan berhenti. Tanda `{` (kurung kurawal buka) merupakan awal dari fungsi `for`.

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

Merupakan perintah menampilkan output fungsi Fibonacci variabel  $m$  dan memberikan spasi.  $M++$  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 `for`.

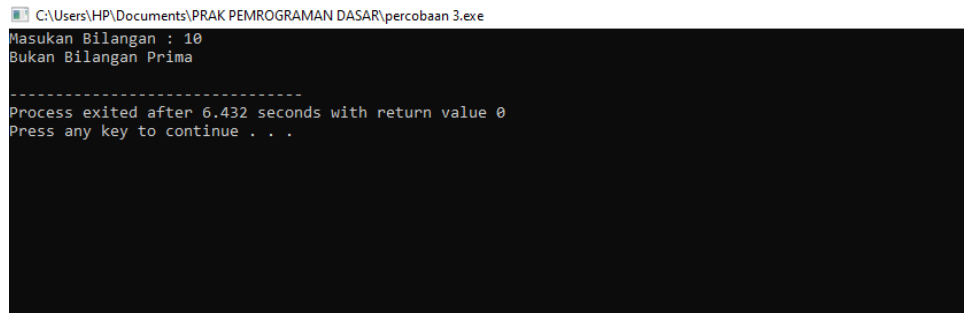
```
return 0;  
}
```

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.

### Percobaan 3: Fungsi rekursi untuk menentukan bilangan prima atau bukan prima

```
#include <iostream>
using namespace std;
int ambil(int bil, int i){
    if (i == 1) {
        return 1;
    }
    else if (bil % i == 0) {
        return 1 + ambil(bil, --i);
    } else {
        return 0 + ambil(bil, --i);
    }
}
int cek(int bil){
    if (bil > 1) {
        return (ambil(bil, bil) == 2);
    } else
        return false;
}
int main(){
    int bil;
    cout<<"Masukan Bilangan : ";
    cin>>bil;
    if (cek(bil)){
        cout<<"Bilangan Prima"<<endl;
    }else {
        cout<<"Bukan Bilangan Prima"<<endl;
    }
    return 0;
}
```

### Hasil percobaan 3



```
C:\Users\HP\Documents\PRAK PEMROGRAMAN DASAR\percobaan 3.exe
Masukan Bilangan : 10
Bukan Bilangan Prima
-----
Process exited after 6.432 seconds with return value 0
Press any key to continue . . .
```

### 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).

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

Merupakan pendeklarasian variabel ambil dengan tipe data integer variabel bil dan i. Bentuk umum dari pendeklarasian fungsi seperti tipe\_fungsi nama\_fungsi (parameter\_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi int ambil.

```
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 dimasukkan adalah 1 maka nilai balik (return value) adalah nilai 1. *}Else if (bil%i==0){ return 1+ambil(bil, --i);* kondisi jika tidak (else if) apabila nilai variabel bil modulo i = 0 maka nilai balik yang dimasukkan akan dihitung menggunakan rumus bilangan prima tersebut. *}Else { return 0+ambil(bil, --i);* kondisi jika tidak (else) maka nilai balik yang dimasukkan akan dihitung menggunakan rumus bilangan prima tersebut. Tanda } (kurung kurawal tutup) merupakan akhir dari fungsi int ambil.

```
int cek(int bil){
```

Merupakan pendeklarasian variabel cek dengan tipe data integer variabel bil. Bentuk umum dari pendeklarasian fungsi seperti tipe\_fungsi nama\_fungsi (parameter\_fungsi); Tanda { (kurung kurawal buka) merupakan awal dari fungsi int cek.

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

Merupakan kondisi jika nilai variabel bil yang dimasukkan adalah (lebih dari) >1 atau bernilai benar maka nilai balik (return value) adalah nilai 2. *}Else return false;* kondisi tidak (else) maka nilai balik yang dimasukkan akan bernilai salah. Tanda } (kurung kurawal tutup) merupakan akhir dari fungsi int cek.

```
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 bil;
```

Merupakan pendeklarasian variabel bil dengan tipe data integer.

```
cout<<"Masukan Bilangan : ";
```

Merupakan perintah untuk menampilkan teks **Masukan Bilangan**.

```
cin>>bil;
```

Merupakan perintah untuk menginputkan nilai variable bil.

```
if (cek(bil)){  
    cout<<"Bilangan Prima"<<endl;  
}else {  
    cout<<"Bukan Bilangan Prima"<<endl;  
}  
return 0;  
}
```

Merupakan kondisi jika nilai yang dimasukkan bernilai benar di cek bil maka akan menampilkan output dalam bentuk kalimat Bilangan Prima pada hasil program.}else { *cout<<"Bukan Bilangan Prima"<<endl;*} kondisi tidak (else) nilai yang dimasukkan salah maka akan menampilkan menampilkan output dalam bentuk kalimat Bukan Bilangan Prima pada hasil program.

#### Percobaan 4: Fungsi rekursi untuk menghitung pangkat

```
#include <iostream>  
using namespace std;  
long int pangkatrekursif(int x, int y);  
int main(){  
    int x,y;  
    cout<<"FUNGSI REKURSIF UNTUK MENGHITUNG PANGKAT"<<endl;  
    cout<<endl;  
    cout<<"Masukan Nilai X = ";  
    cin>>x; cout<<"Masukan Nilai Y = ";  
    cin>>y;  
    cout<<endl;  
    cout<<x<<" Dipangkatkan "<<y<<" = "<<pangkatrekursif(x,y)<<endl;  
}  
long int pangkatrekursif(int x, int y){  
    if (y==0)  
        return 1 ;  
    else  
        return x * pangkatrekursif(x,y-1); }  
}
```

## Hasil percobaan 4

```
C:\Users\HP\Documents\PRAK PEMROGRAMAN DASAR\percobaan 4.exe
FUNGSI REKURSIF UNTUK MENGHITUNG PANGKAT
Masukan Nilai X = 12
Masukan Nilai Y = 6

12 Dipangkatkan 6 = 2985984
-----
Process exited after 5.152 seconds with return value 0
Press any key to continue . . .
```

## 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 int pangkatrekursif(int x, int y);
```

Merupakan fungsi long integer `pangkatrekursif` dengan menggunakan parameter formal fungsi tipe data integer variabel `x` dan `y`. Bentuk umum dari pendeklarasian fungsi seperti tipe\_fungsi nama\_fungsi (parameter\_fungsi);

```
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 x,y;
```

Merupakan pendeklarasian variabel `x` dan `y` dengan tipe data integer.

```
cout<<"FUNGSI REKURSIF UNTUK MENGHITUNG PANGKAT"<<endl;
```

Merupakan perintah untuk menampilkan teks dilanjutkan dengan perintah untuk menampilkan teks **`FUNGSI REKURSIF UNTUK MENGHITUNG PANGKAT`** berikutnya di baris baru.

```
cout<<"Masukan Nilai X = ";
```

Merupakan perintah untuk menampilkan teks **Masukan Nilai X =** pada layar.

```
cin>>x;
```

Merupakan perintah untuk menginputkan nilai variable x.

```
cout<<"Masukan Nilai Y = ";
```

Merupakan perintah untuk menampilkan teks **Masukan Nilai Y =** pada layar.

```
cin>>y;
```

Merupakan perintah untuk menginputkan nilai variable y.

```
cout<<x<<" Dipangkatkan "<<y<<" = "<<pangkatrekursif(x,y)<<endl;
}
```

Merupakan perintah untuk menampilkan output nilai variabel x dengan kalimat Dipangkatkan nilai variabel y yang sama-sama diinputkan oleh keyboard dan menampilkan tanda = beserta dengan variabel pangkatrekursif(x,y) pada hasil program. Kemudian berpindah pada baris baru. Tanda } (kurung kurawal tutup) merupakan akhir dari fungsi main.

```
long int pangkatrekursif(int x, int y)
```

Merupakan fungsi long integer pangkatrekursif dengan menggunakan parameter formal fungsi tipe data integer variabel x dan y yang berisi statement-statement yang selalu diawali dengan tanda { (kurung kurawal buka) dan dihiri dengan tanda } (kurung kurawal tutup). Fungsi long int pangkat rekursif akan menjalankan statement-statement yang berada di dalamnya secara berurutan. Tanda { (kurung kurawal buka) merupakan awal dari fungsi long int pangkat rekursif.

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

Merupakan kondisi jika nilai variabel y yang dimasukkan adalah 1 maka nilai balik (return value) adalah nilai 1. *Else return x \* pangkatrekursif(x,y-1);* kondisi tidak (else) maka nilai balik yang dimasukkan akan dihitung menggunakan rumus pangkat tersebut. Tanda } (kurung kurawal tutup) merupakan akhir dari fungsi long int pangkatrekursif.

## E. Latihan

1. Buatlah 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 << " ";
        }

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

        cout << endl;
    }

    return 0;
}
```

## Analisis

```
#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 factorial..

```
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 dihiri 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. Buatlah 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)
{
    printLebarChar(set, "", n, k);
}

int main()
{
    char set1[] = {'a', 'b', 'c'};
    int k = 3;
    cout << "Jumlah karakter : " ;
    cin >> k;

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

### 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.

```
if (k == 0)
{
    cout << (huruf) ;
    cout << " ";
    return;
}
```

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];
        printLebarChar(set, newHuruf, n, k - 1);
    }
}
```

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. Buatlah 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])
            {
                min=j;
            }
        }
        temp = data[i];
        data[i] = data[min];
        data[min] = temp;
    }
}

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\t 'BINARY SEARCH'"<<endl;
    cout<<"\t\t===== "<<endl;
    cout<<"\nData\t\t\t\t\t: ";
    //tampilkan data awal
    for(int x = 0; x<7; x++)
        cout<<setw(3)<<data[x];
    cout<<endl;
}
```

```

        cout<<"\nMasukkan data yang ingin Anda cari : ";
        cin>>cari;
        cout<<"\nData diurutkan : ";
        //urutkan data dengan selection sort
        selection_sort();
        //tampilkan data setelah diurutkan
        for(int x = 0; x<7;x++)
            cout<<setw(3)<<data[x];

        cout<<endl;

        binarysearch();

        getch();
        return 0;
    }

```

## 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 menginclude 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;
    break;
}
```

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 dikhiri 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 '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 tudak 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. Buatlah 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;

    cout<<"Teknik Hanoi Pada C++" <<endl<<endl;
    cout<<"Masukkan Jumlah Piring: ";
    cin>>jum_piring;

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

#### 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 variabel *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 latihan soal ini, mengubah decimal ke binary dengan method RekursifTail. Prosesnya yaitu 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)
    {
        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);
        }
    }
    return false;
}
```

## 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  
false;  
}
```

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) {  
            return search(x, mid+1, end, n);  
        } else {  
            return search(x, start, mid-1, 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 varabel 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);  
    } else {  
        return search(x, start, mid-1, 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);  
    return mystery(a+a, b/2) + a;  
}
```

### 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 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;
return cube(n-1) + 3*(square(n)) - 3*n + 1;
}
```

## 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.