

**LAPORAN HASIL ANALISA TUGAS PRAKTIKUM KE-6**  
**(BAGIAN C, D, DAN E)**  
**MATA KULIAH PEMROGRAMAN DASAR**



**OLEH :**  
**SALSABILLA OCTAVIANINGRUM**  
**21091397005**  
**D4 MANAJEMEN INFORMATIKA**

**UNIVERSITAS NEGERI SURABAYA**  
**2021/2022**

## C. TUGAS PENDAHULUAN

1. Rekursi adalah suatu proses yang bisa memanggil dirinya sendiri. Dalam sebuah rekursi, terdapat pengertian sebuah prosedur atau fungsi. Perbedaannya adalah rekursi bisa memanggil dirinya sendiri, kalau prosedur atau fungsi harus dipanggil melalui pemanggil prosedur atau fungsi.
2. Fungsi untuk menghitung nilai faktorial :  $n! = n \cdot (n-1)!$  Untuk  $n > 1$   
 $0! = 1$  Untuk  $n = 0$  atau  $n = 1$   
Secara pemrograman dapat ditulis sebagai berikut :  
Faktorial (0) = 1 (1)  
Faktorial (N) = N \* Faktorial (N - 1) (2)  
Persamaan (1) tidak bersifat rekursif, disebut nilai awal atau basis. Persamaan (2) contoh hubungan rekursif yang artinya nilai suatu fungsi dengan argumen tertentu bisa dihitung dari fungsi yang sama dengan argumen yang lebih kecil.
3. Fungsi untuk menampilkan nilai fibonacci dari deret fibonacci :  
Fibonacci(1) = 1 (1)  
Fibonacci(2) = 2 (2)  
Fibonacci(N) = Fibonacci(N-1) + Fibonacci(N-2) (3)

## D. PERCOBAAN

Percobaan 1: Fungsi rekursif untuk menghitung nilai faktorial

```
#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 :

```
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 . . .
```

Hasil 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 variable 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 factorial tersebut. Tanda } (kurung kurawal tutup) merupakan akhir dari fungsi long int faktorial.

Percobaan 2: Fungsi untuk menampilkan deret fibonacci

```

#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 :

```

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 . . .

```

Hasil 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 :**

Merupakan perintah untuk menginputkan nilai variable n.

```
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);

    }return else {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 : ";

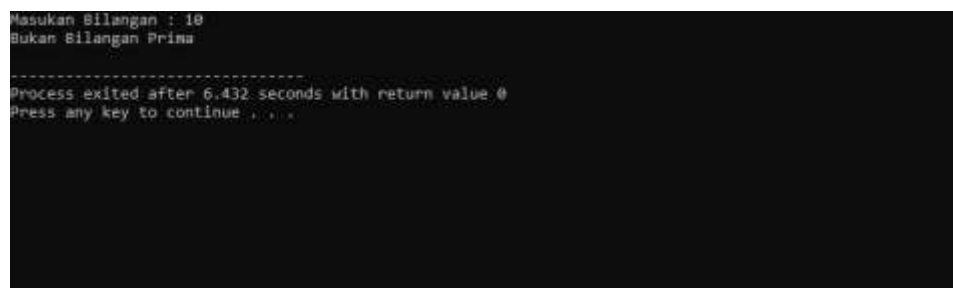
        cinif >>(cekbil(;bil )){

        cout}else <<"Bilangan Prima"{
            <<endl;

            cout<<"Bukan Bilangan Prima"<<endl;
        } return
        0;
    }

```

Hasil percobaan :



```

Masukan Bilangan : 10
Bukan Bilangan Prima

-----
Process exited after 6.432 seconds with return value 0
Press any key to continue:

```

Hasil 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);
    } return else {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 return (bil (>ambil 1) {(
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 :

```

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 . . .

```

Hasil 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 =**

```
cin>>x;
```

Merupakan perintah untuk menginputkan nilai variable x.

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

Merupakan perintah untuk menampilkan teks **Masukan Nilai Y =**

```
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 diakhiri 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 else 1 ;
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 pangkat rekursif.

## E. LATIHAN

1. Buatlah program rekursif untuk menghitung segitiga pascal!

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

int pascal(int row, int col)
{
    if (col == 0 || col == row)
    {
        return 1;
    }
    else
    {
        return pascal(row - 1, col - 1) + pascal(row - 1, col); }
}
```

```

int main() {
    system("cls");    int
    row;    cout<<"Enter
    n : ";    cin>>row;

    for (int i=0;i<row;i++)
    {
        for(int col =0;col<=i;col++)
            cout<<pascal(i,col);

        cout<<"\n";
    }

    return 0;
}

```

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

```

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

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);
    } }

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

int main() {
    char set1[] = {'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 << "jumlah karakter:" << ' ';
    while(cin >> N){      if(N != -1){
    printAllKLength(set1,N, N)  ;
        cout << "jumlah karakter:" << ' ';
        }
        else      if(N>26){
break;
        }
        else{
break;
        }

    }
    return 0;
}

```

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

```

#include <iostream>
#include <conio.h>

using namespace std;

int data[7] = { 1, 8, 2, 5, 4, 9, 7};
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                                     'BINARY           SEARCH'"<<endl;
cout<<"\t===== "<<endl;
    cout<<"\nData          : ";
//tampilkan data awal
for(int x = 0; x<7; x++)
cout<<(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<<(3)<<data[x];

        cout<<endl;

        binarysearch();

        _getche();
        return 0;

    }

```

4. Buatlah program rekursif untuk memecahkan permasalahan Menara Hanoi!

```

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

void menara_hanoi(int disk, char dari, char bantu, char ke)
{
    if(disk>0);
    {
        menara_hanoi(disk-1,dari,ke,bantu);
        printf("Pindahkan Disk %d darimpasak %c ke pasak %c\n", disk,dari,ke);
        menara_hanoi(disk-1,bantu,dari,ke);
    }
}

int main()
{
    int disk;
    printf("Jumlah disk =");scanf ("%d",&disk);
    printf("\n");
    menara_hanoi(disk, 'A','B','C');

    return 0;
}

```

5. Jelaskan proses rekursif untuk program dibawah ini!

menjelaskan tentang fungsi konversi dari desimal ke biner

```

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

```

jika data yang disimpan pada variabel num lebih besar dari 0 akan di bagi dengan 2 dan ditampilkan sisa hasil bagiannya

6. Jelaskan proses rekursif 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;
}

```

menjelaskan tentang ukuran dari sebuah

menjelaskan ukuran dari sebuah bilangan n.

7. Jelaskan proses rekursif untuk program dibawah ini!

```

boolean binarySearch(int[] x
    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,
        } else {
            return search(x, start,
        }
    }
}
}
}

```

menghitung angka  
biner  
menggunakan  
boolean.

hasil awal dan  
akhir dijumlah dan  
dibagi dengan 2.

9. Jelaskan proses rekursif untuk program dibawah ini dengan memanggil mystery(2, 25) and mystery(3, 11)!

program mencari  
data nilai

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;  
}
```

jumlah nilai akan  
dijumlah dan  
dibagi 2

10. 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);  
    }  
}
```

nilai a dan b akan dijumlah  
dan dibagi 2

nilai a dan b dijumlah  
dengan 1 lalu dibagi 2

11. . Jelaskan proses rekursif untuk program dibawah ini!

program nilai n

```
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);
}
```

nilai akan dijumlahkan

12. Jelaskan proses rekursif untuk program dibawah ini dengan memanggil square(5), cube(5), cube(123)?

```
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;
}
```

membuat program dengan 5  
nilai yang berbeda

nilai akan dikurang 1  
ditambah 2 dan nilai lainnya

akan dikurang 1.